COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MASTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Monitorización inteligente y análisis de aprendizaje no supervisado aplicados sobre variables de la Red Eléctrica

Autor: Andrés García Domínguez

Director: Carlos Gaitán Poyatos

Codirector: Álvaro López López

Madrid

Julio de 2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título:

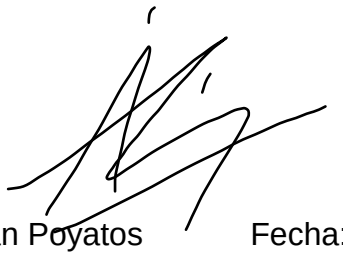
Monitorización inteligente y análisis de aprendizaje no supervisado aplicados sobre variables de la Red Eléctrica

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2020 - 2021 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Andrés García Domínguez Fecha: 23/07/2021

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO

CARLOS GAITAN POYATOS Digitally signed by CARLOS GAITAN POYATOS Date: 2021.07.20 10:47:26 +02'00'



Fdo.: Carlos Gaitán Poyatos Fecha: 23/07/2021

Álvaro López López

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Andrés García Domínguez DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Monitorización inteligente y análisis de aprendizaje no supervisado aplicados sobre variables de la Red Eléctrica**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente).*

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 23 de Julio de 2021

ACEPTA

Fdo: Andrés García Domínguez

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MASTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Monitorización inteligente y análisis de aprendizaje no supervisado aplicados sobre variables de la Red Eléctrica

Autor: Andrés García Domínguez

Director: Carlos Gaitán Poyatos

Codirector: Álvaro López López

Madrid

Julio de 2021

Agradecimientos

Dedico este Proyecto a:

- A todo el equipo investigador de la *Cátedra de Industria Conectada*, que me tutelaron e hicieron de esta beca de aprendizaje una experiencia única.
- A Carlos, que siempre ofreció su mejor opinión del proyecto.
- Mis compañeros de piso de la *Rue Adolphe 39*, que me acompañaron en Luxemburgo a lo largo del camino.
- A Paloma, que me dio la ayuda que necesitaba.
- Y finalmente, a mi padre y mi madre, responsables del ingeniero pero sobre todo la persona en la que me he convertido hoy.

MONITORIZACIÓN INTELIGENTE Y ANÁLISIS DE APRENDIZAJE NO SUPERVISADO APLICADOS SOBRE VARIABLES DE LA RED ELÉCTRICA.

Autor: García Domínguez, Andrés.
Director: Gaitán Poyatos, Carlos.
Co-Director: López López, Álvaro
Entidad Colaboradora: Smart City Málaga

RESUMEN DEL PROYECTO

Con la colaboración de la **Cátedra de Industria Conectada**, este proyecto parte de las variables de operación de los **Centros de Transformación** malagueños para desarrollar **herramientas de extracción de información aglomerada**. El resultado principal es el desarrollo del **tablero de mando** en línea **basado en *Dash Plotly*** para la visualización multiplataforma y **análisis profundo** por técnicas de **reducción de dimensionalidad** para la extracción del espacio latente en 2D y su posterior análisis por clusterización.

Palabras clave: *Aplicación Dash, reducción dimensional, detección de valores anómalos*

1. Introducción

En la actualidad, es creciente el interés que existe por **desarrollar sistemas avanzados de control** sobre la red eléctrica que puedan agilizar las tareas de mantenimiento y operación de las infraestructuras críticas para el bienestar de un país. Conforme la **demanda eléctrica** crece decididamente, la **complejidad de la red** aumenta en escala y naturaleza, y la **interoperación entre activos** se vuelve crítica, mayor es la eficacia y capacidad que se exige a los sistemas de control que subyacen dentro de la arquitectura.

Entre las vertientes mayor interés causan dentro de la revolución que se está dando en el campo, se encuentra el control y supervisión del sistema y, concretamente, el responsable de la **monitorización, control y protección de las subestaciones**. Se ha visto como funciones tradicionalmente aisladas se fusionan hasta basarse en un menor número de dispositivos, pero conformando un sistema mejorado por los modelos probabilísticos, las leyes de inferencia sobre estados latentes a partir de variables medibles o las visualizaciones online para realizar análisis descriptivo, predictivo y en última instancia, prescriptivo.

El interés por este tipo de modelos va en aumento, lo cual está llevando a la creación de cada vez más modelos y más robustos, así como de aplicaciones de procesamiento en tiempo real para la inspección y la construcción de sistemas de alarma alrededor de estos. [1] [2]

2. Definición del proyecto

El Análisis Avanzado de Datos y la Inteligencia Artificial están evolucionando a una velocidad vertiginosa y, sin embargo, aún queda mucho camino por recorrer. En la actualidad, muchos métodos afloran de las aportaciones que llegan desde la Estadística

Aplicada, pero muchas veces es en la implementación donde empresas necesitan de un esfuerzo mayor. Con este proyecto, se pretende responder a las **necesidades de Smart City Málaga** (de aquí en adelante, también denominado SCM) para **desarrollar un modelo web multiplataforma en tiempo real** que aporte informaciones de alto valor añadido y hagan uso de estas técnicas inteligentes. Se pretende pues crear una base para futuros modelos destinados a la resolución de estos problemas o parecidos, desde la asistencia a personal destinada al lugar físico, a la ejecución de gestores de alarmas o detección de anomalías.

En términos de objetivos, han sido acordados y concretados en el listado a continuación:

- Conseguir una latencia e interacción propia de un servicio web (menor de 1 segundo).
- Permitir la generación automática de informes de la salud y congestión de la red.
- Crear un modelo lo más robusto posible y agnóstico al dato y plataforma: evitar la pérdida de calidad frente a variaciones implícitas o explicitas de los datos de entrada, y que el programa no dependa del dispositivo, sistema operativo o versión del visor.
- Apalancar el análisis y conclusiones del estudio sobre herramientas inteligentes de reducción de dimensiones (error menor del 15% sobre el espacio reconstruido).

3. Descripción del modelo/sistema/herramienta

La meta del proyecto es desarrollar un producto digital elaborado para permitir el seguimiento y evolución temporal de la congestión eléctrica dentro de un área geográfica dada. Como herramienta de interacción en tiempo real y con portabilidad flexible en dispositivos de distinto tipo, las condiciones de resultado exigidas eran claras y se apoyaban en tres ejes diferenciados: **facilidad de uso**, **interpretabilidad de los resultados** y **estabilidad de la integridad de la página** (tanto front-end como back-end).

La primera sección se reserva para el **estudio cualitativo y cuantitativo preliminar** que se realiza sobre la totalidad de las variables de partida. Con ayuda de tres métodos distintos de evaluación de datos (librerías Sweetviz [3], Autoviz [4] y Lux [5]), se aprecian las relaciones que atan el comportamiento de ciertas variables físicas frente a otras, así como la propia repartición de observaciones dentro de cada una de ellas.

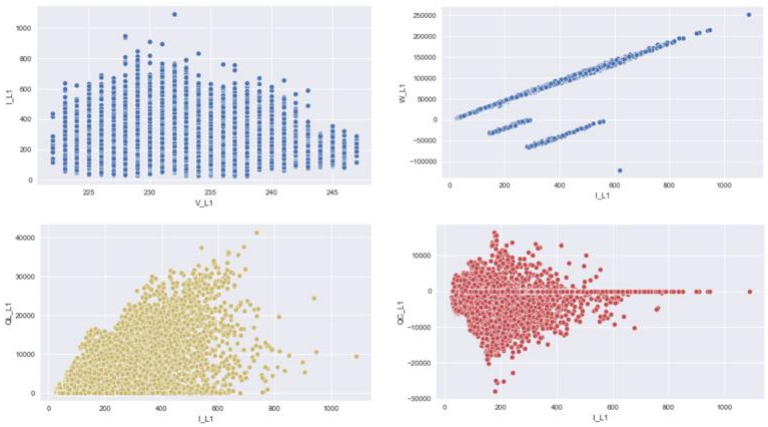


Figura 1: Ejemplo de Análisis Exploratorio de Datos con Autoviz

En base a las conclusiones extraídas (en particular, evaluación de la calidad de los datos, naturaleza de los valores y correlaciones entre variables), se decide una guía de **preprocesamientos** (limpieza, sustitución de valores, formateado y normalización) **y selección de atributos** que redunda en última instancia en la importación de datos sobre la herramienta.

La segunda sección responde a la generación de resultados en una etapa intermedia de la evolución cronológica del proyecto, donde se sientan las bases para definir la mitad de la herramienta encargada del apartado más descriptivo. Más allá de escoger un tipo de gráficas antes que otras, por encima de todo **sienta las bases de la estructura global de la aplicación dentro de la plantilla Dash**, factor clave para la progresión del trabajo. Primeramente, se abordó el diseño esquemático que engloba las gráficas de interés y que en última instancia se traduce al mundo visualización – estilo – interactividad (HTML – CSS – JavaScript).

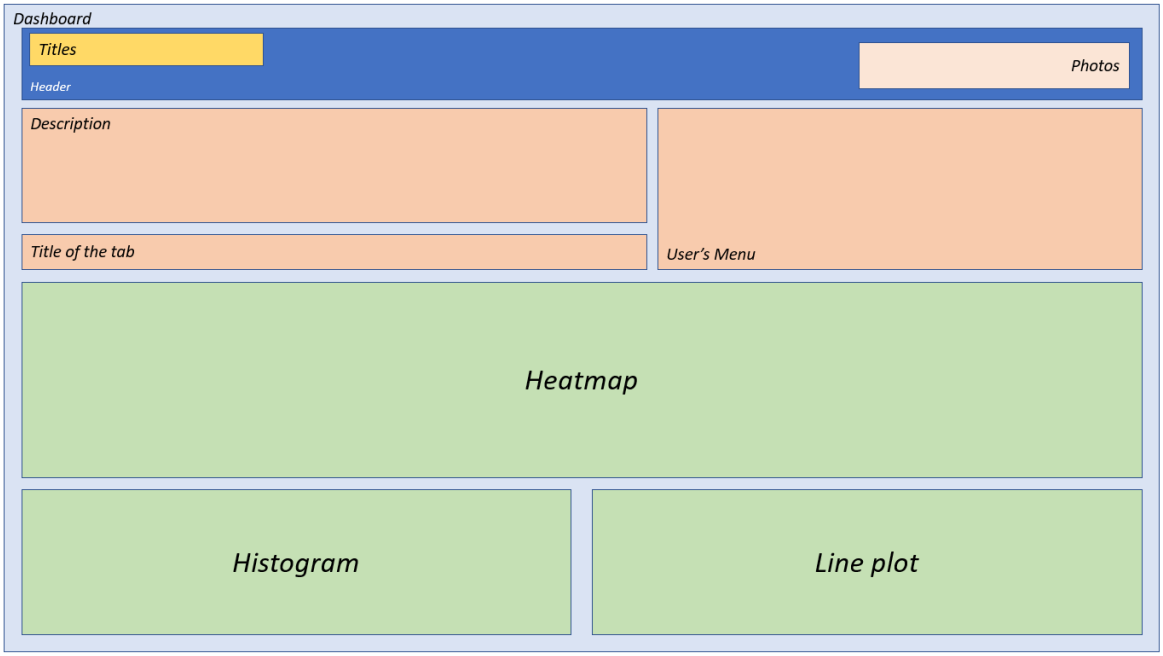


Figura 2: Esquema conceptual del tablero

Tras pasar por iteraciones para conseguir un preprocesamiento correcto y la elección idónea de servicio para la representación geográfica de mapa de calor, todo ello sobre la plataforma Google Colab, se lleva a cabo un trabajo de investigación alrededor de los objetos disponibles en Dash (esencialmente Dash_html_components, Dash_core_components, Dash_bootstrap_components). En base a estos, se sigue la estructura general propuesta de las mejores prácticas para el desarrollo de aplicaciones: 1) **Importar las librerías** y frameworks de Python, 2) **Procesar la información** de partida, 3) Construir la **plantilla (*layout*) de la aplicación**, con elementos de entrada y salida en su interior y 4) Especificar en las funciones finales (denominadas ***callbacks***) las **interacciones que existirán en la página web** frente a eventos del usuario.

El resultado de la primera versión, principalmente descriptiva, es similar a la primera pestaña de la Figura 4 y Figura 5.

En la tercera sección se enfrenta de lleno la mitad de la herramienta encargada del trabajo de análisis profundo, y presenta el trabajo de análisis exhaustivo que se realizó alrededor de los dos problemas de aprendizaje automático previamente evocados: **la reducción dimensional**, y la **detección de mediciones anómalas**. Apoyando las decisiones técnicas sobre resultados objetivos y consideraciones subjetivas en la experiencia de usuario, se detallan los bloques que componen este tomo así como aclara la justificación de la solución y el valor aportado.

Aunque en un primer momento se planteará el caso de uso con un enfoque clasificatorio, para predecir el Centro de Transformación generador a partir de las mediciones observadas, se acaba modificando la idea inicial hasta acabar mutando en un **algoritmo no supervisado de detección de valores anómalos**. Partiendo de la selección del CT, resulta interesante para las partes interesadas evaluar la posibilidad con la que ocurre un valor anómalo con respecto al resto de los puntos de operación más frecuentes.

En definitiva, se define un valor anómalo como una medida que **diverge de un patrón general común a una muestra** y para dar respuesta al reto técnico, se plantea en un primer momento reducir dimensionalmente el problema original para, en un segundo tramo, inferir los patrones internos de la distribución con ayuda de los algoritmos no supervisados. Previamente, se seleccionan cinco atributos principales (Tensión, Intensidad, Potencia Activa y Potencia Reactiva, Inductiva y Capacitiva) en la totalidad de los más de 160.000 registros. Una vez filtrados, se evalúan distintos modelos de reducción: **PCA** (*Principal Component Analysis*), **AE** (*Auto-Encoder*, en versión Simple, Multicapa y No Lineal) y **t-SNE** (*t-Distributed Stochastic Neighbor*) a partir de la capacidad que un clasificador (como es un *Random Forest Classifier*) entrenado sobre **las codificaciones** de cada modelo tiene de asociar medida con CT.

Considerando que el problema no es linealmente separable pero las regiones de puntos sí son localizadas, el clasificador no destaca por su porcentaje de éxito, pero sí ofrece **una medida objetiva y ordenable** (evaluada sobre un conjunto de Validación y de Prueba) con la que estimar la calidad de la reducción dimensional. Esta métrica, sumada a la eficiencia del entrenamiento del algoritmo, constituye la función objetivo que se pretende maximizar con la elección de modelo. Aunque los resultados de este análisis se desarrollan en 4. Resultados, cabe mencionar que el algoritmo escogido finalmente es el **PCA (n=3).**

Una vez se establecen las codificaciones del PCA, se evalúa la **varianza capturada** durante la proyección (la cual debe ser suficientemente alta para asegurar la coherencia de los resultados) y se analiza la **interpretabilidad de los vectores propios** a través de sus cargas (para comentar si la traducción entre ambos espacios es directa o compleja). Estas últimas permiten la traslación de un espacio a otro, lo que beneficia al resultado por partida doble: por un lado, se permite la visualización de los análisis dentro de la propia app, y se mantiene un conjunto de datos más denso en información con los cuales predecir los valores atípicos para un CT dado.

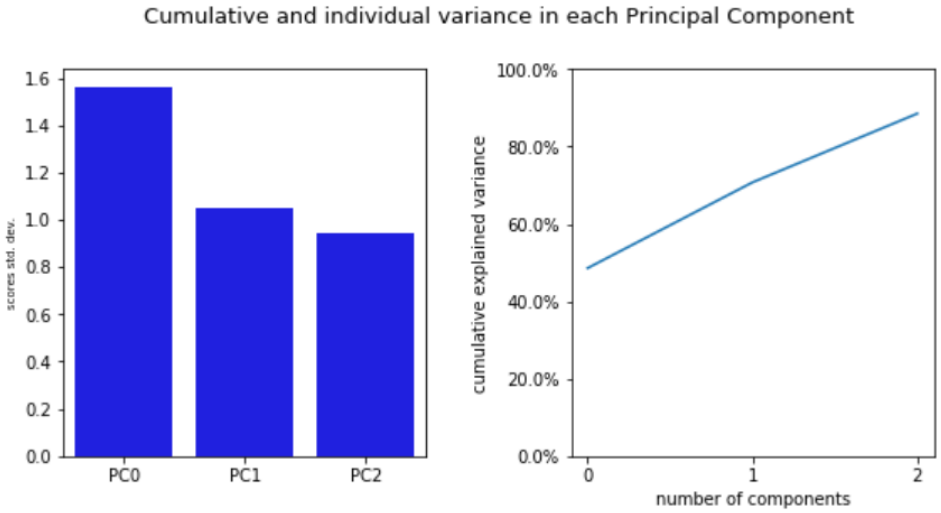


Figura 3: Varianza de codificaciones y Varianza explicada de los primeros PC

En el informe, se realiza una taxonomía de las familias de soluciones y problemas de este campo, de aplicaciones diversas como son la detección de fraude, la inspección médica o la ciberseguridad de redes. Con ayuda de la implementación de modelos de **Scikit-learn**, se comparan esta vez los resultados de los modelos **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*)**, OPTICS** (*Ordering Points To Identify the Clustering Structure*) y **LOF** (*Local Outlier Factor*)**.** Los comentarios del repaso teórico subrayan los beneficios y riesgos de cada uno de ellos, los casos de uso donde mejor se adaptan y, por el apartado práctico, se llevan a cabo los análisis de las detecciones y materializan los gráficos visualizadores que más tarde entran en juego dentro de la aplicación.

Una vez más, análisis de resultados de conclusiones, compromisos entre objetivos y conclusiones son comentadas con detalle en los **Resultados**. Sin embargo, para continuar con el desenlace en este resumen, se adelanta que el modelo utilizado por prestaciones en resultados y tiempos de carga no es otro que el modelo **DBSCAN (eps = 0.55).**

En el último y cuarto apartado cuarto, se reserva el emplazamiento para la **explicación de la integración de los dos módulos anteriores**. El desarrollo va más allá del código que se utiliza en el apartado descriptivo para profundizar más en los medios y métodos usados para **entrelazar ambas secciones** y hacer su uso intuitivo como si de una aplicación comercial se tratase, y en particular del uso privilegiado y primordial que se reserva a los **callbacks.**

Los callbacks son **funciones Python** que son automáticamente llamadas por Dash cada vez que la propiedad de algún componente cambia. Cuando la propiedad de entrada cambie, la función del decorador del callback es llamada automáticamente y Dash lanza la ejecución de la función con el nuevo valor de entrada para devolver el nuevo valor de salida. Esto es denominado ***Reactive Programming*** y, a excepción que el valor se sobrescriba manualmente, permite la interactividad de la aplicación. Los callbacks más complejos son ahora capaces de tomar varias entradas o salidas en una misma función, incluir transiciones suaves a los cambios en gráficos… y se encuentran al origen de la aplicación, desde la generación de mapas de calor o renderizados en 3D donde el usuario tiene el poder para

seleccionar la vista que mejor le convenga, a la discriminación de estilos o el sistema de pestañas implementado para poder navegar del panel descriptivo al analítico, y viceversa.

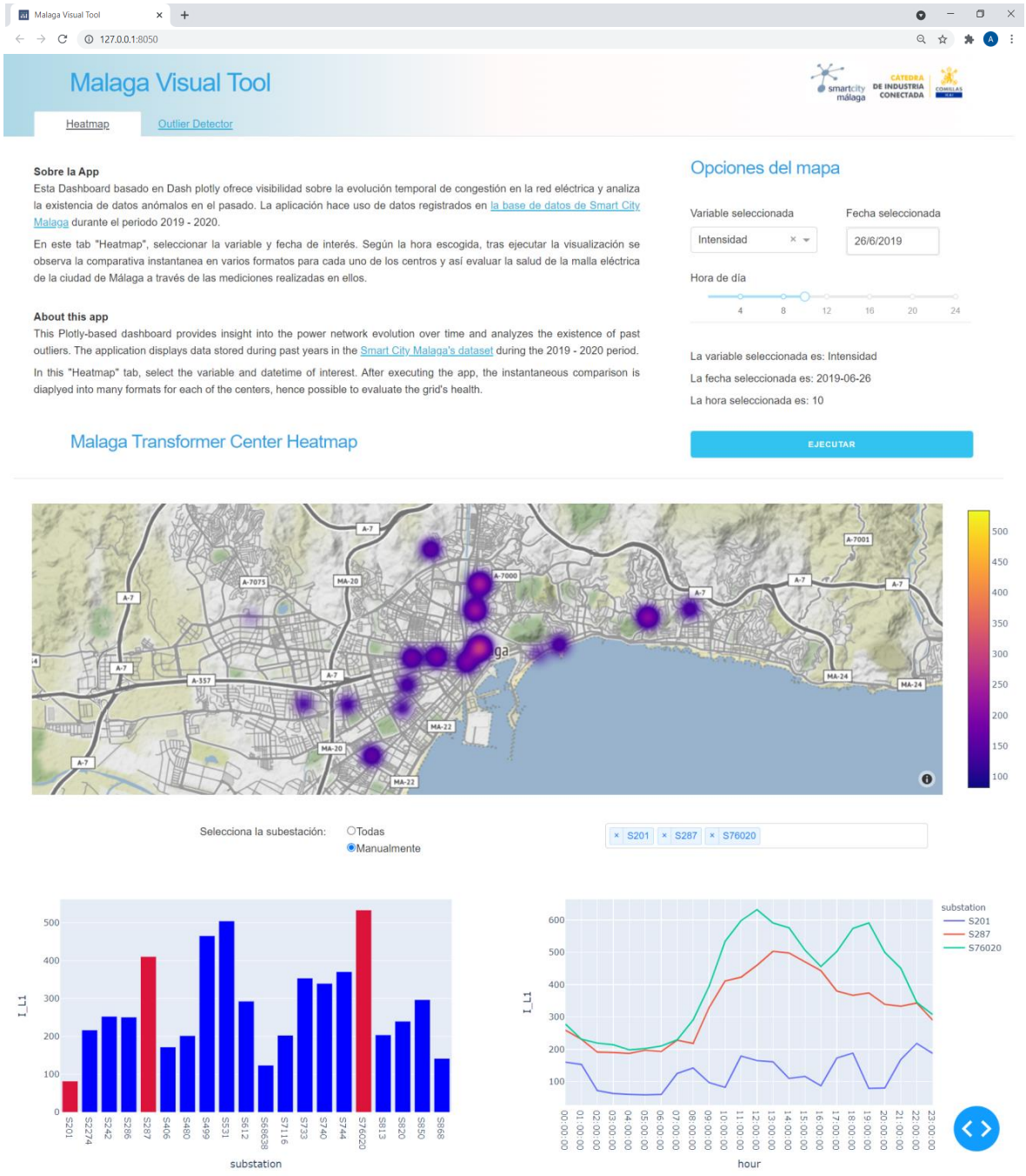


Figura 4: Presentación de la primera pestaña de la aplicación Malaga_Visual_Tool

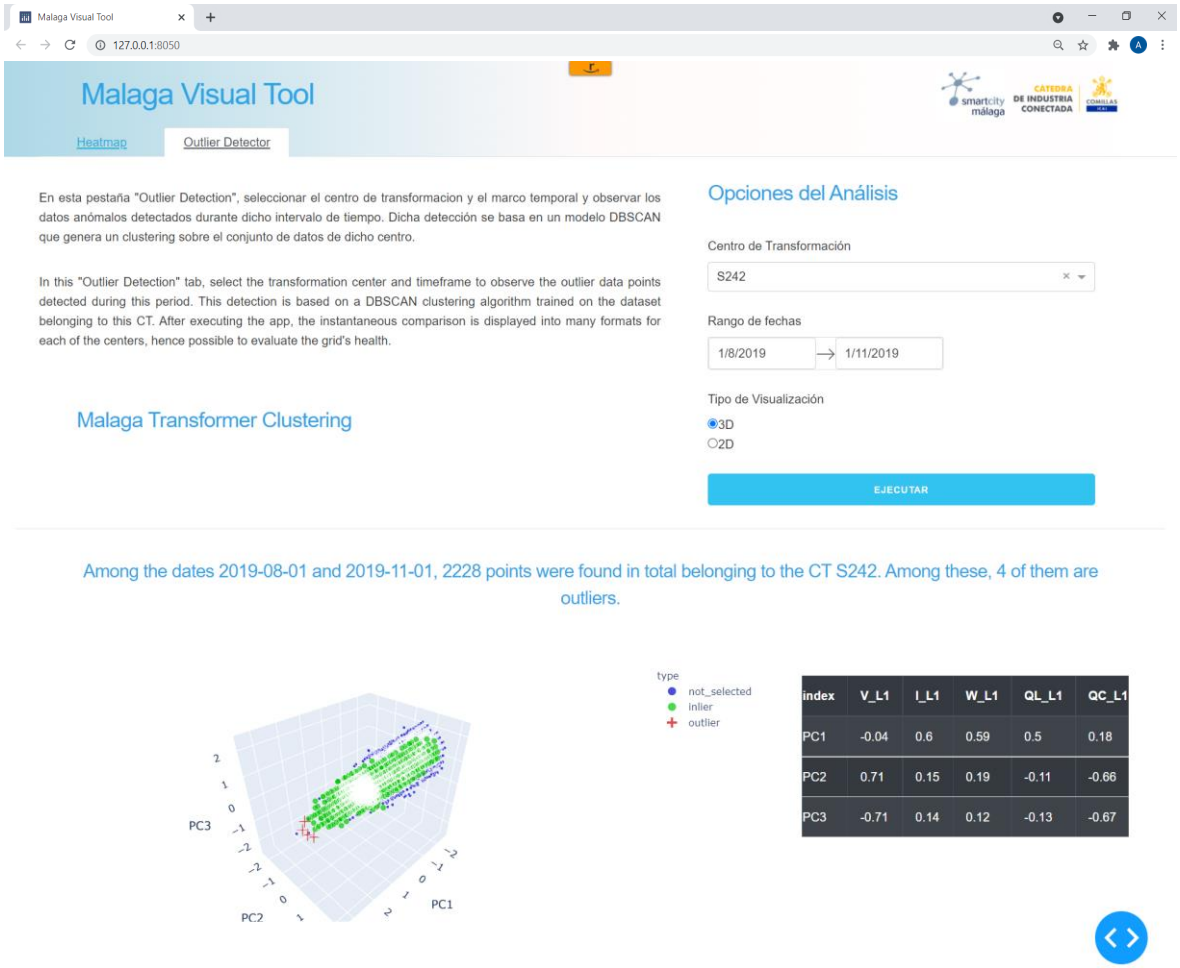


Figura 5: Presentación de la segunda pestaña de Malaga_Visual_Tool

Para acabar, el quinto título acaba enumerando el conjunto de operaciones realizadas para optimizar el uso de la herramienta, ya sea mejorando el resultado entregado en la salida o utilizando los recursos computacionales mínimos para no entrar en conflicto con la fluidez del sitio web. Entre las técnicas escogidas, destacan la elección de figuras, aunque más complejas, más flexibles y mantenibles en el tiempo, el perímetro trazado de cara a las opciones disponibles de selección para el usuario y, no menos importante, la adaptación de la plantilla y objetos al tamaño y características del visor y así abstraer el despliegue de elementos del dispositivo, consiguiendo que la **herramienta sea multiplataforma**.

4. Resultados

En este capítulo se repasan los resultados de los estudios llevados en el título anterior, tanto para la comparativa de modelos que intervienen en la reducción dimensional como en los candidatos para la identificación de valores atípicos.

En el análisis de la reducción dimensional, se valoran los resultados que el RFC arroja sobre la predicción de Centro de Transformación (CT) basado en las codificaciones de cada sistema para el conjunto de Validación y de Prueba. Valorar ambas puntuaciones tiene

interés para evaluar el **compromiso sesgo – varianza de cada modelo**, y conocer **si se da infra o sobreentrenamiento**.

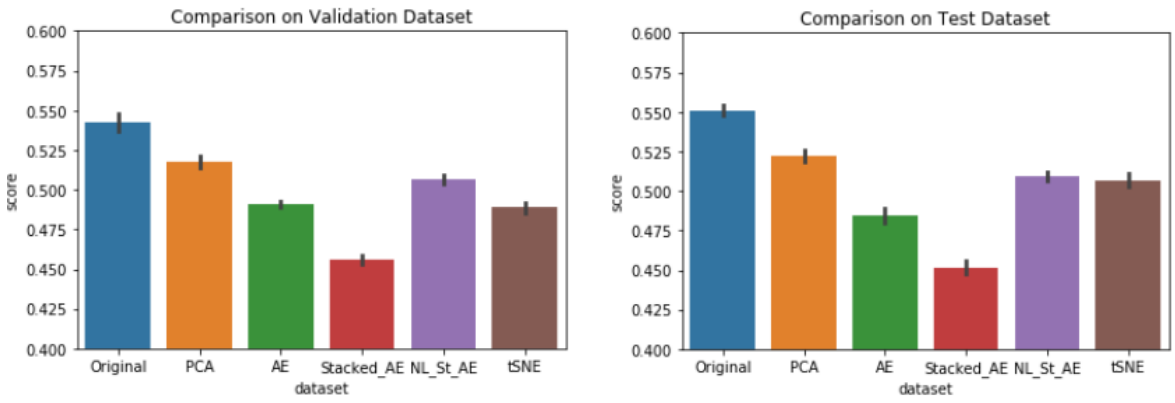


Figura 6: Resultados RFC sobre el conjunto Validation y Test

Viendo las diferencias en rendimiento de la clasificación entre el espacio original y los diferentes espacios, se observan resultados bastante concluyentes. En términos de eficiencia, los modelos más rápidos fueron el PCA (tiempo de ajuste = 3.13 s) y el AE simple (tiempo de ajuste = 2.5, optimizado con el paquete Keras).

La conclusión es que, para mantener fielmente la distribución de puntos y posibilitar su representación, la mejor opción es **reducir las dimensiones de cinco a tres por medio de un PCA**. Una vez se ha tomado como piedra angular de la transformación, es conveniente juzgar la claridad de los vectores propios y su relación con las variables de partida.

Indice	Tensión	Intensidad	Potencia Activa	Potencia inductiva	Potencia Capacitiva	Varianza Capturada
PC1	-0,04	**0.60**	**0.59**	0.5	0.18	*0.49*
PC2	**0.71**	0.15	0.19	-0,11	**-0,66**	*0.22*
PC3	**-0,71**	0.14	0.12	-0,13	**-0,67**	*0.18*

Tabla 1: Pesos de vectores propios del PCA en la base original

En esencia, el PC1 cuenta con una fuerte dependencia de **Intensidad y Potencia Activa** y registra los casos de **sobreintensidad ante desbalances de carga** (además, alberga casi el 50% de la varianza). PC2 pone el acento sobre los valores altos de **Tensión** y bajos de **Potencia Capacitiva**, fijando su atención en casos donde la tensión supera la consigna de regulación de la red. Por último, el PC3 es la tercera componente perpendicular y se fija en las bajadas de tensión fuerte (causadas por fallas de equipos suministradores o de infraestructura, encendido de grandes cargas, …).

Por acabar, el último gran bloque de análisis lo protagoniza la comparativa de modelos para la detección de valores anómalos. Para cada uno, se miden los **tiempos de entrenamientos** sobre una fracción aleatoria del conjunto y se analiza la discriminación que hacen.

MODELO	DBSCAN eps = 0.3	DBSCAN eps = 0.7	OPTICS	LOF
Tiempo entrenamiento	*1'037*	*1'911*	*1"15'60*	*00'5514*

Tabla 2: Tiempos de entrenamiento para DBSCAN, OPTICS y LOF

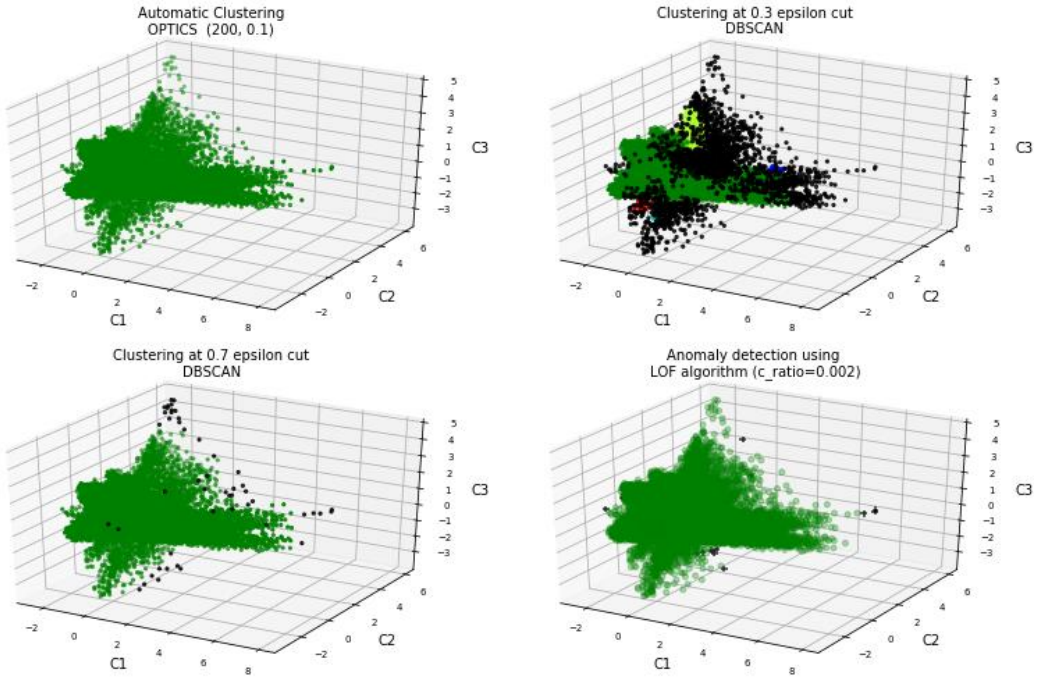


Figura 7: Resultados de detección de outliers

Entre los cuatro casos, es el modelo **DBSCAN** quien ofrece mejores prestaciones en facilidad de uso, agilidad y mejor se adapta a una más amplia gama de tamaños de entrenamiento (que variará según la selección del usuario).

5. Conclusiones

A lo largo de este proyecto, se ha conseguido procesar, explorar, extraer información y presentarla dentro de una **aplicación multiplataforma, ágil y distribuible** para el soporte y apoyo de tareas de mantenimiento de Centros de Transformación de la ciudad de Málaga.

En este apartado se incluye una guía extensiva para el despliegue de la aplicación en dos alternativas de Plataforma como Servicio (PaaS) que facilitarían la distribución del proyecto, la puesta en común y alineación de objetivos e integración en el sistema IT de la compañía.

La aplicación Dash es **holística frente a datos y dispositivos**, y consigue **generar una herramienta analítica de presentación de datos descriptivos y predictivos** alrededor de observaciones medidas sobre las líneas de centros de la red eléctrica. La **latencia** de la página web es **inferior al segundo**, presenta una **interfaz intuitiva al usuario** y responde a las problemáticas que hoy en día se hallan a la hora de explotar las extensas bases de datos

históricas con registros de las variables físicas medidas. Los objetivos iniciales se satisfacen globalmente e incluso se añaden funcionalidades que no estaban programadas en origen, dentro de un plazo menor al detallado en el ANEXO I. Diagrama de Gantt.

De cara al CIC, esta herramienta se encuentra ahora en un punto dulce donde se abren miles de ampliaciones y variaciones de adaptación para nuevos colaboradores de la Cátedra que quieran extraer información de valor con el poder de la Ciencia de Datos, muy necesario en el siglo XXI y en el futuro que queda por delante.

6. Referencias

[1] C. Yao, Z. Zhao, Y. Mi, C. Li, Y. Liao and G. Qian, "Improved Online Monitoring Method for Transformer Winding Deformations Based on the Lissajous Graphical Analysis of Voltage and Current," in IEEE Transactions on Power Delivery, vol. 30, no. 4, pp. 1965-1973, Aug. 2015, doi: 10.1109/TPWRD.2015.2418344. https://ieeexplore.ieee.org/document/7086082

[2] A. Singh and P. Verma, "A review of intelligent diagnostic methods for condition assessment of insulation system in power transformers", 2008 International Conference on Condition Monitoring and Diagnosis, 2008, pp. 1354-1357, doi: 10.1109/CMD.2008.4580520.: https://ieeexplore.ieee.org/document/4580520

[3] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 1”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-1-f5f2b7d548e5

[4] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 2”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-2-f03083f42ecf

[5] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 3”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-3-d04352b83072

SMART MONITORING AND UNSUPERVISED ANALYSIS APPLIED ON ELECTRICAL POWER GRID NETWORK'S OPERATIONAL VARIABLES

Author: García Domínguez, Andrés.
Supervisor: Gaitán Poyatos, Carlos.
Collaborating Entity: Smart City Málaga

ABSTRACT

In collaboration with the **Connected Industry Chair**, this project starts with the operational variables from the **Transformer Centers** in the city of Malaga to develop an **analytical tool for value-added information extraction** on data aggregation. The main outcome is the development of an **online dashboard application** based on the *Plotly* framework, enabling cross-platform visualization and performing deep analysis by means of dimensionality reduction techniques into a reduced latent space, then followed by a subsequent clustering prediction.

Keywords: *Dash Application, Dimensionality Reduction, Outlier Detection*

1. Introduction

Nowadays, there is a growing interest in developing **advanced control systems** over the electricity power grid that can streamline maintenance and operation of critical infrastructures in favour of the well-being of a country. As **electricity demand** grows steadily, the **complexity of the network** increases in scale and nature, and **interoperation between assets** becomes critical, **efficiency** and required **capacity** of the control systems that underlie the architecture become a limiting factor.

Among aspects that cause interest in the revolution that is taking place in the field, is the control and supervision of the system and, specifically, the responsible one for the **monitoring, control and protection of the substations**. Traditionally isolated functions are being merged to the point of being based on a smaller number of devices, but forming a system enhanced by probabilistic models, the laws of inference on latent states from measurable variables or online visualizations to perform descriptive, predictive analysis and ultimately, prescriptive.

The focus on this type of model is on the rise, which is leading to the creation of more and more robust models, as well as real-time processing applications for the inspection and construction of alarm systems around them. [1] [2]

2. Project Definition

Advanced Data Analytics and Artificial Intelligence are evolving at breakneck speed, yet there is still a long way to go. Currently, many methods emerge from the contributions born in Applied Statistics, but in many occasions, it is the part of the **implementation** where companies need a greater support. This project's aim is to **respond to the needs of Smart**

City Málaga (hereinafter also called SCM) to develop a **multiplatform web-based model in real time** providing information of high added value and making use of intelligent techniques. Thus, it is intended to create a **basis for future models** aimed at solving this type of problem or similar ones, from the assistance to personnel destined for the physical place, to the execution of alarm managers or detection of anomalies.

In terms of objectives, they have been agreed and specified in the following list:

- Achieve latency and interaction levels as of a web service (less than 1 second).
- Allow automatic generation reports over network congestion and grid health.
- Create a model as robust as possible and agnostic to the data and platform: avoid quality loss in the face of implicit or explicit variations in the input data, and that the program does not depend on the device, operating system or version of the viewer.
- Leverage the analysis and conclusions of the study on intelligent tools for reducing dimensions (less than 15% error with respect to the reconstructed space).

3. Description of the Model

The goal of the project is to develop a digital product built to allow the monitoring and temporal evolution of electrical congestion within a given geographic area. As a real-time interaction tool with flexible functionality on different type of devices, the required result conditions were clear and based on three differentiated axes: **ease of use**, **interpretability of the results** and stability of the page's integrity (both *front-end* as *back-end*).

The first section is reserved for the **preliminary qualitative and quantitative study** carried out on the starting variables. With the help of three different data evaluation methods (Sweetviz **Error! Reference source not found.**, Autoviz **Error! Reference source not fo und.** and Lux **Error! Reference source not found.** libraries), the relationships that bind the behavior of certain physical variables with others are assessed and quantified, as well as the distribution of observations within each of them.

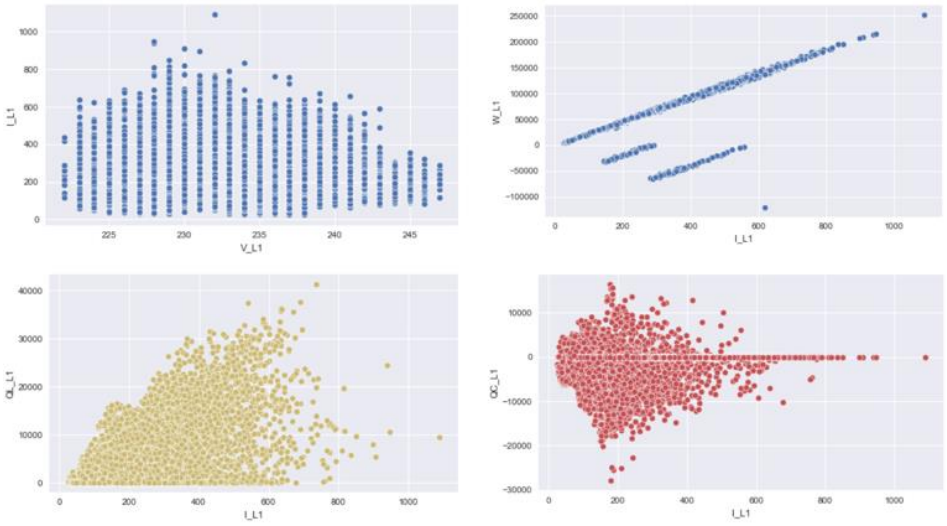


Figure 1: Data Exploratory Analysis Example with Autoviz

Based on the conclusions drawn (in particular, evaluation of the quality of the data, nature of the values and correlations between variables), a **preprocessing guide** (cleaning, substitution of values, formatting and normalization) and **selection of attributes** are decided, that ultimately replicate in the import of data inside the tool.

The second section translates the generation of results in an intermediate stage of the project's chronological evolution, where a basis is provided to define the share of the tool in charge of the most descriptive part. Beyond choosing one type of graph instead of others, above all it lays the foundations of the global structure for the Dash application template, a key factor for the progression of the work. First, the schematic design that encompasses the graphs of interest was addressed and that ultimately translates to the visualization – style – interactivity world (HTML – CSS – JavaScript).

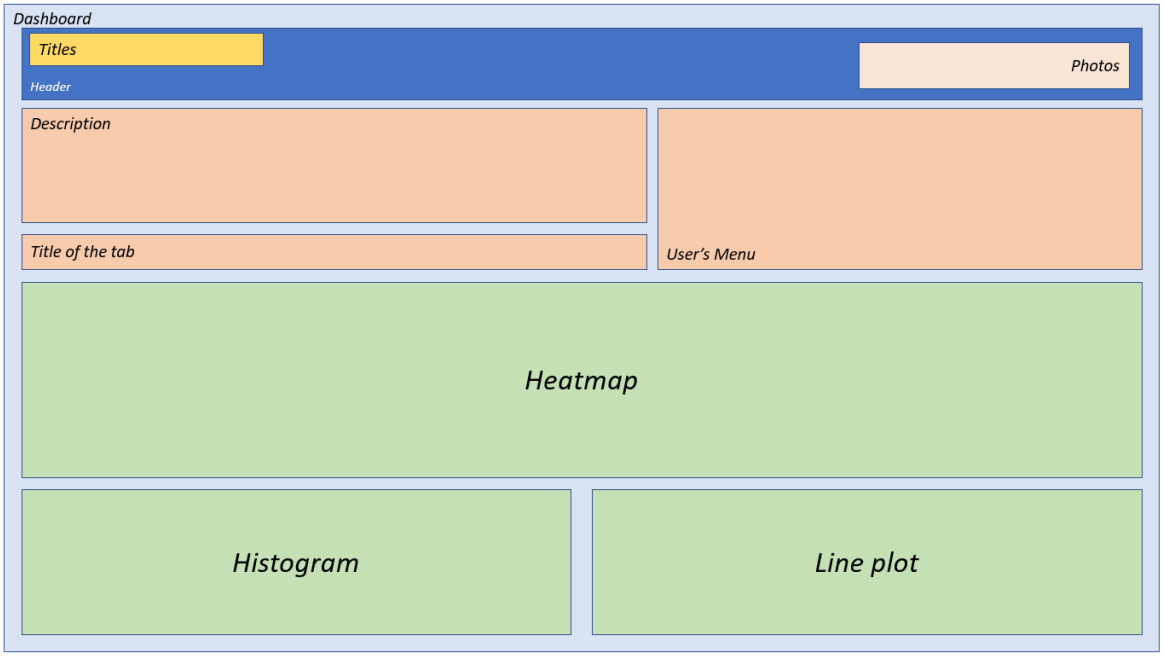


Figure 2: Dashboard's conceptual scheme

After going through several iterations to achieve the desired preprocessing and the ideal service for the geographical representation of the heat map, all performed on the Google Colab platform, a research work is done around the available objects in Dash (essentially Dash_html_components, Dash_core_components, Dash_bootstrap_components). Based on these, the proposed general structure, inspired on the best practices for application development, is as follows: 1) **Import Python libraries and frameworks**, 2) **Process the starting information**, 3) **Build the application layout**, with input and output elements inside and 4) Specify at the end the functions (called **callbacks**) that describe **existing interactions on the web page** in response of user events.

The result of the first version, mainly descriptive, is similar to the first tab Figure 4 and Figure 5.

The third section fully confronts the half of the tool in charge of the deep analysis work, and presents the exhaustive analysis work that was performed around the two machine learning problems previously evoked: **dimensional reduction**, and **outlier detection**. Supporting the technical decisions on objective results and subjective considerations with respect to user experience, it details the blocks that compose this volume as well as clarifies the justification of the solution and the value provided.

Although at first the use case was approached as a **classificatory task to predict the generating Transformer Substation** from the observed measurements, it is eventually modified to mutate into an **unsupervised outlier detection algorithm**. Ingesting the TC selection, it is interesting for the stakeholder parties to evaluate the possibility of an outlier occurring considering the rest of the most frequent operating points.

In short, an outlier is defined as a measurement that **diverges from a general pattern of a sample** and, in order to fulfill the technical challenge, the original problem is first dimensionally reduced and, in a second step, the internal patterns of the distribution are inferred with the help of unsupervised algorithms. First, **five main attributes (Voltage, Current, Active Power and Reactive Power, Inductive and Capacitive)** are selected from the total of more than 160,000 records. Once filtered, different reduction models are evaluated: **PCA** (Principal Component Analysis), **AE** (Auto-Encoder, in Simple, Multilayer and Nonlinear versions) and **t-SNE** (t-Distributed Stochastic Neighbor) based on the ability of a classifier (such as a Random Forest Classifier) trained on the encodings of each model to associate measurement with its TC.

Considering that the **problem is not linearly separable** but the regions of points are **localized**, the classifier does not stand out for its success rate, but it does provide an **objective and orderly measure** (evaluated on a **Validation and Test dataset**) from which to estimate the quality of the dimensional reduction. This metric, coupled with the training efficiency of the algorithm, constitutes the objective function that the model choice is intended to maximize. Although the results of this analysis are developed in 4. Results, it is worth mentioning that the algorithm finally chosen is the **PCA (n=3).**

Once the PCA encodings are established, the **cumulated variance** captured by the projection is evaluated (which must be high enough to ensure the consistency of the results) and the **interpretability of the eigenvectors** is analyzed through their loadings (in order to comment on whether the translation between both spaces is straight or complex). The latter allow the transformation from one world to the other, which benefits the result in two ways: on the one hand, it allows the visualization of the analyses within the app itself, and it maintains a more information-dense dataset with which to predict outliers for a given TC.

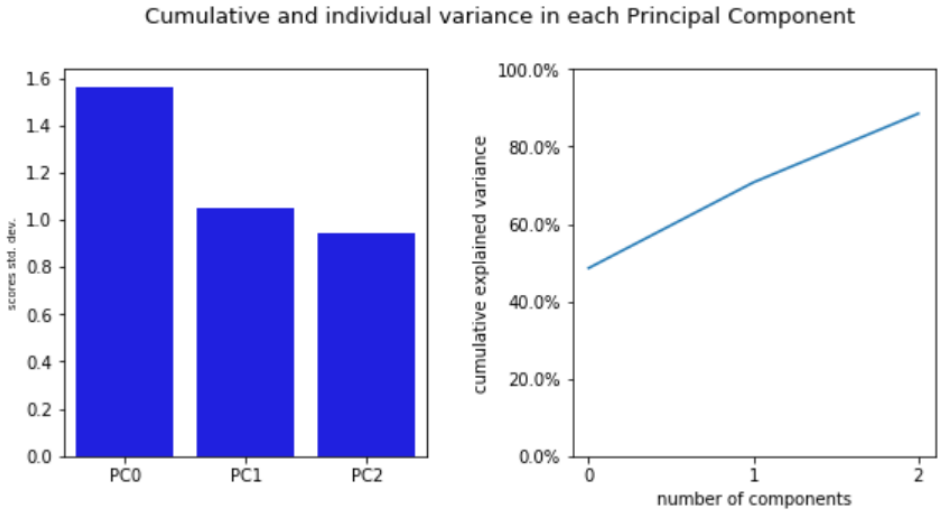


Figure 3: Loadings std dev and Explained Variance with three first PC

In the report, a taxonomy of the solutions and problems families in this field of diverse applications such as fraud detection, medical inspection or network cybersecurity. With the help of the **Scikit-learn** model implementation, the results of the **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*), **OPTICS** (*Ordering Points To Identify the*

Clustering Structure) and **LOF** (*Local Outlier Factor*) models are compared. The comments on the theoretical review underline the benefits and risks of each, the use cases to which they are best suited and, for the practical section, analysis of the detections are carried out and materialize the visualization graphs that later come into play within the application.

Once again, analysis of results of conclusions, compromises between objectives and conclusions are discussed in detail in the Results. However, in order to continue with the outcome in this summary, it is anticipated that the model used for performance in results and loading times is none other than the **DBSCAN model (eps = 0.55).**

In the last and fourth section, a space is reserved for the **explanation of the integration of the two previous modules**. The development goes beyond the code used in the descriptive section, rather going deeper into the means and methods used to **interlace both sections** and make their use intuitive as if it were a commercial application, and in particular, the privileged and primordial use reserved for **callbacks**.

Callbacks are **Python functions automatically called by Dash** whenever the property of some component changes. When the input property mutates, the callback decorator function is automatically invoked and Dash launches the execution of the function with the new input value to return the new output. This is called **Reactive Programming** and, unless the value is manually overwritten, it enables application interactivity. Advanced callbacks are nonetheless able to take several inputs or outputs in the same function, include smooth transitions to changes in graphics... and are at the origin of the application, from the generation of heat maps or 3D renderings where the user has the power to select the view that best suits him, to the discrimination of styles or the tab system implemented to navigate from the descriptive to the analytical panel, and vice versa.

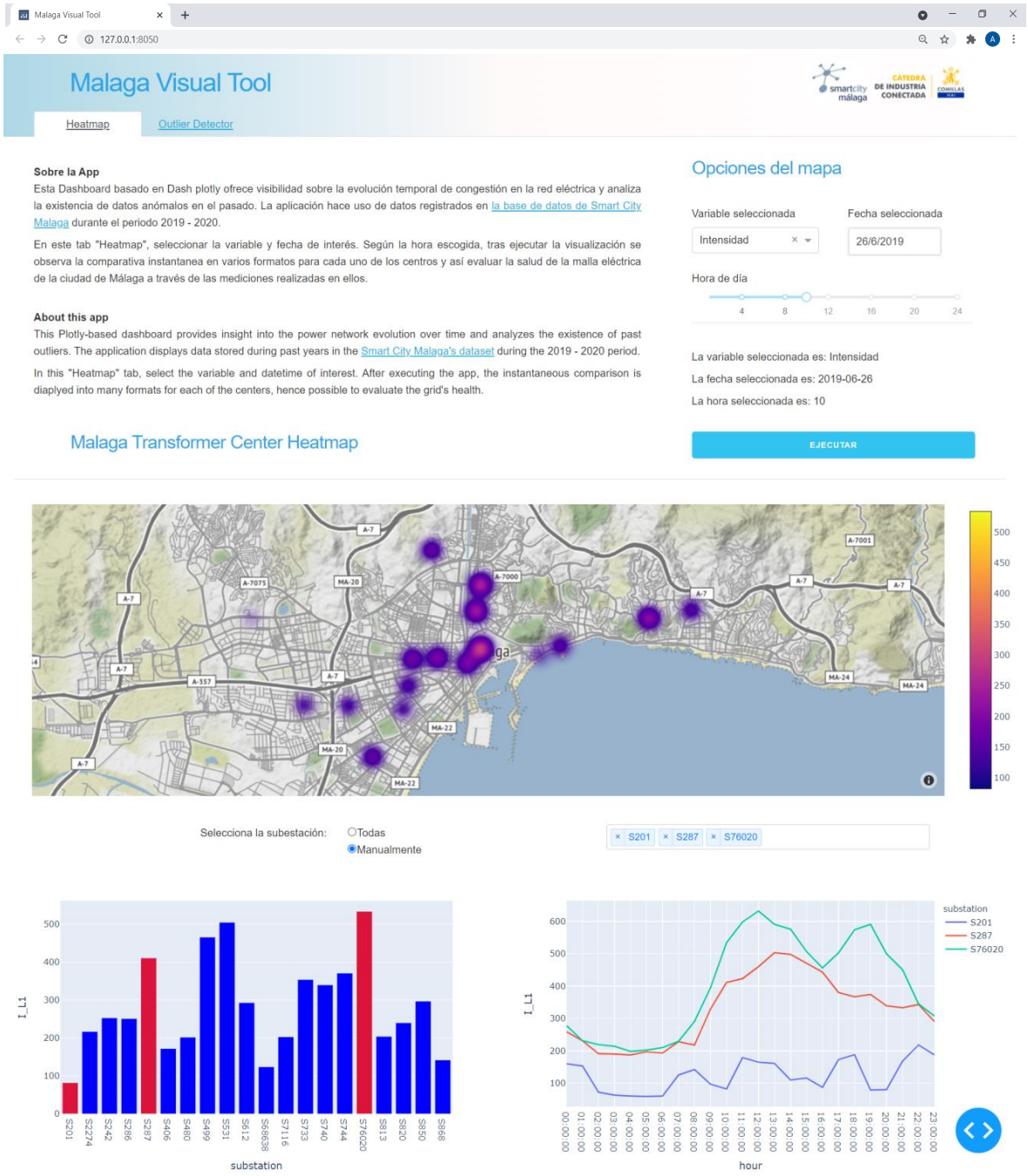


Figure 4: First tab presentation of the Malaga_Visual_Tool

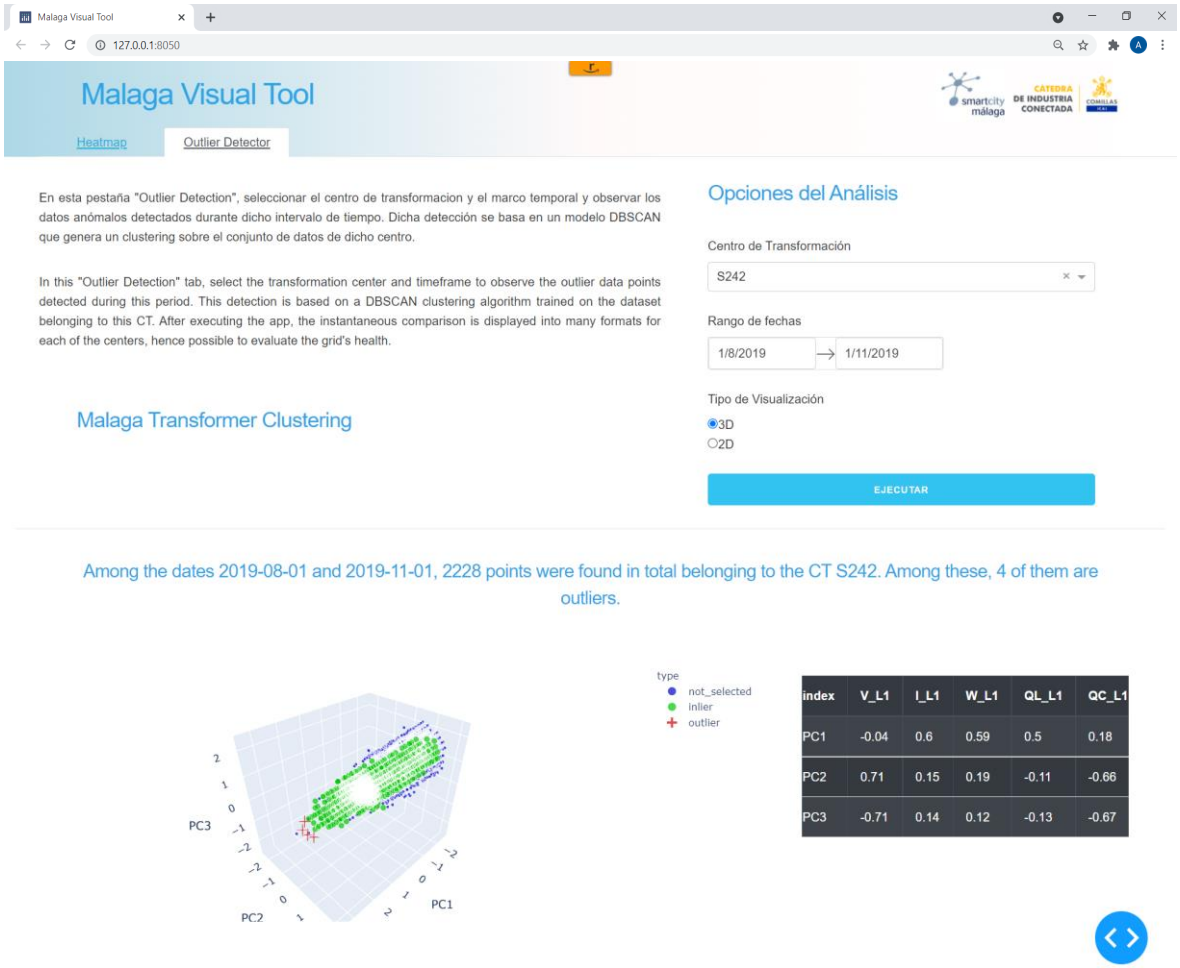


Figure 5: Second tab presentation of the Malaga_Visual_Tool

Finally, the fifth chapter ends up listing the set of operations performed to optimize the use of the tool, either by improving the result delivered in the output or by using the minimum computational resources so as not to conflict with the fluidity of the website. Among the chosen techniques, we highlight the slight adaptation of figures, despite being more complex, gain flexibility and maintenance over time; the perimeter traced in the face of the selection options available to the user and, last but not least, the adaptation of the template and objects to the size and characteristics of the device's screen, thus abstracting the display of elements of the device and **making the tool cross-platform**.

4. Results

This section reviews the results of the studies mentioned in the previous chapters, both for the comparison of models involved in dimensional reduction and the candidates for outlier identification.

In the analysis of dimensional reduction, the results that the RFC yields on Transformation Centre (TC) prediction based on the codings of each system are assessed, both for the

Validation and Test sets. Evaluating both scores is of interest to evaluate the **bias-variance trade-off** of each model, and to know if there is **under or overtraining**.

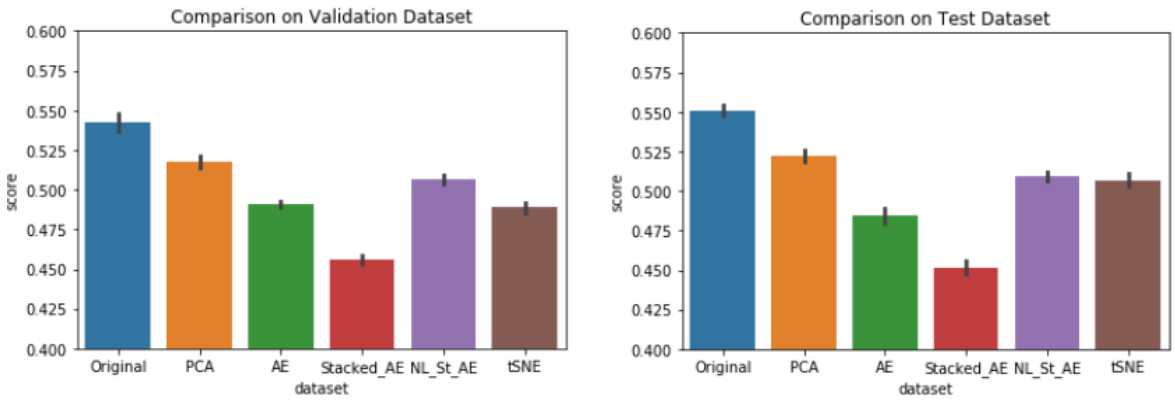


Figure 6: RFC's Cross-validation and Test results

Looking at the differences in classification performance between the original space and the different others, discussions are quite conclusive. In terms of efficiency, the fastest models were the PCA (fitting time = 3.13 s) and the simple AE (fitting time = 2.5, optimized with the Keras package).

The conclusion is that, in order to faithfully maintain the point distribution and enable its representation, the best option is to **reduce the dimensions from five to three by means of a PCA**. Once this has been taken as the cornerstone of the transformation, it is convenient to judge the clarity of the eigenvectors and their relation to the starting variables.

Indice	Tensión	Intensidad	Potencia Activa	Potencia inductiva	Potencia Capacitiva	Varianza Capturada
PC1	-0,04	**0.60**	**0.59**	0.5	0.18	*0.49*
PC2	**0.71**	0.15	0.19	-0,11	**-0,66**	*0.22*
PC3	**-0,71**	0.14	0.12	-0,13	**-0,67**	*0.18*

Table 1: PCA eigenvectors' loadings

Essentially, PC1 has a strong dependence on **Current and Active Power** and catches the cases of **overcurrent in the event of load unbalance** (it also accounts for almost 50% of the variance). PC2 emphasizes **high Voltage and low Capacitive Power values**, focusing its attention on cases where the voltage exceeds the network regulation setpoint. Finally, PC3 is the third perpendicular component and focuses on **strong voltage dips** (caused by failures of supply equipment or infrastructure, switching on of large loads, ...).

Finally, the last major block of analysis is the comparison of models for the detection of outliers. For each one, **training times** are measured on a random fraction of the set and their discrimination is analyzed.

MODELO	DBSCAN eps = 0.3	DBSCAN eps = 0.7	OPTICS	LOF
Tiempo entrenamiento	*1'037*	*1'911*	*1"15'60*	*00'5514*

Table 2: Training times for DBSCAN, OPTICS and LOF

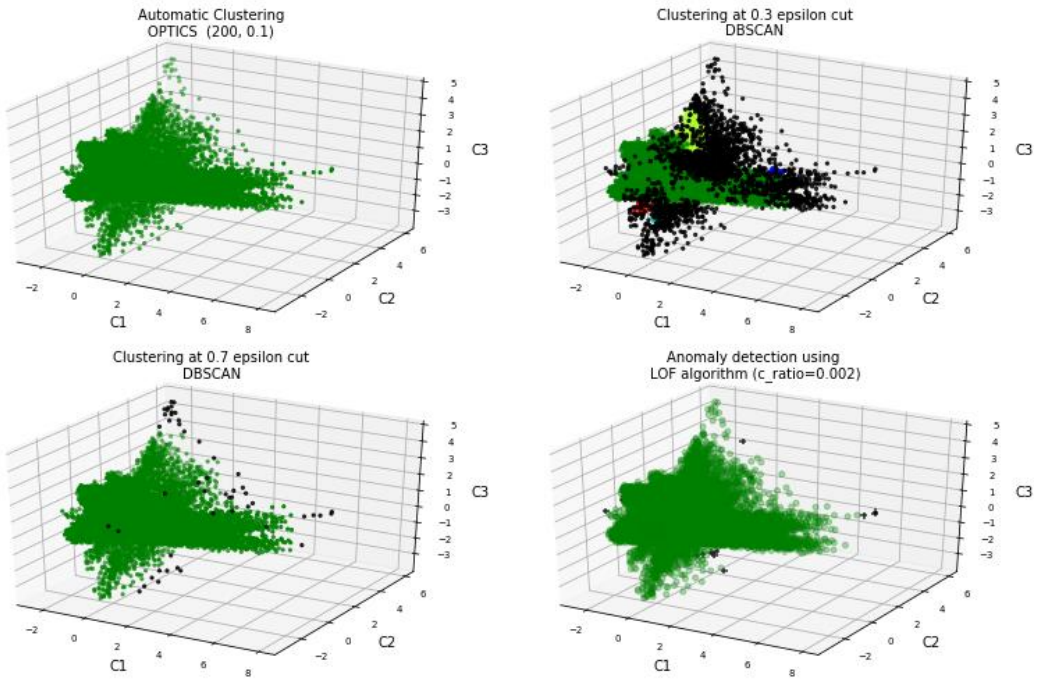


Figure 7: Outlier Detection results

Among the four cases, it is the **DBSCAN** model the one that offers the best performance in terms of ease of use, agility and better adaptability to a wider range of training sizes (which will vary according to the user's selection).

5. Conclusions

Throughout this project, it has achieved to process, explore, extract information and present it in a **multiplatform, agile and distributable application** for the support and support of maintenance tasks of Transformer Substations in the city of Malaga.

This section includes an extensive guide for the deployment of the application in two Platform as a Service (PaaS) alternatives that would facilitate the distribution of the project, the sharing and alignment of objectives and integration into the company's IT system.

The Dash application is **holistic to data and devices**, and manages to **generate an analytical tool for presenting descriptive and predictive estimations** around observations measured on the power grid lines. The web page latency is **less than one second**, it presents an **intuitive user interface** and responds to the problems encountered today in exploiting large historical databases with records of measured physical variables. The initial objectives

are satisfied globally and even functionalities that were not originally programmed are added, within a shorter time frame than detailed in ANEXO I. Diagrama de Gantt

With regard to the CIC, this tool is now in a sweet status where thousands of extensions and adaptation variations are open for new collaborators of the Chair who want to extract valuable information with the power of Data Science, very necessary in the XXI century and in the future that lies ahead.

6. References

[1] C. Yao, Z. Zhao, Y. Mi, C. Li, Y. Liao and G. Qian, "Improved Online Monitoring Method for Transformer Winding Deformations Based on the Lissajous Graphical Analysis of Voltage and Current," in IEEE Transactions on Power Delivery, vol. 30, no. 4, pp. 1965-1973, Aug. 2015, doi: 10.1109/TPWRD.2015.2418344. https://ieeexplore.ieee.org/document/7086082

[2] A. Singh and P. Verma, "A review of intelligent diagnostic methods for condition assessment of insulation system in power transformers", 2008 International Conference on Condition Monitoring and Diagnosis, 2008, pp. 1354-1357, doi: 10.1109/CMD.2008.4580520.: https://ieeexplore.ieee.org/document/4580520

[3] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 1”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-1-f5f2b7d548e5

[4] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 2”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-2-f03083f42ecf

[5] Himanshu Sharma, “Automating Exploratory Data Analysis – Part 3”, Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-3-d04352b83072

Índice

Sección 1. INTRODUCCIÓN 7

1.1 Contexto global: el sector eléctrico 7

1.2 Contexto particular del Proyecto 11

1.3 Motivación 13

1.4 Objetivos 13

1.5 Metodología 15

1.6 Recursos empleados 16

Sección 2. Descripción de las tecnologías 17

2.1 Estado de la cuestión: contexto eléctrico 17

2.2 Ciencias impulsoras frente al desafío técnico 19

2.3 Estándares prácticos: marcos de trabajo 23

2.3.1 Desarrollo web: Plotly frente al paradigma tradicional 23

2.3.2 HTML 25

2.3.3 CSS 25

2.3.4 JavaScript 26

2.3.5 Dash 27

2.3.6 Inteligencia artificial: scikit-learn 28

Sección 3. Definición del problema 33

3.1 Descripción del caso de estudio y requerimientos 33

3.2 beneficios y riesgos 37

3.3 Alcance del trabajo, presupuesto y estado de partida 38

Sección 4. Implementación de las soluciones 43

4.1 Análisis exploratorio de datos 44

4.1.1 Análisis preliminar 45

4.1.2 Librería Sweetviz [25] 50

4.1.3 Librería Autoviz [26] 55

4.1.4 Librería Lux [27] 58

4.1.5 Preprocesamiento y selección de atributos 59

4.2 Herramienta de visualización descriptiva 61

4.2.1 Importación de librerías y preparación de datos *65*
4.2.2 Construcción de la plantilla *66*
4.3 Análisis agregativo y detección de valores anómalos 71
4.3.1 Estudio de Modelos para la Reducción dimensional *78*
4.3.2 Estudios de modelos para la detección de valores anómalos *86*
4.4 Integración e interacción con la aplicación 92
4.5 Tareas de optimización 97

Sección 5. Resultados y Análisis 100
5.1 Análisis de la Reducción Dimensional 100
5.1.1 Interpretabilidad del PCA *103*
5.2 Análisis del Detector de Valores anómalos 104

Sección 6. Conclusiones y Futuros pasos 108
6.1 Resultados del proyecto 108
6.2 Distribución de la herramienta 109
6.3 Futuros pasos y ampliaciones 111

Sección 7. Bibliografía 112

ANEXO I. Diagrama de Gantt 116

ANEXO II. Reflexión sobre los o.d.s. 117

ANEXO III. Malaga Visual Tool, Versión inicial 118

ANEXO IV. Malaga Visual Tool, Versión Final 119

ANEXO V. Instrucciones para el despliegue de la aplicación Dash 122

Índice de Figuras

Figura 1: Esquema simplificado del sistema de suministro eléctrico [2] 8
Figura 2: Avance de tropas napoleónicas hacia Moscú, de Charles Joseph Minard [15] ... 20
Figura 3: Ejemplo de PCA en 2D [16] 21
Figura 4: Representación esquemática de AE, y diferencia linealidad frente al PCA [17]. 22
Figura 5: Diferencia entre HTML plano y HTML+CSS 26
Figura 6: Frameworks de IA desarrollados por Google (TF), Facebook (Pytorch) o ONEIROS (Keras) 29
Figura 7: Ejemplo de Clasificación y Regresión 31
Figura 8: Estéticas de manuales de usuario entre una máquina de escribir de los 80 y un teclado Mac de hoy 34
Figura 9: Nuevas formas de publicidad en el siglo XXI (caso de Coca-Cola) 35
Figura 10: Ejemplo del visor retransmitiendo deporte en directo en la app de YBVR [21] 36
Figura 11: Ejemplo de gráfico Boxplot de tensión en linea 1 para cada CT 40
Figura 12: Primera Componente Principal 41
Figura 13: Trazado del conjunto de datos 41
Figura 16: Esquema ilustrativo de los componentes de potencia y energía [24] 47
Figura 17: Informe explorador de LVSM_Def según Sweetviz 51
Figura 18: distribución de valores en V_L1 y App_SW 52
Figura 19: Comparativa 2D entre datos del CT S201 y el resto 53
Figura 20: Comparativa entre tensiones en Sweetviz 53
Figura 21: Matriz de asociación entre variables físicas medibles 54
Figura 22: Ejecución del generador de gráficos (n = 406) 55
Figura 23: Gráficos Autoviz 56
Figura 24: Histograma de Intensidad en Linea 1 57
Figura 25: Histograma y Distribución de Potencia Activa en linea 1 57
Figura 26: Gráficas Lux 58
Figura 27: Esquema conceptual del tablero 61
Figura 28: Gráfica Heatmap con Plotly Express 62

Figura 29: Generación Python y llamada a API GMaps 63
Figura 30: Heatmap GMaps 64
Figura 31: Construcción de la app y servidor 66
Figura 32: Lanzamiento del servidor que soporta el Dash 66
Figura 33: Explorador de temas de Bootstrap y de CELURIAN en el proyecto [30] 67
Figura 34: Extracto del archivo css de la carpeta assets 67
Figura 35: Ejemplo de árbol de divisiones en la plantilla 69
Figura 36: Constructores Dash de una imagen (1), de un Menú Desplegable (2), de un Botón (3) y de dos gráficos alineados horizontalmente (4) 70
Figura 37: Valores de tensión el 26/06/19 desde la app 72
Figura 38: Reparto Train - Validation – Test 74
Figura 39: Varianza de atributos de entrada 75
Figura 40: Normalización de datos de entrada 76
Figura 41: Mapa de Calor de Correlación 76
Figura 42: Esquema ilustrando el principio de funcionamiento del AE [31] 77
Figura 43: Entrenamiento y Codificado de datos de entrada 78
Figura 44: Evolución del Error en X_Tr y X_valid (1) y Modelo del Auto-Encoder (2) ... 79
Figura 45: Representación de los dos primeros códigos del espacio latente junto con el CT como leyenda 79
Figura 46: Transformación inversa de las proyecciones frente a originales 80
Figura 47: Comparativa entre codificaciones PCA y AE en las primeras componentes 80
Figura 48: Comparativa de varianza en PCA y AE Simple (1) y Mapa de Correlación entre componentes (2) 81
Figura 49: Representación de las codificaciones en PCA y en Stacked_AE 82
Figura 50: Diferencias en funciones de activación lineal y no lineal 82
Figura 51: Comparativa de codificaciones entre PCA y NL_St_AE 83
Figura 52: Implementación del t-SNE con Scikit-learn 84
Figura 53: Codificación de t-SNE (N = 20.000) 85
Figura 54: Varianza explicada por PC0, PC1 y PC2 86
Figura 55: Explicación visual del principio de funcionamiento de DBSCAN [34] 89

Figura 56: Consecuencia práctica sobre el clustering de DBSCAN y OPTICS para grupos de densidad variable 90
Figura 57: Gráfico de Alcanzabilidad de PCA_codings (N = 20.000) 91
Figura 58: Representación de LOF scores según estimador de densidad [36] 92
Figura 59: Ejecución simple de un callback 93
Figura 60: Ejecución de callback con Estado 95
Figura 61: Definición y callback del sistema de pestañas 96
Figura 62: Implementación del mapa de calor con plotly.express y plotly.graph_objects . 98
Figura 63: Diferencia por activación de viewport en la creación de la aplicación [38] 99
Figura 64: Problema de Generalización del modelo y diferencia en errores de Entrenamiento y Prueba [39] 101
Figura 65: Ejecución de Benchmark sobre el conjunto Prueba 102
Figura 66: Resultados del RFC sobre Validation 102
Figura 67: Resultados del RFC sobre Test 102
Figura 69: Resultados de detección de valores anómalos con los cuatro modelos 105
Figura 70: Visualización del detector de valores anómalos 107

***Nota**: las figuras sin referencia son propias o de autor desconocido*

Índice de Tablas

Tabla 1: Tabla Listado_Trafos [23] 45
Tabla 2: Encabezado de Tabla LVSM_Def [23] 46
Tabla 3: Encabezado de Dataframe LVSM_Def 49
Tabla 4: Análisis describe() de la tabla LVSM_Def (1) 49
Tabla 5: Análisis describe() de la tabla LVSM_Def (2) 50
Tabla 6: Tabla recapitulativa de los métodos de Clustering en sklearn 88
Tabla 7: Pesos de los vectores propios del PCA en la base original 104
Tabla 8: Tiempos de entrenamiento para DBSCAN, OPTICS y LOF 105
Tabla 9: Volumen de detección de observaciones atípicas 106

Sección 1. INTRODUCCIÓN

1.1 CONTEXTO GLOBAL: EL SECTOR ELÉCTRICO

En la actualidad, es creciente el interés que existe por **desarrollar sistemas avanzados de control** sobre la red eléctrica que puedan agilizar las tareas de mantenimiento y operación de las infraestructuras críticas para el bienestar de un país. Conforme la **demanda eléctrica** crece decididamente, la **complejidad de la red** aumenta en escala y naturaleza, y la **interoperación entre activos** se vuelve crítica, mayor es la eficacia y capacidad que se exige a los sistemas de control que subyacen dentro de la arquitectura. A esta evolución orgánica de la red eléctrica se suman nuevas tendencias nada despreciables, como son la llegada de las ***Smart Grids***, el refuerzo de las **conexiones internacionales** y la variabilidad en origen que existe durante la **generación energética** (en particular, con la presencia creciente de energías renovables y de tecnologías de almacenamiento).

Para comprender el trasfondo de la cuestión, es conveniente remontarse a los fundamentos de las redes de distribución hoy en día y para ello, es necesario conocer también su propósito en el producto más elemental: la **electricidad**. La electricidad es hoy la **forma de energía más utilizada en todas las actividades humanas, industriales, comerciales y domésticas**. Es un instrumento relativamente fácil de producir en grandes cantidades, de transportar a largas distancias, de transformar en otros tipos de energía y de consumir de forma aceptablemente limpia, lo cual la convierte en la fuente de **energía secundaria predominante a nivel mundial** (aunque existen otras en auge, como es el hidrógeno). Esta electricidad se dice secundaria pues es el resultado de las **transformaciones de las fuentes de energía primaria** (como son las centrales térmicas, nucleares o parques eólicos, por ejemplo), quienes aprovechan elementos presentes en la naturaleza, pero carentes de un tratamiento específico que los vuelva eficaces y manejables. [1]

A pesar de su importancia, su historia es **relativamente reciente**, ya que el inicio de la tecnología eléctrica se sitúa a finales del siglo XIX, apoyada en la base científica de la

electricidad y el magnetismo. Sin embargo, el sistema que suministra la electricidad al usuario es altamente complejo e incluye el conjunto de medios y elementos útiles para la **generación**, **transporte** y **distribución** de la energía. Este conjunto está dotado de **mecanismos de control**, seguridad y protección en un sistema integrado que, a su vez, está regulado por un **sistema de control central** que garantiza la explotación de los recursos para igualar la carga de consumo (ante la imposibilidad de almacenarla a nivel industrial), a la vez que asegura una **calidad de servicio** acorde, compensando posibles incidencias, fallas producidas y fluctuaciones imprevistas durante planificación.

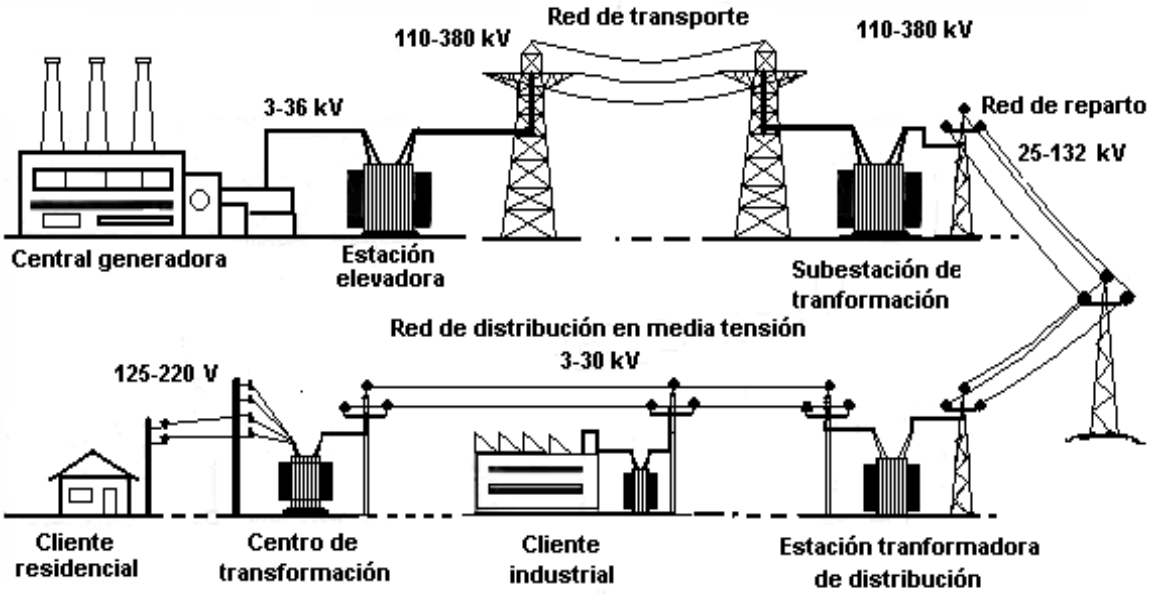


Figura 8: Esquema simplificado del sistema de suministro eléctrico [2]

Dentro de las grandes transformaciones acontecidas en estos momentos, destaca la transición que se está dando entre **sistemas de generación centralizados a distribuidos**. Tradicionalmente, el esquema seguía una forma de pirámide donde instalaciones con sistemas basados en pares de **turbina – alternador** para proveer energía a grandes superficies, las cuales eran cubiertas por la red de distribución que por capilares llegaban hasta el consumidor final. Sin embargo, con la llegada de los suministros distribuidos, la energía es ahora recolectada en centrales, pero **también en muchos de los propios nodos**

de la red que hasta ahora habían sido solo consumidores (dando la figura conocida hoy como *prosumidor*). De esta manera, el rol pasivo del usuario acaba siendo remplazado por un rol más proactivo que no sólo es generador de su propio consumo (*autoconsumo*), sino que es capaz almacenar energía para consumirla más tarde o incluso, verter este superávit energético de vuelta a la red para ser vendido a precio de mercado en ese momento, haciendo así de la red un canal bidireccional. [3] Supone así la **inversión de la estructura de la red**: de unos pocos grandes generadores, a muchos de pequeño tamaño (llamados en la literatura *Distributed Energy Resources*).

A nivel de **transporte**, la red es también encargada de enlazar los puntos de generación de la electricidad con aquellos de utilización. Para evitar ineficiencias y optimizar el reparto, las agrupaciones de puntos del sistema se interconectan según una **estructura de forma mallada** a distintos niveles: regional, nacional, europeo… Con esta arquitectura se persigue reducir las pérdidas en el transporte, conectar por varios caminos puntos alejados y permitir distintos puntos de operación igual de óptimos, fortaleciendo así un sistema de por sí robusto.

Por último, la **distribución** es la encargada de habilitar el uso de la energía una vez ha sido transportada. Para ello, el operador del sistema eléctrico (en España, R.E.E.) y las compañías suministradoras (conocidas como distribuidoras) han de construir y mantener no solo las líneas necesarias para llegar a centros urbanos de clientes, sino también de establecer las **subestaciones** que permitan reducir la tensión hasta valores utilizables de manera segura y controlada.

A pesar de las problemáticas visibles ya sobre lo comentado anteriormente, se suman factores de mayor variabilidad: la coexistencia de sistemas monofásicos y trifásicos, el balance generación-consumo ante picos de demanda o limitaciones de capacidad, líneas aéreas y subterráneas, transformaciones entre alterna y continua… Las configuraciones posibles son muy numerosas y por ello, es vital la importancia que cobran los **sistemas de control en sus distintos niveles**. Ellos regulan los puntos de operación en los nodos finales, los cuales han sido dimensionados para hacer frente unas **constantes de operación** (como son la tensión de salida y la frecuencia) **determinados por el marco regulatorio vigente**.

Entre las diversas características deseadas dentro de un sistema, destacan la **continuidad de suministro**, encargada de garantizar un mínimo nivel de Interrupción de Alimentación, y el **mantenimiento de frecuencia y tensión** (llevados a cabo gracias a la regulación primaria, secundaria y terciaria, cada cual actuando en distintos horizontes temporales). [4]

Una de las vertientes de este control es la responsable de la **monitorización, control y protección de las subestaciones**. La arquitectura ha cambiado en los últimos diez años desde múltiples dispositivos electromecánicos a todo un espectro de I.E.D. (*Intelligent Electronic Devices*) que son los pilares que sustentan nuevas lógicas de control. [5] Estos sistemas han seguidos caminos muy diversos según el país o momento de aplicación, pero se pueden destacar tres grandes etapas: el **periodo preprocesador**, el **periodo del microprocesador** y el **periodo de los dispositivos multifuncionales**, actualmente vigente.

En la primera época, se definieron las funciones que debían ser aseguradas para contar con una operación autónoma del subsistema y establecieron los primeros protocolos para hacerlo posible. En la era temprana del microprocesador, trajeron una gran revolución al mejorar drásticamente el comportamiento y calidad de los elementos electromecánicos, así como dio inicio al desarrollo gradual de dispositivos multifunción. En la actualidad, se desarrollan soluciones que usan poderosos **dispositivos multifunción acoplados a un *software* SCADA** en la subestación maestro. Se ha visto como funciones tradicionalmente aisladas se fusionan hasta basarse en un menor número de dispositivos, pero conformando un sistema mejorado por los modelos probabilísticos, las leyes de inferencia sobre estados latentes a partir de variables medibles o las visualizaciones online para realizar análisis **descriptivo**, **predictivo** y en última instancia, **prescriptivo**.

El interés por este **tipo de modelos va en aumento**, lo cual está llevando a la creación de cada vez más modelos y más robustos, así como de **aplicaciones de procesamiento en tiempo real** para la **inspección y la construcción de sistemas de alarma** alrededor de estos. Entre muchos ejemplos, destacan los ejemplos de lógica difusa para la prevención de daños por cortocircuito debido a deformaciones en el cableado [6] o bien el uso de métodos de diagnóstico inteligentes para la evaluación de daños en los sistemas de aislamiento de los

centros de transformación a partir del análisis de contaminantes en los aceites (técnica DGA – *Dissolved Gas Analysis*). [7]

1.2 CONTEXTO PARTICULAR DEL PROYECTO

Este Proyecto tiene lugar en el primer semestre del Segundo Año del Máster de Ingeniería Industrial que cursó el autor del documento durante el año 2020 -2021. Dentro de este periodo, se lleva a cabo unas Beca de Investigación en el seno de la Cátedra de Industria Conectada (C.I.C.) de la Universidad ICAI Pontificia Comillas a partir del mes de Noviembre.

Dentro de este contexto, el rol principal de un estudiante investigador es trabajar bajo el nombre de la Universidad para dar respuesta a retos prácticas a los cuales las compañías privadas y públicas encuentran en su realidad operativa, estratégica o comercial. Entablando relaciones con equipos internos de la empresa por un lado, y con investigadores propios del mundo académico por otro, el alumno debe identificar oportunidades, desarrollar casos de negocio sólidos y de impacto para la entidad y posibilitar acciones concretas para la consecución de objetivos de las prácticas. A inicios de curso, los alumnos candidatos que deseen unirse a las filas de los investigadores de la universidad entran en comunicación con los promotores de los proyectos que se han programado para el año que quede por delante y, en base a las necesidades de la misión, la disponibilidad del estudiante y sus intereses, prioriza una serie de horas semanales que irán destinadas a la persecución de la solución técnica, ya sea en grupo o en equipos individuales.

En orden cronológico, los primeros meses fueron destinados para la comprensión del problema y familiarización con las acciones realizadas por equipos anteriores que hubieron ya abordado el proyecto. A partir de ese momento, se llevó a cabo un trabajo progresivo del producto digital, construyendo mejoras en sucesivas iteraciones sobre un modelo base. Este ritmo continuo de trabajo se intercalaba a su vez de sesiones de intercambio entre alumnos

y partes implicadas en el proyecto para comentar los avances realizados y reorientar la dirección del proyecto si fuera necesario. Entre estos grupos de interés, destacan particularmente los tutores del proyecto en ICAI, José Portela González – doctor investigador del Instituto de Investigación Tecnológica (I.I.T.) y Álvaro López López – Profesor Investigador del I.I.T. y Coordinador de C.I.C., y por otro lado Carlos Gaitán Poyatos, Responsable de Innovación en Endesa, quien fue el punto de contacto con la entidad empresarial.

A lo largo del primer semestre, se realizó el trabajo de desarrollo *Software* en paralelo con los estudios universitarios hasta el mes de Febrero, cuando se presentaron los resultados de esta primera fase de estudio y se relegó el trabajo en curso a equipos que pudiesen continuarlo. Los resultados de este trabajo convergieron en la implementación de la versión preliminar del tablero de mandos que conseguía integrar, procesar, analizar y presentar los datos brutos en visualizaciones amigables al usuario que, aunque básicas, superaban la barrera de aprendizaje del módulo Plotly y marcaba un punto dulce a partir del cual las posibilidades de generación de informes particulares a otras empresas se multiplicaban.

Con el segundo semestre, llega simultáneamente la segunda mitad de este trabajo. Durante las prácticas internacionales curriculares del Doble Máster de Industria e Ingeniería Industrial del autor, se retoma a partir de Abril el proyecto desde Luxemburgo (donde se trabaja en paralelo en Amazon como *Business Analyst Intern*), consiguiendo ampliar las capacidades de la herramienta e reintegrando ambas funcionalidades – la parte antigua de visualización con la nueva de analítica – para dar con un producto concluido para ser embarcado sobre una página web en línea y ser utilizado en abierto por equipos internos del cliente. En lo restante de este documento, se trata de explicar claramente la estructura, lógica interna y estética de la aplicación, así como la justificación técnica y de negocio de las decisiones que fueron tomadas a lo largo del trabajo. Para facilitar la ilustración del desarrollo de la herramienta, se muestra en el ANEXO I. Diagrama de Gantt la planificación que se persiguió según las semanas que se invirtieron para cada apartado del proyecto.

1.3 MOTIVACIÓN

El Análisis Avanzado de Datos y la Inteligencia Artificial están evolucionando a una velocidad vertiginosa y, sin embargo, aún queda mucho camino por recorrer. En el estado del arte se vio cómo muchos métodos afloran de las aportaciones que llegan desde la Estadística Aplicada, pero muchas veces es la parte de la implementación donde las empresas necesitan de un esfuerzo mayor. Con este proyecto, se pretende responder a las **necesidades de SmartCity Málaga** (de aquí en adelante, también denominado *SCM*) para desarrollar un **modelo web multiplataforma en tiempo real** que aporte **informaciones de alto valor añadido** que hagan uso de estas **técnicas inteligentes**. Se pretende así crear una base para futuros modelos destinados a la resolución de este tipo de problemas u otros parecidos, desde la asistencia a personal destinada al lugar físico, a la ejecución de gestores de alarmas o detección de anomalías.

En definitiva, son muchos los caminos que quedan por explorar en el ambiente de actual transformación, a la vez que son necesarios esfuerzos de modernización en las instalaciones existentes. En este trabajo, se trata de dar solución a un caso de negocio que Endesa, empresa patrono de la Cátedra de Industria Conectada de ICAI, hace llegar a la Cátedra a través de Málaga Smart-City para agregar algunas de estas técnicas en sus mecanismos de control de activos e instalaciones.

1.4 OBJETIVOS

El propósito de este proyecto es proveer conocimiento útil apoyándonos en los datos pasados para generar nuevo valor en los datos que estén por venir del futuro de cara al funcionamiento interno y operaciones de Smart City Málaga y de Endesa en última instancia, así como para que CIC disponga de un prototipo base a partir del cual generar nuevos proyectos de interés creciente para empresas industriales y tecnológicas con alto volumen de datos con potencial de impacto.

En términos de objetivos, han sido acordados y concretados en los listados a continuación:

- Conseguir una latencia e interacción propia de un servicio web (menor de 1 segundo).
- Permitir la generación automática de informes de la salud y congestión de la red.
- Crear un modelo lo más robusto posible y agnóstico al **dato y plataforma**: evitar la pérdida de calidad frente a variaciones implícitas o explicitas de los datos de entrada, y que el programa no dependa del dispositivo, sistema operativo o versión del visor.
- Apalancar el análisis y conclusiones del estudio sobre herramientas inteligentes de reducción de dimensiones (error menor del 15% sobre el espacio reconstruido).

Es importante mencionar, no obstante, que debido a la limitación temporal que caracteriza el alcance de este proyecto – teniendo lugar en menos de un año académico completo –, evaluar la consecución de estos objetivos no serán siempre empíricamente imposibles y será realmente medible únicamente llegué el momento de ser implementado en tiempo real dentro del sistema imperante a día de hoy dentro de la compañía (al cual por razones obvias no se tiene acceso durante esta fase de prototipado). Sin embargo, con el propósito de valorar el impacto esperado, el proyecto contempla a lo largo de su extensión las estimaciones de mejora emitida sobre cada una de estas métricas.

En el ANEXO II. Reflexión sobre los o.d.s, se detalla una breve reflexión alrededor del interés que puede suscitar una herramienta como esta en la persecución de los Objetivos de Desarrollo Sostenible de la ONU, así como casos de uso que ilustren el planteamiento.

1.5 METODOLOGÍA

El desarrollo de este proyecto se divide en cuatro partes:

- Análisis Exploratorio de los Datos, Preprocesamiento e Importación.
- Generación de figuras y modelos inteligentes.
- Creación del modelo inicial e Interfaz de Usuario.
- Optimización, Integración y Entrega Final.

La primera tarea del proyecto es la de reconocer la **estructura**, **sintaxis** y **semántica** de la base de datos. El propósito es doble: por un lado, se desea realizar una **tarea de preprocesamiento que aborde el contenido**, analice la calidad del dato y reconozca relaciones preliminares entre ellas; y, por otro lado, **abordar la forma** de las variables de partida: procesamiento sobre celas vacías, *outliers* justificados (ejemplo: error en el registro, valores negativos en campos categóricos, …) y no justificados (de interés para este proyecto), normalización, … La base de datos queda así **balanceada** y **significativa**, evitando sesgos sistemáticos indeseados y generalizando la solución.

Tras haber preparado la base de datos, se procede a generar los **primeros objetos de interés**. Entre ellos, son muchas las posibilidades: diagramas de dispersión, histogramas o diagramas comparativos son algunas de las opciones para poder desplegar todo el potencial del dato y así transformarlo en información útil para el usuario.

El proceso de crear un modelo funcional inicial se basará en montar la programación de toda la estructura anterior en un modelo de **desarrollo web**, que permita ser compartido, alojado y visionado en distintas plataformas con acceso a Internet. Esto requerirá el uso de determinadas librerías disponibles en el lenguaje de programación de partida que permitan correctamente la carga de datos, inicialización del modelo, creación del **modelo cliente – servidor**, interacción dinámica con el usuario y funciones adicionales que se presuponen de tal servicio: estabilidad, manejabilidad, seguridad, baja latencia...

Por último, se debe optimizar el programa, integrar sus componentes en un único *dashboard* de sencilla lectura y asegurar el acceso. Se realizará una comparativa entre técnicas de clusterización para según la tipología de los puntos de partida y una demostración de la utilidad del PCA vs los *autoencoders*. Por último, se deberá llevar a cabo sesiones de prueba y demostración junto al cliente, así como el desarrollo de un modelo de entrega que sea sencillo y práctico para el despliegue del programa (denominado como Fase de Entrega).

1.6 RECURSOS EMPLEADOS

Para este proyecto no se necesitaron de recursos de hardware ni de inversiones materiales destacables. Por citar brevemente los primordiales, fue suficiente con el ordenador portátil de trabajo y, por temas de computación, distribución y mejora del producto, se acudió al uso de herramientas en la nube que facilitan el trabajo. En particular, cabe mencionar el uso de Google Colab (servicio ***freemium*** propio de la compañía estadounidense donde es posible ejecutar código alojándolo en clusters provistos por ellos), el de librerías *Open-Source* de largo alcance (como son típicamente *numpy* o *pandas*, pero también más particulares como *Dash* para el desarrollo del *front-end* o *scikit-learn* para el entrenamiento de modelos) y por último, los datos de variables de centros de transformación provistos por SmartCity Málaga para diseñar el producto.

Sección 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Esta sección será dedicada para explicar la realidad de las soluciones tecnológicas que se utilizan en el medio industrial para responder a las problemáticas evocadas durante la INTRODUCCIÓN, utilizando un enfoque recopilatorio en cada uno de sus vértices. En un segundo plano, se continuará explicando las funcionalidades que posibilitan la realización de este proyecto en su sentido más general, y por último explicando las bondades que aportan estas últimas, incluyendo tanto los estándares de desarrollo web utilizados como las técnicas inteligentes de *Machine Learning* que se abordan en el contenido del documento.

2.1 ESTADO DE LA CUESTIÓN: CONTEXTO ELÉCTRICO

La sensorización, automatización e implementación de redes inteligentes dentro del control de la red están a la orden del día dentro de la industria eléctrica de este país y resto del mundo. Siendo este campo de actualidad, cabe destacar la gran popularidad que han adquirido la aplicación de técnicas de aprendizaje automático para la gestión y mantenimiento de activos en los últimos años. Este tipo de algoritmos descriptivos inteligentes, de gran utilidad para el análisis de activos en otras metodologías como por ejemplo por pruebas de vibración, inspección no invasiva o de la temperatura infrarroja [8], están hoy siendo utilizados para la detección de defectos en los centros de transformación con grandes resultados y es muy predominante su uso en el **mantenimiento de activos y prevención de desgaste, y en última instancia fallo**.

Los **transformadores de potencia** son el núcleo de los sistemas eléctricos de distribución y transmisión. Sin embargo, estos quedan constantemente a merced de **esfuerzos eléctricos, mecánicos, térmicos y ambientales**. Por otro lado, los transformadores de potencia son hoy en día uno de los componentes más **críticos** y **costosos** del sistema eléctrico, ocupando en ciertos casos el 60% de la inversión total. Debido a la inversión por adelantado que suponen

antes siquiera de iniciar la actividad comercial, la monitorización y mantenimiento de la condición de estos son tareas fundamentales sobre el terreno. Actualmente, existen varios métodos de diagnóstico para monitorear el estado de salud de los transformadores que vienen desarrollados en extensa literatura. Sin embargo, dichos métodos muchas veces no son capaces de lograr evaluar la condición cuando ocurren múltiples fallas y la aparición de la inteligencia artificial permite combinar atributos rastreados históricamente para lidiar con dicho problema. [9]

Entre las fallas más comunes, se hallan desde deformaciones en el bobinado debido a sobrecargas o desgaste, el deterioro del aislamiento e incluso desperfectos materiales debido a infracciones humanas. Estas van siendo resueltas cada vez más con métodos inteligentes sobre datos extraídos telemáticamente sin necesidad de intervención humana periódica preventiva, muy ineficaz. Gracias a la democratización de la inteligencia artificial, las herramientas de visualización de datos y la proliferación de instrumentos de medida inalámbricos es hoy posible acotarse a variables comunes como la corriente y la intensidad para después aplicar desde un análisis gráfico tipo *Lissajous* para estimar imperfecciones mínimas dentro del bobinado [10], la aplicación de ecuaciones matemáticas y de criterio experto para calcular y explotar el THI (*Transformer's Health Index*) [11] o la gestión por parte de empresas de servicios públicos de transformadores de tipo residencial en medio urbano [12].

Otra de las grandes aplicaciones son los enfoques numéricos utilizados para la **estimación de consumo** basándose en algoritmos profundos. El desarrollo de las *Smart Grids* o del propio mercado eléctrico propicia el incentivar técnicas más potentes para la estimación del consumo, ya sea industrial, urbano o doméstico [13]. Este consumo es cada vez menos normativo entre distintos hogares de incluso mismas geografías y debido a la naturaleza no almacenable de la energía eléctrica, esta tarea de dimensionamiento de la carga en tiempo real es fundamental para la asignación de recursos en el lado generador.

En tercer lugar, otro ámbito donde estas lógicas cobran sentido para mejorar la operación de la red es la **predicción de flexibilidad del sistema**. La definición de flexibilidad varía con

la literatura y depende fuertemente del objeto de estudio. Por ejemplo, flexibilidad de la unidad de generación se define por sus capacidades de potencia, sus límites de variación de flujo energético (*ramping limits*) o tiempos de respuesta. A estos además se podrán añadir otros nuevos dependiendo de si se habla de generadores eólicos o de panales solares instalados en edificios inteligentes. Sin embargo, de manera holística se puede definir esta como la **capacidad del sistema eléctrico de potencia para mantener un suministro continuo** y regulado **frente a variaciones** en la generación o consumo, independientemente de su origen. [14] Este concepto es hoy criterio fundamental en el diseño de redes.

2.2 CIENCIAS IMPULSORAS FRENTE AL DESAFÍO TÉCNICO

Aunque bastante distintas en objetivo, las aplicaciones anteriormente explicadas guardan puntos en común en términos de medios utilizados para aprender información de la multitud de datos y en particular, destacan dos: las **herramientas de visualización de datos** y el **reconocimiento de patrones para la estimación de comportamientos futuros**.

Por un lado, la **ciencia de la visualización** ha evolucionado enormemente en las últimas décadas. La aparición de nuevos dispositivos, el desarrollo *app* y *web* y la creciente complejidad de la toma de decisiones ha acelerado enormemente las técnicas de representación de información. Sin ellas, los datos son incomprensibles para las personas y sistemas, pero acompañados de una narrativa, centrándose en una dimensión o facilitando su lectura, breves figuras permiten entender problemas complejos en cuestión de segundos. Utilizando elementos como **cuadros, gráficas o mapas**, se provee así de pasos de acceso a la comprensión de tendencias, anomalías o patrones entre el desorden. Autores contemporáneos reputados como Edward Tufte o John Tukey han marcado la tónica general hacia un paradigma más minimalista y utilitario, allanando el camino hacia técnicas más potentes e interactivas potenciadas por los desarrollos de *software*, que estén al servicio del usuario en lugar de pivotar alrededor de la complejidad de partida.

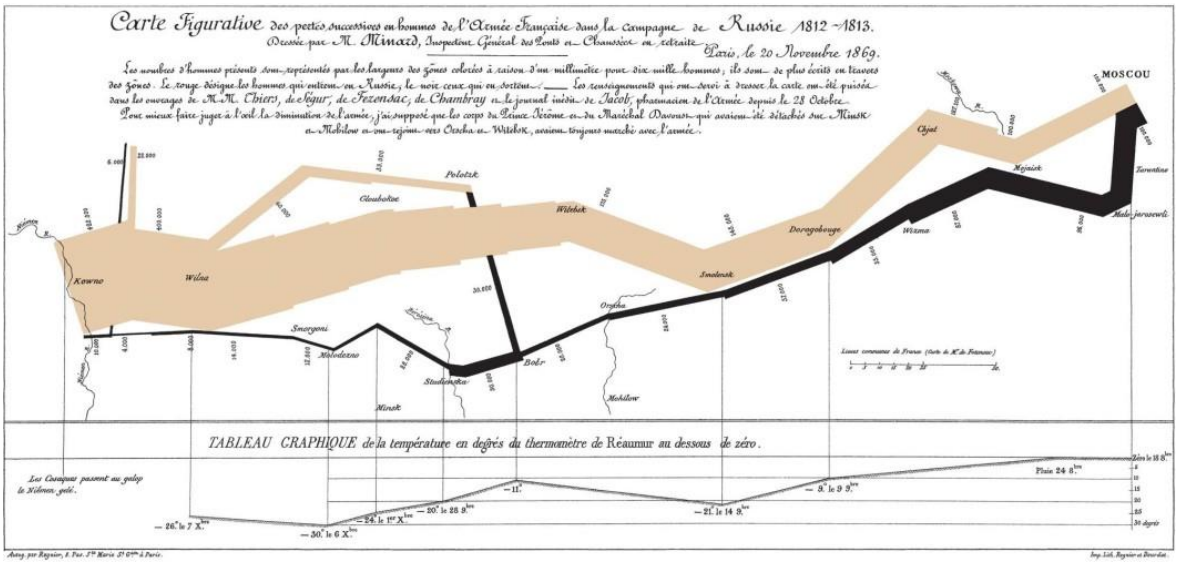


Figura 9: Avance de tropas napoleónicas hacia Moscú, de Charles Joseph Minard [15]

Por otro lado, entre las técnicas más conocidas dentro de las familias de algoritmos de clasificación o predicción, se encuentra el **uso de modelos estadísticos** como los modelos basados en cadenas de Markov, aprendizaje no supervisado y en particular, la **clusterización divisiva de elementos** (algoritmo de *k-means*, OPTICS, DBSCAN, …) o **métodos aglomerativos** (*clustering* jerárquico), pasando por **modelos de aprendizaje profundo** nacidos de la explosión del *Deep Learning*, como los modelos **GRU** (*Gated Recurrent Units*) o los **LSTM** (*Long-Short Term Memory*), donde se hace uso de datos secuenciales (por ejemplo, series temporales o Lenguaje Natural).

Debido a las limitaciones del proyecto, no es posible probar la profundidad de técnicas disponibles, pero sí es de especial interés el utilizarlos para **representar de manera comprensible** la información útil para el usuario y conocer cuáles son las **variables más explicativas** para a partir de un subconjunto, ser capaces de reconocer comportamientos destacables que se puedan agrupar. Formalmente, estos problemas se conocen como la **selección de atributos** (o *feature selection*), **análisis de grupos** (o *cluster analysis*) y **reconstrucción** del espacio de origen en menor dimensionalidad (o ***dimensionality reduction***). El objetivo es desarrollar un programa que proporcione figuras descriptivas del problema de manera fehaciente, demuestre el poder comparativo entre centros de

transformación y, de manera metódica y eficaz, reconocer cuales son los parámetros que mejor describen nuestra población. A continuación, se comentan las técnicas de reducción de dimensionalidad de dos más utilizadas: ***Principal Component Analysis*** (PCA) y ***Auto-encoders*** (AE).

- El PCA selecciona un nuevo **subconjunto de variables no correlacionadas** (o "componentes") ortogonales que son ordenados en función de la **varianza del conjunto de partida que explica**. Para ello, el algoritmo calcula la **matriz de covarianza** de la matriz de partida *X*, computa sus vectores propios y valores propios asociados y en función de estos últimos, se filtran los *k* primeros. Matemáticamente, esta técnica equivale a buscar aquella **proyección óptima de los datos sobre una nueva base** tal que el error por **mínimos cuadrados es mínimo**.

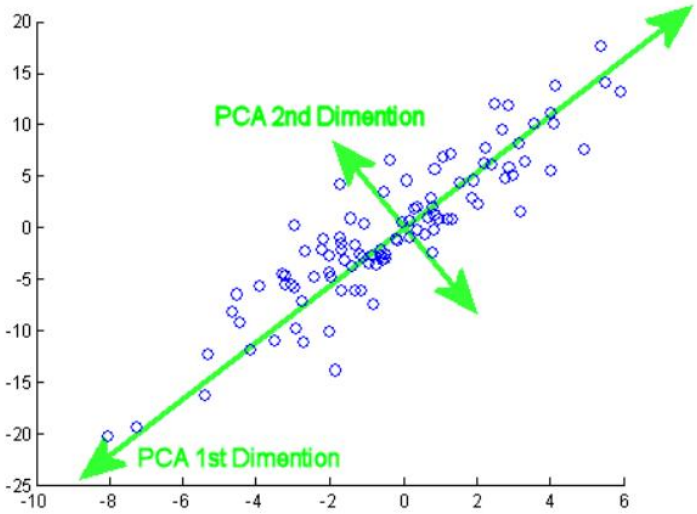


Figura 10: Ejemplo de PCA en 2D [16]

- Los *auto-encoders* sin embargo son **redes neuronales artificiales** utilizados para aprender **codificaciones eficaces** de datos de manera **no supervisada**. El objetivo es aprender una **representación** del conjunto de partida entrenando de manera a ignorar el "ruido" de la señal. La estructura del AE es simétrica, y cuenta con un lado de "reducción de la redundancia" complementado de otro encargado de la

"reconstrucción" del conjunto de partida. Existen muchas variantes, desde ejemplos regularizados de modelo para evitar sobreentrenamiento (o o*verfitting*) a los AE variacionales (utilizados para modelos generativos de datos) y sus aplicaciones cubren desde el reconocimiento facial, la extracción de atributos o el aprendizaje semántico del lenguaje natural.

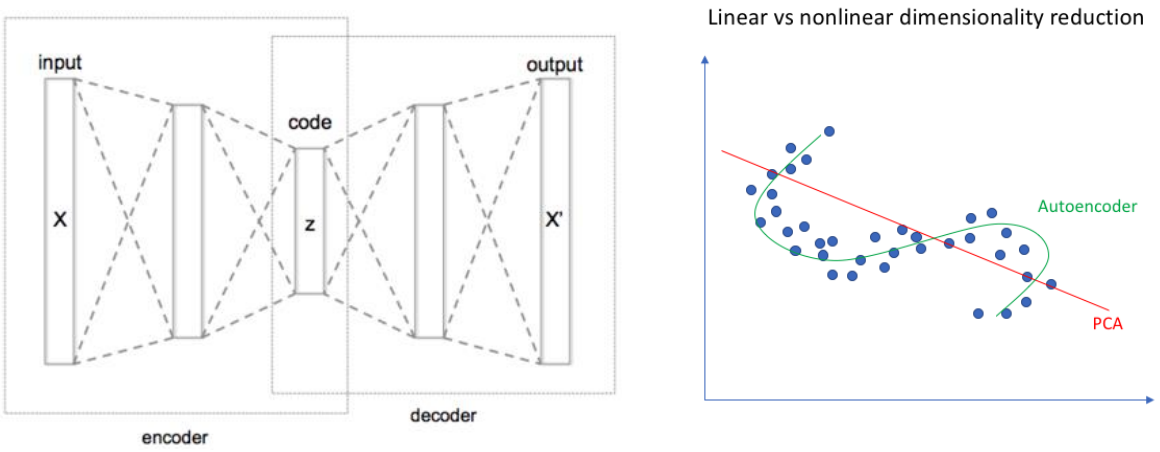


Figura 11: Representación esquemática de AE, y diferencia linealidad frente al PCA [17]

El desarrollo de estas técnicas será desarrollado más en profundidad en este trabajo y serán aplicadas a nuestro conjunto de datos de partida en distintas versiones (capas intermedias, funciones de activación no lineales, …).

2.3 ESTÁNDARES PRÁCTICOS: MARCOS DE TRABAJO

Esta sección es dedicada a la introducción dentro de las herramientas referencia utilizadas en las dos grandes mitades de este proyecto: la visualización de la página web, y el análisis por inteligencia artificial. Cada una de ellas se apoya sobre la otra y hace uso de las bondades de la opuesta para potenciar el propósito de cada una, siendo complementarias. De lo más general a lo más particular en este proyecto, se pretenden dar unas breves pinceladas antes de entrar de lleno más tarde durante la Implementación de las soluciones.

2.3.1 DESARROLLO WEB: PLOTLY FRENTE AL PARADIGMA TRADICIONAL

Se habla de desarrollo web para referirse a todo trabajo o esfuerzo involucrado en el desarrollo de un sitio web, ya sea de dominio público (Internet) o para el seno de una red privada (Intranet). El desarrollo web es muy extenso y abarca desde el desarrollo más simple de páginas estáticas con texto plano a complejas aplicaciones, herramientas electrónicas de negocio o los servicios de Redes Sociales. Dentro de cada organización, la escala del trabajo puede variar desde una persona especializada o un servicio externalizado a suponer el núcleo de la estructura empresarial (como es el ejemplo de Instagram o incluso Amazon Retail). Sin embargo, cabe destacar que todas ellas guardan similitudes, como son las metodologías *Agile* para referirse a las metodologías de trabajo incrementales propias de estos equipos, o incluso las grandes ramas de especialización como son el *Front-End* y el *Back-End*, ocupándose respectivamente del comportamiento y visuales, o del alojamiento de recursos en servidores para nutrir la página.

En tanto que industria, el desarrollo web no ha dejado de crecer desde la democratización de Internet. Guiado por la aparición de más y más modelos de negocio digitales para la publicidad, venta (*e-commerce*) o personalización del producto, una serie de herramientas Open Source aparecen para estandarizar los conocimientos a la par que estabiliza la comprensibilidad de páginas en un mundo virtual donde las fronteras como se conocían pasan a un segundo plano.

Dentro de este ecosistema, la variedad de productos se multiplica para a día de hoy, paradójicamente, converger en una serie de lenguajes de programación para estructurar las páginas web (o crear paquetes de gran popularidad abstrayéndose de lenguajes particulares, como es el caso reciente de React). En particular, este apartado profundiza en el famoso y fundacional paradigma del **HTML – CSS – JavaScript.**

HTML, CSS y Javascript son los tres pilares complementarios que constituyen la gran mayoría de páginas web en la actualidad. Hoy en día, cuando un desarrollador web persigue dar solución a un objetivo final, el primero de las tareas consiste en subdividir todas aquellas funcionalidades de esa gran idea, traducirla en instrucciones accionables y saber asociarla a alguno de los tres lenguajes anteriores para así construir un código coherente para los buscadores web como Chrome o Safari y puedan así mostrar un contenido a los usuarios interesados, siguiendo una lógica implícita dentro del programa.

Para explicar la utilidad de esta configuración, es necesario remontarse un tiempo atrás ya que, siendo rigurosos, HTML y CSS no son técnicamente lenguajes de programación sino **estructurador de página** y **encapsulador de estilo** respectivamente y estos se encuentran en la parte frontal de la aplicación.

A inicios de 1990, HTML era el único lenguaje disponible en la web. Entonces los desarrolladores se veían abocados a desarrollar código estático poco robusto, de bajo nivel y página a página. Sin embargo, hoy cada uno de ellos tiene su lugar y función propia:

- HTML provee la estructura básica de la página web, el esqueleto que marca los elementos visibles y tangibles de la aplicación.
- CSS se utiliza para potenciar la visibilidad de los elementos anteriores y se encarga de controlar la presentación, formateado y despliegue de los bloques, todo lo referido a la estética.
- JavaScript se utiliza para describir la lógica y comportamiento de los distintos elementos, reaccionando a eventualidades fruto de la interacción con el usuario, datos externos, otras páginas, …

2.3.2 HTML

HTML está en el núcleo de cada página web, sin importar la complejidad o el número de tecnologías montadas sobre el esqueleto base. Es el primer paso del desarrollo pues pone sobre la mesa los elementos que entrarán en juego dentro de la aplicación, dando forma al contenido de la página (a través de párrafos de texto, imágenes, hipervínculos, videos, …). HTML viene de la expresión *HyperText Markup Language* y las últimas dos siglas significan que en lugar de utilizar un estilo de programación tradicional para ejecutar funciones, HTML utiliza **etiquetas** de diferente tipo para determinar el tipo de contenido que sirve a la página. Los lenguajes *Markup*, en efecto, se encargan de clasificar los tipos de contenido y por lo general siguen distinciones de inglés plano – como *header* o *paragraph*.

A su vez, dentro de un código HTML web se pueden encontrar muchas etiquetas que se siguen una a la otra o incluso se anidan, de la misma manera que un dominio puede estar compuesto de varias páginas HTML que se conectan por medio de links. Las etiquetas por lo general envuelven el contenido al cual se refieren para encapsular así el tipo de contenido al cual se asocia, como ocurre en el ejemplo a continuación para determinar un párrafo simple:

```
<p>Esto es un párrafo.</p>
```

Utilizando HTML, es posible crear con total libertad, formatear párrafos, controlar la sangría, crear listas, subrayar texto o insertar imágenes. Sin embargo, aún no se puede controlar la **manera o formato** con la cual estos elementos se presentan al usuario.

2.3.3 CSS

CSS se define a partir de la expresión *Cascading Style Sheets*. Este lenguaje dicta cómo los elementos previamente definidos en HTML deben aparecer en el *Front-End* de la página. CSS aporta complejidad a los elementos brutos durante la estructura del contenido para, de manera eficaz, ofrecer un estilo particular al contenido para presentarlo de la manera deseada. La utilidad de separar ambos lenguajes es para marcar una división clara entre contenido y estética.

Los parámetros disponibles incluyen la fuente, el color de fondo, justificación, colocación de bloques… CSS aporta estilo a la página y definitivamente la vuelve más amigable. Este lenguaje afecta a la harmonía global y es poderosa sobre todo cuando se engloban estilos para grupos de elementos, que a su vez pueden ser inteligentemente sobrescritos para el caso particular.

Figura 12: Diferencia entre HTML plano y HTML+CSS

En definitiva, CSS se encarga de establecer una lista de reglas asociadas a atributos de las etiquetas HTML. En un inicio este lenguaje nació de la dificultad existente en la primera época para sostener todas las funciones dentro de HTML, depurándolo y marcando claramente una diferencia clara entre ambos propósitos.

Cada buscador web tiene una "hoja de estilos" por defecto, por lo que cada página web del mundo es por lo menos afectada por una de ellas, independientemente de si el creador genera una customizada para su caso o no. Este es sobrescrito por desarrolladores para sacarle ventaja al formato para presentar la información de manera más legible, y así cada elemento sigue un recorrido en "cascada" hasta llegar al estilo más prioritario, que es el que finalmente acaba determinando la presentación del objeto en la web.

Sin embargo, hasta ahora nuestra página sigue siendo estática y el usuario no puede aún hacer más que ingresar en ella y leer de su contenido.

2.3.4 JAVASCRIPT

Es el más reciente y complejo de los tres lenguajes, pero también el más potente y desarrollado de todos ellos. Lanzado en su primera versión durante 1995, hoy JavaScript

(JS) es soportado por todos los buscadores modernos y es utilizado por la mayoría de ellos para conseguir las funcionalidades más singulares y complejas.

JS es un lenguaje de programación basada en lógica computacional para conseguir modificar el contenido del sitio web en respuesta a los estímulos del usuario. Entre los cases de uso más conocidos, se encuentran las cajas de chequeo, las llamadas de acción o añadir nuevas entidades a la plantilla de partida. En definitiva, JS trae la interactividad a la página, desde tareas de comunicación, seguridad o servicios avanzados (como *pop-ups* o *callouts*).

2.3.5 DASH

Concluye este capítulo con la introducción a Dash pues este es la plataforma base sobre la cual la solución es desarrollada, debido a su facilidad de uso, integración y portabilidad.

Dash es un **entorno de producción** del lenguaje de programación **Python** creado para desarrollar **aplicaciones web de analítica**. Escrito sobre los *frameworks* de Flask, Plotly.js y React.js, Dash es la opción idónea para construir aplicaciones orientadas a la visualización de datos con una interfaz personalizable para el usuario, y a bordo de la programación Python (acompañada de todos sus puntos positivos, en particular de Ciencia de Datos). Como enuncian en el github oficial, el código de las apps Dash es declarativo y reactivo, lo que hace fácil construir complejas apps interactivas. El desarrollador es así capaz de declarar las propiedades de un resultado sin tener porqué explicitar las secuencias de pasos a seguir, el "cómo" siendo dejado de la mano del interpretador del programa

Dash es especialmente interesante pues consigue abstraer las tecnologías y protocolos utilizados tradicionalmente en el mundo web (en particular, los de HTML, CSS y JS) y supone una diferencia revolucionaria pues a partir del conocimiento en Python, el desarrollador es capaz de desarrollar aplicación interactivas basadas en web alrededor del proyecto de ciencia de datos ya alojado en este lenguaje, sin tener que remontar la curva de aprendizaje de tres idiomas distintos y consiguiendo una fusión indistinguible dentro del mismo programa. Las aplicaciones Dash son renderizadas directamente en los buscadores web de preferencia. Es posible desplegar las aplicaciones a servidor y compartirlos a través

de URL, lo que hace de esta herramienta un método multiplataforma y listo para mostrar incluso en móvil.

Dash es una librería Open Source, con una página extensiva e increíblemente bien documentada y una comunidad muy activa. La herramienta se distribuye bajo la permisiva licencia MIT y Plotly es la empresa desarrolladora de Dash, ofreciendo también servicios de consultoría para el despliegue de aplicaciones Dash en entorno de producción. [18]

En términos de solucione similares, destacan las alternativas como Power BI o Qlik Sense. Frente al primero, aunque la diversidad de integración de datos es vasta para PowerBI, Dash cuenta con los módulos disponibles para cualquier archivo Python (integración de APIs de data estructurada o no estructurada, e incluso de herramientas de Big Data como Spark). Power BI soporta una gama reducida de generación de informes que muchas veces imposibilita su compatibilidad con aplicaciones de terceros. Las agregaciones de datos son en ocasiones simplificaciones con filtros cruzados y, aunque la manejabilidad se ve mejorada por su sencillez de uso, Dash cuenta con la eficacia de la programación de Python junto con los módulos de Ciencia de Datos propios de librería programadas nativamente en C. Frente a Qlik Sense, los usuarios reconocen que este último es más sencillo de utilizar. Sin embargo, Dash es la herramienta multiusos que satisface necesidades de empresas de todo tipo. Dash es más manejable para su gestión, personalización y administración. Ambos proveen un nivel de asistencia similar pero en términos de actualizaciones y desarrolladores futuros, los usuarios siguieron prefiriendo Dash sobre el primero. [19]

Es por esta serie de razones y muchas más que se irán presentando por lo cual Dash era la opción a elegir sin lugar a dudas, ofreciendo grandes resultados con un relativo esfuerzo moderado respecto al resto de alternativas.

2.3.6 INTELIGENCIA ARTIFICIAL: SCIKIT-LEARN

De la predicción a la clasificación, pasando por el reconocimiento de patrones característicos o la generación de ejemplos ficticios a partir de un conjunto de entrenamiento, muchas son las misiones que han sido emprendidas en el pasado con la ayuda de la inteligencia artificial,

mejorando en cuestión de años el nivel del estado del arte en múltiples ocasiones, incluyendo el nivel del rendimiento humano. Sin embargo, pocas veces se aprecia la importancia que cobra en esta inundación de la inteligencia artificial la creación de librerías de programación de código abierto que permiten a usuarios y aprendices una sencilla creación de los bloques fundamentales del *Machine Learning*. En efecto, a la vez que la carrera académica anunciaba nuevos hallazgos teóricos a través de la investigación fundamentalmente matemática, compañías por todo el mundo consiguen traer la democratización de la Inteligencia Artificial, facilitando a interesados el poder aprender de manera autónoma y probar el nuevo instrumento de moda desde un ordenador personal.

Figura 13: Frameworks de IA desarrollados por Google (TF), Facebook (Pytorch) o ONEIROS (Keras)

Entre estos famoso módulos, aparece públicamente en 2010 la librería de *Machine Learning* llamada **Scikit-learn** (también conocida como sklearn, o SL). SL es una **librería software gratuita** de Aprendizaje Automático para Python que integra modelos de algoritmos para la clasificación, regresión y agrupamiento. Su última versión data de Abril de 2021 y, hasta la fecha, el módulo permite incorporar con apenas algunas líneas una gran variedad de modelos simples (*Support Vector Machine, k-means, Random Forests, …*). El proyecto de scikit-learn nació como un proyecto de verano en el corazón de Google de la mano del ingeniero David Cournapeau, aunque el código original ha sido revisado en sus lanzamientos posteriores a la inicial. A día de hoy, SL es uno de las librerías más populares en el mundo, de fácil comprensión y usabilidad.

SL tiene como ambición la de resolver con ayuda de su caja de herramientas los problemas de tipo aprendizaje. Como aparece indicado en su página web, “un problema de aprendizaje considera un conjunto de muestra de datos para, a continuación, intentar predecir las

propiedades de otros datos desconocidos" [20]. Este conjunto de datos pueden venir provistos de uno o más atributos y engloban problemas que pueden separarse en la siguiente estructura:

- Aprendizaje supervisado: aparte de ser caracterizados por los valores que toman en cada atributo, en este caso los datos también son acompañados de atributos que se desea predecir en observaciones no conocidas. Se dice que los datos han sido **recategorizados** por una variable o serie de variables adicional. Estas etiquetas componen las salidas a predecir a través del entrenamiento de modelos y según su naturaleza, se distinguen dos familias.
    - Clasificación: Las muestras pertenecen a una o más clases y se pretende predecir la clase de aquellos datos nunca observados antes atendiendo al valor de sus atributos de partida. A modo de ejemplo, se puede visualizar dicho problema al imaginar un modelo que registre la presencia de semáforos en la imagen de la cámara frontal de un automóvil (siendo la variable en salida entonces binaria) o la presencia de un tipo particular de vehículo (donde entonces habría una clase para camiones y otra para motocicletas). En definitiva, la variable en salida del modelo es del tipo **discreto**.
    - Regresión: En esta ocasión, la salida del algoritmo deseada es **continua**. Dados los atributos de entrada, se desea conocer la magnitud de una variable que puede esta vez tomar toda una infinidad de valores posibles. Ejemplo de ello sería un algoritmo estimador del precio de venta de un edificio dada la localización geográfica y calidad residencial de la zona del inmueble.

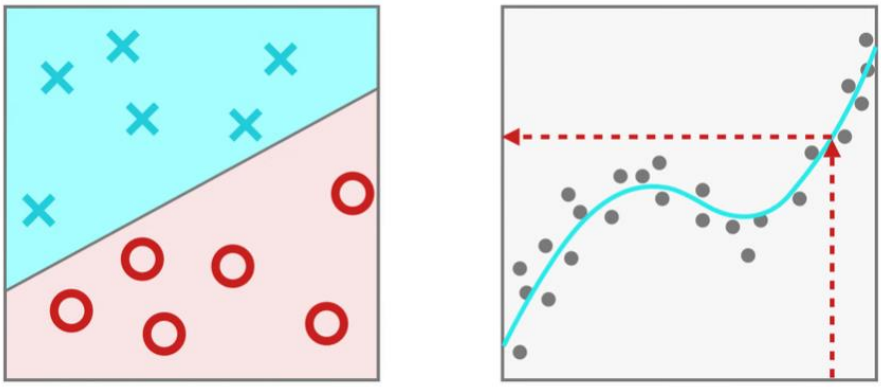


Figura 14: Ejemplo de Clasificación y Regresión

- Aprendizaje No Supervisado: La diferencia fundamental de esta familia es la ausencia de variable objetivo a predecir. Por lo contrario, en este problema se pretenden inferir características comunes a través del descubrimiento de patrones similares en grupos de datos o transformar el espacio de partida para dar con un resultado transformativo de manera óptima. Entre los casos más conocidos, se encuentran los problemas de agrupamiento (donde la intención es la de formar corpúsculos entre puntos que estén lo suficientemente cercanos entre ellos y lejanos de otros), de estimación de densidad (aproximación probabilística a la caracterización de la familia de datos) o la proyección de datos para posibilitar su visualización (en dos o tres dimensiones).

En este proyecto, las llamadas a la librería de Scikit-learn intervienen en dos partes del programa para tratar dos problemas de origen distinto:

- En un primer lugar, se hace uso de modelos de reducción dimensional para la reducción de parámetros de los atributos de entrada. En el núcleo del trabajo, se ha realizado una comparativa de resultados entre dichos modelos para reconocer cuál de ellos mantiene la distribución interna de la manera más eficaz y eficiente, consiguiendo así los resultados más convenientes con la menor deformación y menor tiempo posible.

- En un segundo lugar, otro conjunto de modelos es utilizado para realizar una detección de valores anómalos registrado en cada centro de transformación de la ciudad de Málaga, dados los valores registrados de entrada. Un segundo trabajo de análisis es realizado para alcanzar los mejores resultados medibles, respetando la latencia de la aplicación Dash y arrojando información sobre la operación de un centro de transformación dentro de un rango de tiempo dado.

Tanto los dos análisis como la representación y resultado del candidato escogido son comentados más en profundidad a continuación en la Sección 4. Implementación de las soluciones, donde se profundiza tanto en la técnica como en la lógica de la implementación de ambas soluciones y, en definitiva, del proyecto en general.

Sección 3. DEFINICIÓN DEL PROBLEMA

Una vez se ha abordado el contexto, se debe delimitar el alcance del trabajo, conocer las capacidades a nuestra disposición y conocer cuál es el estado del proyecto antes de entrar de lleno (pues en este caso sí se contaba con un trabajo previo de análisis de los datos). Como en cualquier proyecto de envergadura, esta fase fue particularmente crucial pues suponía la familiarización con los medios técnicos y resultados esperados al finalizar, conocer el estado actual y trazar un plan intermedio con pasos incrementales que concluyesen en una herramienta de utilidad práctica y funcional.

3.1 DESCRIPCIÓN DEL CASO DE ESTUDIO Y REQUERIMIENTOS

Fueron durante los primeros meses del proyecto cuando las sucesivas reuniones con el representante por la parte de Smart City, socio del proyecto, tuvieron lugar. Aunque la definición de las guías argumentales fueran aún muy maleables, se tenía claro el objetivo global: hacer uso de las variables de medición físicas existentes en los centros de transformación de Endesa en la ciudad malagueña para analizar descriptivamente los datos arrojados en el pasado en un primer momento, para eventualmente proponer una herramienta predictiva e incluso prescriptiva al alcance del equipo interno, tanto de ingenieros de la sala de control a operarios de mantenimiento de campo.

En aquel momento dado, existían otros proyectos en paralelo concertados con otros equipos de investigación para resolver distintos casos de estudio. Entre los más sonantes, destacan trabajos analíticos sobre los datos de clientes y de sus consumos en contadores inteligentes para la detección de posibles fraudes, integración de soluciones IoT o la clusterización de grupos de consumición. Sin embargo, la misión que se propuso era el desarrollar una **herramienta de visualización interactiva** sobre variables de la red eléctrica.

En relación con sus proyectos sucesores, la meta de la empresa era el desarrollar un producto digital elaborado para permitir el seguimiento y evolución temporal de la congestión eléctrica dentro de un área geográfica dada.

Como herramienta de interacción en tiempo real y con portabilidad flexible en dispositivos de distinto tipo, las condiciones de resultado exigidas eran claras y se apoyaban en tres ejes diferenciados: **facilidad de uso, interpretabilidad de los resultados y estabilidad de la integridad de la página** (tanto *front-end* como *back-end*). A continuación, se repasa qué significan los anteriores puntos en términos de exigencias objetivas y subjetivas de la solución.

En primer lugar, se habla de facilidad de uso de una aplicación cuando nos referimos al grado de **correspondencia entre intención y resultados** que el usuario medio siente al utilizar una herramienta clásica digital, es decir, lo intuitivo que sea conseguir el cometido esperado dentro de la aplicación. En el universo programador, es conocida la regla no escrita de la manejabilidad implícita: la cantidad de veces que un usuario deba dirigirse al manual de usuario suele ser inversamente proporcional con la eficacia que demuestra en satisfacer su propósito inicial. Es un fenómeno que ha sido guía narrativa de los últimos años y que ha marcado la diferencia entre los productos de la actualidad y sus análogos de hace veinte años (ya sean videojuegos, aplicaciones o teléfonos móviles). Para el desarrollo de la aplicación, se tiene en mente esta máxima no solo para mejorar la **experiencia de usuario,** sino también para amplificar el efecto del segundo punto de esta lista.

Figura 15: Estéticas de manuales de usuario entre una máquina de escribir de los 80 y un teclado Mac de

La interpretabilidad de los datos – y más en particular, de los gráficos, figuras y análisis – reagrupa toda funcionalidad o característica puesta en lugar para facilitar la lectura de información difícilmente analizable con respecto a su forma de presentación más básica. De manera general, el público reconoce los beneficios que pueden aportar sobre este punto elementos comunes como la elección de diagrama, la paleta de tonos escogida o la distribución de las viñetas dentro de la plantilla general. Sin embargo, los métodos que potencian esta característica son mucho más diversos y han sido estudiados en dominios tan distintos como son la publicidad, la política o la educación. Esta condición cobra especial interés cuanto más densa en información es el resultado y puede suponer la diferencia en la adopción de la herramienta por lo que también es una prioridad en este proyecto.

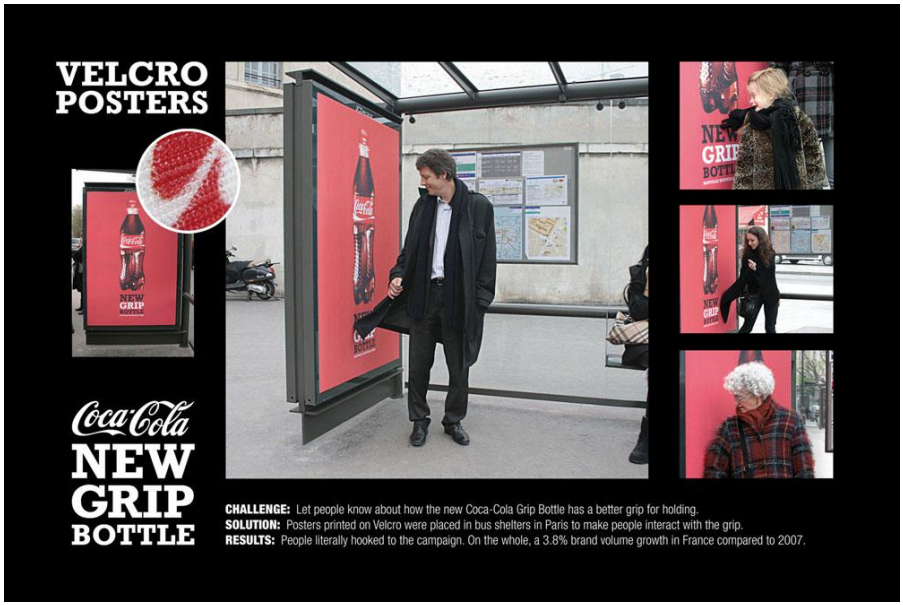


Figura 16: Nuevas formas de publicidad en el siglo XXI (caso de Coca-Cola)

Para finalizar, la estabilidad juega un papel clave en la satisfacción de usuario, y quien haya utilizado un visor de video online en una zona con poca cobertura sabrá confirmar la hipótesis. Entre otros muchos casos anecdóticos, suele ser este un reto importante al que ha día de hoy se están enfrentando empresas de entretenimiento (como es el caso de *Twitch* como plataforma de *streaming* de ocio, o de *YBVR*, la *start-up* española que tiene como

propósito la retransmisión en directo de eventos deportivos con tecnología de Realidad Virtual) y para lo cual esperan que mejoras como el *Edge computing* o el 5G no sólo consigan mejorar la experiencia sino que abran paso a modelos de negocio y herramientas imposibles hoy en día.

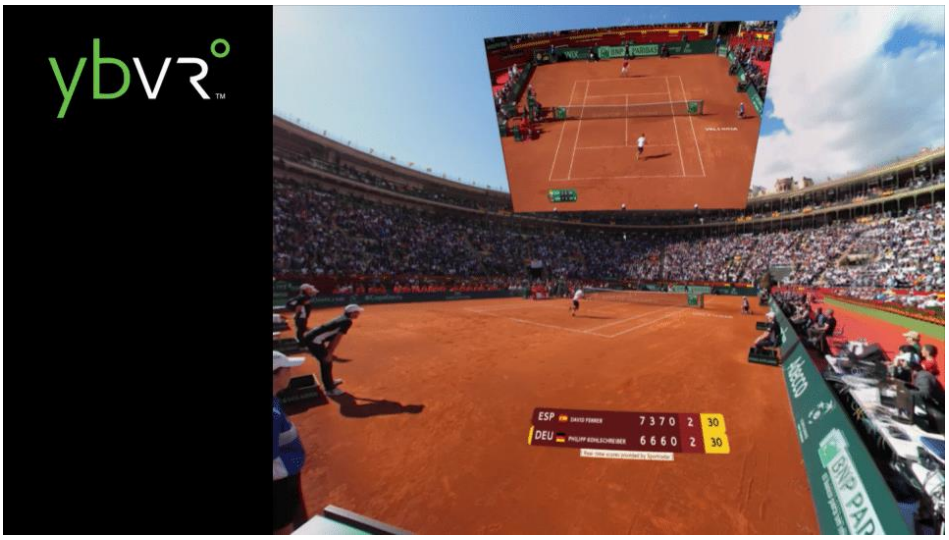


Figura 17: Ejemplo del visor retransmitiendo deporte en directo en la app de YBVR [21]

Evidentemente, a este conjunto de consignas se suman todas aquellas que son comunes a la multitud de aplicaciones presentes hoy en día en el mercado – por ejemplo, optimización de recursos, escalabilidad, compartimentalización de las funciones, seguridad, …) pero en este apartado se ha explicado la particular importancia de estos principios. La reflexión que sigue por lo tanto es saber cuáles son las consecuencias que cada una de ellas tiene sobre la interacción con el usuario, qué beneficios supone y cómo son las posibles limitaciones que anticipamos que puedan ocurrir a futuro.

3.2 BENEFICIOS Y RIESGOS

En el desarrollo web, se da frecuentemente un compromiso importante entre latencia y definición de resultados. Cuando un usuario empieza sesión en la aplicación de *Youtube* mientras ve su video favorito, ocurre que ante limitaciones de transferencia de datos, el video reconoce la limitación de capacidad del canal y decide reducir la resolución de la pantalla para evitar el momento de parada del vídeo para cargar datos y así continuar con la reproducción (en inglés, estos términos son conocidos como *resolution downgrade* y *buffering*, y han demostrado en el pasado la importancia que estos eventos tienen en la sensación del espectador) [22][22][22]. En el desarrollo de este proyecto, es preciso reconocer estos conflictos para utilizarlos como oportunidad y optimizar el empleo de recursos en lo mínimamente necesario para mostrar un resultado de calidad sin ir en detrimento de su presentación, como se verá más adelante en la Sección 4.

En otro plano del proyecto, existe el problema de la gestión de datos de entrada. Si no se presta la suficiente atención a este apartado, puede provocar un deterioro del sistema que no sabe responder al formato particular de los archivos de inicio. Tanto a nivel de formateo como de limpieza, agrupación, procesamiento, es importante dedicar acciones concretas para hacer frente a la ingesta de datos no esperados, campos vacíos o incorrectos que, si no tienen respuesta, pueden imposibilitar la ejecución de las visualizaciones o, en el peor de los casos, interpretar conclusiones sistemáticamente erróneas.

En tercer lugar, es importante dentro del programa segmentar convenientemente las secciones de código. Esta buena práctica permite delimitar explícitamente las funciones de cada apartado, evita la duplicación innecesaria de líneas presentes en distintas partes por medio del uso de funciones y delimita el acceso a determinados atributos o métodos que se quieran restringir de los elementos de nuestra aplicación (como ocurre en la programación orientada a objetos). El sistema de comunicaciones y verificaciones que se establece entre objetos permite controlar el flujo de operaciones, personalizar las acciones posibles o no sobrecargar la carga que sufre algún recurso compartido.

Sin embargo, por el lado de las limitaciones, cada uno de los tres puntos anteriores también suponen degradaciones en alguna característica del sistema. En el caso del compromiso entre prestaciones, se induce dentro de la aplicación una priorización de objetivos que quizás no se corresponde con la del usuario y de la situación donde use la aplicación, o en el segundo caso, es posible conocer el formato original y prototípico que se espera, pero nunca se podrá anticipar todas las versiones posibles del *corpus* que alimenta el programa e incluso en tal caso, se estaría yendo a la contra de la optimización de recursos. En el tercer caso, desplegar un sistema orientado a objetas obliga a programar las interacciones que se quieran admitir entre instancias de clase, pero la complejidad crece exponencialmente con la diversidad de objetos que se quiere implementar.

Estos riesgos son revisados a lo largo de la implementación de la solución y explican en cierta medida las decisiones que se han tomado, aunque ello haya podido suponer un empeoramiento de otra característica más obvia.

3.3 ALCANCE DEL TRABAJO, PRESUPUESTO Y ESTADO DE PARTIDA

Como se ha venido comentado en los párrafos previos, la etapa del proyecto era muy incipiente cuando se dio inicio a finales de 2020 y las posibilidades quedaban abiertas a discusión y modificación. Sin embargo, considerando las directrices del CIC, el enunciado de Smart City y la propia limitación de la beca, se delimitó el estado final de las prácticas a desarrollar un prototipo funcional de la futura herramienta final partiendo de los datos cosechados durante los años 2019 y 2020.

A nivel de objetivos y supervisión, la prueba de concepto debía ser discutida a lo largo de su construcción y necesitaba contar con el apoyo de las partes interesadas, todo ello antes de llegar a la presentación de otros equipos, testear con datos en tiempo real o montar la aplicación en la plataforma corporativa (dichas acciones quedaban por lo tanto fuera del radio de acción del proyecto).

En términos de horizonte temporal, este pasó en un inicio de alcanzar sólo hasta mediados de Marzo como beca de investigación a abarcar el año escolar completo para ser presentado como Trabajo de Fin de Máster. De esta manera, se consiguió ampliar las características y capacidades de la herramienta al mismo tiempo que se mejoró la estructura estética y de viñetas, como será explicado en la sección 4.2 Herramienta de visualización descriptiva

Por último, a nivel de presupuesto se partió de un inicio de trabajo de investigación para llegar al desarrollo de un producto digital sin distribuir, por lo que realmente no existieron exigencias financieras consecuencia de alguna etapa del trabajo: el hardware utilizado no fue mayor que el disponible en la universidad o los equipos personales del grupo, y las fuentes de código o módulos utilizados para el cuerpo del programa son de *open-source* en Python y de libre licencia (lo que también supuso una ventaja sobre la agilidad en las acciones a tomar e incluso se utilizó como criterio para la elección de paquete utilizado para desplegar la información particular de contexto geográfico, como se verá también en la sección 4.2 Herramienta de visualización descriptiva)

Por último, antes de finalizar con este apartado, es conveniente realizar un breve repaso sobre el trabajo de investigación realizado por compañeros estudiantes del ICAI durante sus respectivas becas de investigación en el verano de 2020.

El propósito de aquel trabajo fue, a partir de las variables registradas durante la misma ventana temporal, tratar de inferir similitudes entre transformadores en base a su reducción dimensional a través de la técnica PCA, y poder agrupar su funcionamiento.

Durante este informe, cuyo resumen y formateado fue realizado como tarea aconsejada al autor de este proyecto para familiarizarme con la temática. Se utilizó el lenguaje de programación R para realizar un estudio particularmente analítico en el análisis exploratorio de los datos, usando desde gráficas de tipo *boxplot* para evaluar la presencia de valores atípicos, como se muestra a continuación.

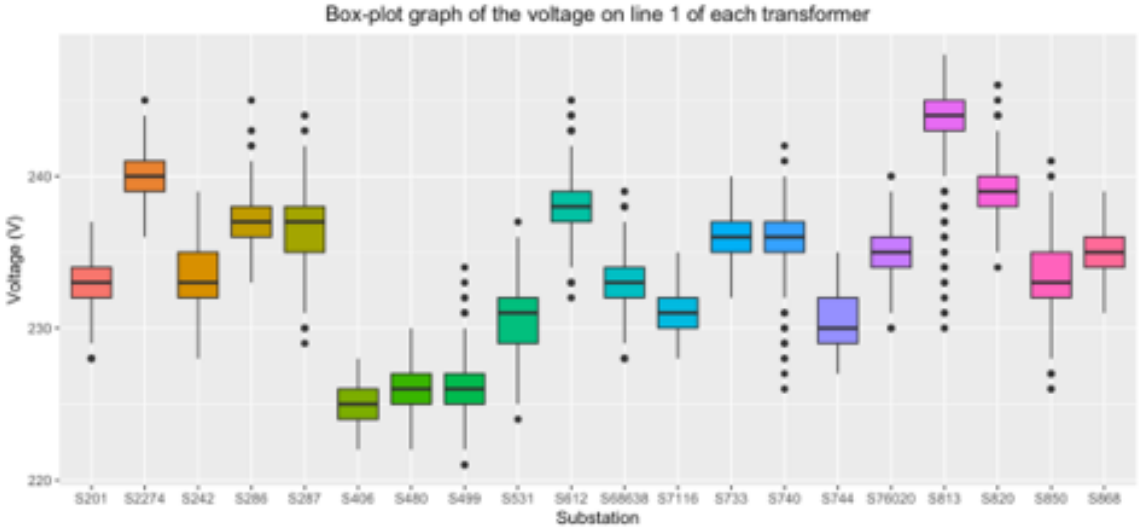


Figura 18: Ejemplo de gráfico Boxplot de tensión en linea 1 para cada CT

Se preprocesaron los datos atípicos para valores nulos, los horarios de lectura duplicados y por último, se identificaron horarios donde no se registra lectura (en particular, muchos de ellos dándose en momentos de reparación de los propios CT).

Durante el análisis *a priori*, los autores del articulo destacan una **fuerte correlación lineal entre potencia activa y corriente** donde se agrupan los datos alrededor de dos a cuatro rectas, así como una formación leve de clusters entre potencia activa y reactiva inductiva, o entre tensión y potencia activa. Se aconseja al lector el referirse a la SECCIÓN 5 para referirse al análisis particular que se realiza en este proyecto donde se hacen uso de herramientas potentes propias de Python.

Para acabar, el análisis de los centros arroja información alrededor de las posibles agrupaciones que se hacen entre CT según la orientación relativa que presentan sobre la base de Componentes Principales (P.C.). En primera instancia, se valora la combinación lineal que tienen los PC en base a las variables originales como se muestra a continuación.

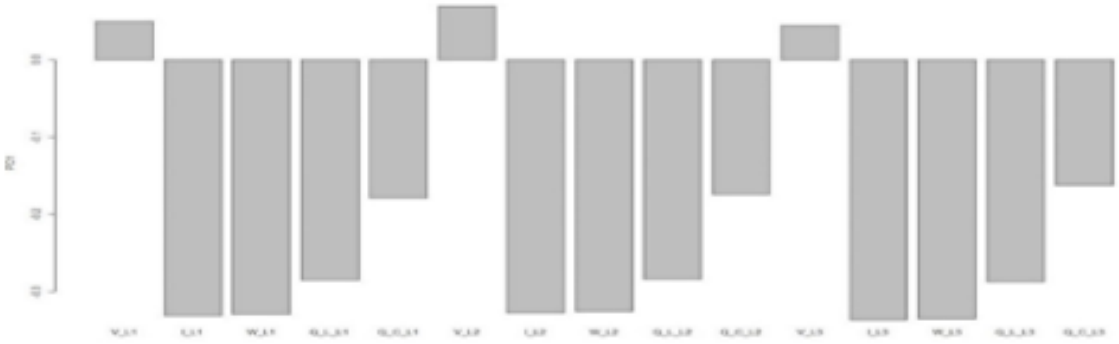


Figura 19: Primera Componente Principal

Según este estudio previo, en el PC1 existe una fuerte presencia de la intensidad, potencia reactiva (tanto inductiva como capacitiva). Esto se debe a que, al aumentar la carga asociada a un CT, aumenta la intensidad circulante y a su vez las pérdidas. La variabilidad a lo largo de esta fenómeno queda registrado en esta componente.

Según este reparto de los dos componentes principales escogidos para el análisis, acaba provocando la proyección de datos siguiente.

Figura 20: Trazado del conjunto de datos

Tras agrupar los CT según la positiva relativa al origen, la variabilidad de la nube de puntos de cada CT y orientación de la nube, se acaban distinguiendo las familias siguientes de CT: grupo 1 de CT dando soporte a **zonas residenciales urbanas de tipo apartamento** (S531, S612 y S286), grupo 2 de **niveles de carga similares** (S68638, S406 y S201), grupo 3 en **zonas residenciales del sur de Málaga** (S744, S733 y S850), grupo 4 de **fuerte influencia turística** (S2274 y S813), grupo 5 abasteciendo **colonias de pisos con perfil de consumo similar** (S242 y S7116), grupo 6 sin **similitudes aparentes** (S820 y S76020) y el grupo 7, representando la **miscelánea** y con cada CT particular.

Para finalizar el proyecto, el grupo trata de realizar un estudio sobre los desequilibrios de la red proponiendo un nuevo estudio PCA sobre las diferencias entre tensiones de líneas consecutivas. Aunque desgraciadamente, los resultados no arrojan información destacable, sí observan irregularidades alrededor de momentos donde se produjeron incidentes registrados y son fruto dos tipos de desequilibrios: de carga principalmente, o de niveles de tensión en línea de forma más leve. Estas diferencias se volvieron particularmente virulentas durante los primeros meses de pandemia, comentario que también se evoca en el estudio.

Hecho el repaso sobre este punto de partida, da comienzo a partir de este punto el comentario detallado de la respuesta en la Sección 4. y más detallado y técnico en la Sección 5. del nuevo trabajo de visualización y análisis de la herramienta desarrollada, así como el manual de uso y los principales casos identificados donde puede ser de gran utilidad.

Sección 4. IMPLEMENTACIÓN DE LAS SOLUCIONES

En el apartado que aquí da comienzo se presta atención al desarrollo de los componentes que dan entidad a la aplicación, así como al trabajo de integración que se ha llevado a cabo para la entrega de un producto digital alineado con los objetivos de la misión y en consonancia con las exigencias técnicas del proyecto.

La primera sección se reserva para el estudio cualitativo y cuantitativo preliminar que se realiza sobre la totalidad de las variables de partida de los archivos provistos sobre las mediciones del 2019 – 2020. Con ayuda de tres métodos distintos de evaluación de datos, se aprecian las relaciones que atan el comportamiento de ciertas variables físicas frente a otras, así como la propia repartición de observaciones dentro de cada una de ellas.

La segunda sección responde a una evolución cronológica del proyecto, y es que la primera mitad de visualización de la herramienta hace un trabajo más descriptivo que analítico que la segunda. Este primer resultado marca el resultado que se obtuvo a finales del primer semestre de proyecto y, más allá de escoger un tipo de gráficas antes que otras, sobre todo sienta las bases de la estructura global de la aplicación dentro de la plantilla Dash, factor clave para la continuación del trabajo.

Seguidamente, la tercera sección aborda de lleno la segunda mitad de la herramienta, encargada del trabajo de análisis profundo, y presenta el trabajo de análisis exhaustivo que se realizó alrededor de los dos problemas de aprendizaje automático previamente evocados: la reducción dimensional, y la detección de mediciones anómalas. Apoyando las elecciones técnicas sobre resultados objetivos y consideraciones subjetivas de cara a la experiencia de usuario, se detallan los bloques que componen este tomo hasta aclarar la justificación de la solución y el valor añadido que aporta.

El apartado cuarto está reservado a la explicación de la integración de los dos módulos anteriores. La explicación va más allá del código que se utilizó en el apartado dos para

representar gráficas elementales de datos, y profundiza más en los medios y métodos usados para entrelazar ambas secciones y hacer su uso intuitivo como si de una aplicación comercial se tratase.

Para acabar, el quinto título acaba enumerando el conjunto de operaciones realizadas para optimizar el uso de la herramienta, ya sea mejorando el resultado entregado en la salida o utilizando los recursos computacionales mínimos para no entrar en conflicto con la fluidez del sitio web.

4.1 ANÁLISIS EXPLORATORIO DE DATOS

El Análisis Exploratorio de Datos (E.D.A.) se corresponde con el primer proceso crítico de un proyecto donde evaluar en una investigación inicial sobre el conjunto de datos de partida posibles tendencias destacables, ya sea sobre subconjuntos de este, o características observables y de interés en la totalidad del *corpus*. Independientemente del tipo de proyecto, de dato o de problema al que un equipo se ve enfrentado, ocurre muchas veces que un estudio exhaustivo en esta fase ahorra problemas y tiempo en probar hipótesis a futuro. También permite desenredar la complejidad de los datos y arroja luz sobre las particularidades estadísticas que los definen.

Este capítulo cobra sentido sobre todo antes de atacar el problema pues es una buena práctica la de entender los datos (tanto sintáctica como semánticamente) y extraer sentido de ellos antes de dar solución al problema. Para abordar la tarea desde distintos puntos de vista, se han utilizado distintos recursos que ofrecen librerías de Python y en particular, cuatro: **Pandas**, **Sweetviz**, **Autoviz** y **Lux**.

4.1.1 ANÁLISIS PRELIMINAR

Los dos archivos que se disponen al inicio de este proyecto se denominan **Listado_Trafos.csv** y **LVSM_Def.csv**, conteniendo información propia de cada Centro de Transformación mientras el segundo alberga los valores temporales de las mediciones.

CD	TRAFO	SUBESTACION	LINEA_MT	FABRICANTE	MODELO	NUM_SERIE	AÑO	POTENCIA	UNIDAD_MEDIDA	STATO_MAT	LATITUD	LONGITUD	NOTAS	GRUPO	NUMERO
201	TR1	MIRAFLOR	S_ANTON	OASA	250/24/10-20(400)VO-PA	12794	1972	250	kVA	E	36,73112967	-4,361444268	1 Trafo	resi2	1
2274	TR1	SRAFAEL	TIROPICHON	IMEFY	1000I/24/20B2O-PE-GE-FND001	105801	2011	1000	kVA	E	36,70937527	-4,454017011	1 Trafo	resi	2
242	TR1	MONTES	PI_LIMONAR	OASA	630I/24/20B2O-PA-GE-FND001	29822	1988	630	kVA	E	36,72096715	-4,402892186	1 Trafo	resi	2
286	TR1	MIRAFLOR	LA_CERDA	ORMAZABAL	1000I/24/20B2O-PE-GE-FND001	242584	2015	1000	kVA	E	36,72787923	-4,37128178	1 Trafo	resi2	1
287	TR1	MONTES	PQ_CLAVERO	DIESTRE	630CA/24/20B2O-PA	48035	1989	630	kVA	E	36,72631026	-4,381495361	1 Trafo	resi2	1
406	TR1	PERCHEL	MUELLE_UNO	DIESTRE	630CA/24/20B2O-PA	48034	1989	630	kVA	E	36,71926536	-4,407921545	1 Trafo	resi	2
480	TR1	PERCHEL	HOTEL.	INCOESA	630I/24/20B2O-PE-GE-FND001	153597	2006	630	kVA	E	36,71989436	-4,423080293	1 Trafo	mixta	3
499	TR1	CENTRO	S_BARTOLOM	GEDELSA	630I/24/20B2O-PA-GE-FND001	29645	1900	630	kVA	E	36,72767029	-4,423206077	1 Trafo	resi	2
531	TR1	CIUDAD_J	MEDITERRAN	ALKARGO	630I/24/20B2O-PA-GE-FND001	29921	1982	630	kVA	E	36,73275141	-4,422138893	1 Trafo	resi	2
612	TR1	CIUDAD_J	LA_PALMA.	INCOESA	630I/24/20B2O-PA-GE-FND001	136555	2003	630	kVA	E	36,73947438	-4,43393217	1 Trafo	resi	2
68638	TR1	RAMOS	PUENTE.	IMEFY	630I/24/20B2O-PA-GE-FND001	44373	2001	630	kVA	E	36,72622993	-4,477380426	2 Trafos	resi2	1
7116	TR1	SRAFAEL	DUENDE	ABB	400I/24/20B2O-PA-GE-FND001	308910	2000	400	kVA	E	36,7094919	-4,464695815	2 Trafos	indus	4
733	TR1	SECUNDAR	FELIX_SAEN	IMEFY	630I/24/20B2O-PA-GE-FND001	62076	2005	630	kVA	E	36,71753734	-4,425379271	1 Trafo	resi	2
740	TR1	SECUNDAR	POLIGONO1	JARA	630I/24/20(400)VO-PA	24723	2002	630	kVA	E	36,71841364	-4,438550471	1 Trafo	resi	2
744	TR1	PERCHEL	ALMANSA	CEE	630CA/24/20(400)VO-PA	254207	1989	630	kVA	E	36,71863324	-4,43251387	2 Trafos	mixta	3
76020	TR1	PERCHEL	HOTEL.	INCOESA	1000I/24/20B2O-PA-GE-FND001	140161	2004	1000	kVA	E	36,72042988	-4,421984346	2 Trafos	mixta	3
813	TR1	SECUNDAR	MONTILLA.	ORMAZABAL	400I/24/20B2O-PE-GE-FND001	253303	2016	400	kVA	E	36,70869402	-4,440802337	1 Trafo	resi	2
820	TR1	SECUNDAR	S_A_VERS	ALKARGO	630I/24/20(400)VO-PA	29924	1992	630	kVA	E	36,71325087	-4,439646949	2 Trafos	resi	2
850	TR1	S_SEBAST	MISERICORD	DIESTRE	630CA/24/20B2O-PA	44504	1987	630	kVA	E	36,69947612	-4,448194263	2 Trafos	resi	2
868	TR1	POLIGONO	GUAD_HOR_2	MACE	400I/24/20B2O-PE-GE-FND001	27799	2008	400	kVA	E	36,68826906	-4,478807804	1 Trafo	indus	4

Tabla 3: Tabla Listado_Trafos [23]

En la tabla previa, las variables aportan particularidades definitorias de cada Centro de Transformación. En particular, destaca el código del transformador, el nombre de la subestación, el fabricante y modelo, la potencia instalada y las coordenadas geográficas. A estos últimos se añaden anotaciones en grupo que se corresponden con las conclusiones arrojadas del análisis anterior a este proyecto, comentado en la sección Alcance del trabajo, presupuesto y estado de partida.

Reading_Date	Branch	Organization	Substation	Transformer	App_SW	V_L1	I_L1	W+L1	Q_L_L1	Q_C_L1	Cos(phi)_L1	Angle_L1	V_L2
2019-06-16 01:00	AE	SZZ	S201	TR1	003F	234,00	65,00	14964,00	1988,00	0,00	0,00	7,00	235,00
2019-06-16 02:00	AE	SZZ	S201	TR1	003F	233,00	57,00	13091,00	1110,00	0,00	1,00	4,00	234,00
2019-06-16 03:00	AE	SZZ	S201	TR1	003F	236,00	55,00	12847,00	1309,00	0,00	0,00	8,00	237,00
2019-06-16 04:00	AE	SZZ	S201	TR1	003F	234,00	135,00	30517,00	4860,00	0,00	0,00	10,00	235,00
2019-06-16 05:00	AE	SZZ	S201	TR1	003F	235,00	102,00	23069,00	4399,00	0,00	0,00	9,00	236,00
2019-06-16 06:00	AE	SZZ	S201	TR1	003F	231,00	45,00	10181,00	758,00	0,00	1,00	1,00	233,00
2019-06-16 07:00	AE	SZZ	S201	TR1	003F	234,00	34,00	8045,00	733,00	0,00	1,00	3,00	235,00
2019-06-16 08:00	AE	SZZ	S201	TR1	003F	233,00	47,00	10913,00	314,00	44,00	1,00	3,00	235,00
2019-06-16 09:00	AE	SZZ	S201	TR1	003F	231,00	58,00	13161,00	3054,00	0,00	0,00	15,00	233,00
2019-06-16 10:00	AE	SZZ	S201	TR1	003F	231,00	159,00	35304,00	7840,00	0,00	0,00	13,00	232,00
2019-06-16 11:00	AE	SZZ	S201	TR1	003F	232,00	167,00	37086,00	8878,00	0,00	0,00	15,00	232,00
2019-06-16 12:00	AE	SZZ	S201	TR1	003F	232,00	78,00	17739,00	3766,00	0,00	0,00	16,00	233,00
2019-06-16 13:00	AE	SZZ	S201	TR1	003F	232,00	90,00	20543,00	4341,00	0,00	0,00	9,00	233,00
2019-06-16 14:00	AE	SZZ	S201	TR1	003F	231,00	195,00	43742,00	8016,00	0,00	0,00	8,00	232,00
2019-06-16 15:00	AE	SZZ	S201	TR1	003F	231,00	182,00	41127,00	6934,00	0,00	0,00	8,00	232,00
2019-06-16 16:00	AE	SZZ	S201	TR1	003F	232,00	72,00	16529,00	2654,00	0,00	0,00	15,00	233,00

Tabla 4: Encabezado de Tabla LVSM_Def [23]

En la imagen superior se observa un pequeño tramo de la matriz de datos de la base de mediciones temporales del proyecto. Por motivos de visibilidad, sólo es posible reflejar una sección del extenso archivo de casi 50 Mb, pero un rápido examen del archivo permite observar que la base de datos es del tipo **tabular ordenada**, incluyendo datos variables propios a un instante dado, pero también al contexto del CT. Las variables que se muestran incluyen la serie de atributos:

- **Reading_Date**: columna de tipo *date_time* que informa del momento de la medición. Será de máxima importancia en todo el proyecto para dar una ordenación al resto de datos cronológicamente.
- **Branch**: Se refiere a la zona o región de pertenencia del dato (en este caso, Andalucía Este para la ciudad de Málaga). Esta variable es exclusivamente de uso interno para seccionar fácilmente la base de datos nacional, por lo que no aportará información para este proyecto.
- **Organization**: Se refiere a Endesa Andalucía (en concreto, "Antigua Sevillana" se corresponde al código SZZ).
- **Substation**: Se corresponde con el código identificando el grupo transformador. Es una variable categórica fundamental pues conecta con la segunda tabla **Listado_Trafos.csv** donde es **Llave Primaria –** lo que hace de este atributo una llave extranjera (*foreign key*).

- **Transformer**: Es el transformador de Media Tensión/Baja Tensión que hace el salto de tensión entre la parte de Red de Transporte y la Red Doméstica.
- **App_SW**: Variable interna de la compañía, este código alfanumérico informa sobre la versión del **Software** usada para la medición.

Hasta este punto, los atributos son todos categóricos y pueden ser comunes a muchas filas en sus distintas combinaciones. Cada uno de ellos ofrece información ligada al momento o espacio de la lectura, pero no al valor registrado. Sin embargo, de aquí en adelante se suceden una serie de variables físicas que se replican para cada una de las líneas, razón por el número que las acompañan, y que se pasan a comentar a continuación:

- **V, I, W+, Q_L, Q_C**: Tensión e intensidad de la línea, Potencia activa y Potencia Reactiva Capacitiva e Inductiva.
- **Cos(phi) y Angle**: factor de potencia y ángulo de la línea.
- **Temp_amb**: Se trata de la temperatura ambiente de la instalación física.
- **A+ y A-**: Energía activa importada y exportada (en Wh)
- **R+L, R-L**, **R+C y R-C**: Componentes de Potencia importada y exportada.

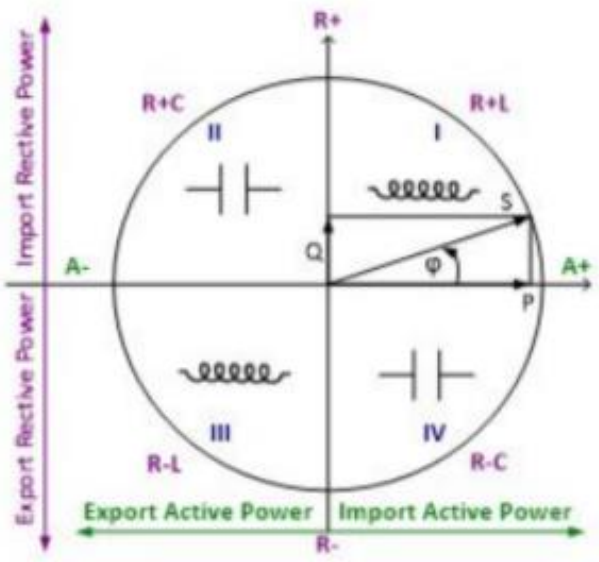


Figura 21: Esquema ilustrativo de los componentes de potencia y energía [24]

Conocida la naturaleza de los atributos, es posible ya determinar la **granularidad de la tabla**: vistas los encabezados descriptivos de los que se dispone, se establece el convenio de aquí en adelante que cada registro es **únicamente identificado** por la combinación de valores entre **Reading_Date y Substation**, pues el concebir dos medidas distintas en un mismo punto temporal y espacial dado no tiene sentido en el contexto de este proyecto (teniendo en cuenta que las columnas Branch, Organization y Transformer son atributos **monoreferenciales**, sólo cuentan con único valor respectivamente).

Por otro lado, los registros aportan información sobre el volumen de la tabla. En particular, se observa que el archivo cuenta con **167.549 registros**, que abarcan un periodo temporal de fecha **2019-06-16 a la 01:00** al **2020-06-08 a las 14:00** (duración total de casi 12 meses). A primera vista, no existe columna que parezca particularmente hueca o con valores erráticos. Sin embargo, se constata la **pobreza en calidad de dato** que tiene la columna relativa a los **ángulos**: ni los valores de coseno parecen corresponderse a los de ángulo (que están en unidad de grados, pues varía del valor 0 al 359) ni los valores sin contexto de cos(phi) aportan información (pues están restringidos a ser de valor 0 o 1 únicamente).

Como comentario preliminar, es posible anunciar que las columnas **Branch, Organization, Tranformer y cos(phi)** son atributos con valor único o semi-único, por lo que no tienen valor descriptivo para nuestro modelo. A pesar de la importancia de esta observación, Excel no ofrece mejores capacidades que las que conseguirá Pandas.

Una vez importado, es buena práctica comprobar la correcta lectura del archivo. Los errores que pueden ocurrir en este paso puede venir del codificador usado para la lectura, el delimitador del archivo o la interpretación del tipo de columna.

	date	hour	branch	organization	substation	transformer_code	App SW	V_L1	I_L1	W_L1	...	temp_amb	aplus_L1	RplusL_L1	RminusC_L1	aplus_L2	RplusL_L2	RminusC_L2	aplus_L3	RplusL_L3	RminusC_L3
0	2019-06-16	01:00:00	AE	SZZ	S201	TR1	003F	234	65	14964	...	30	16082	1983	0	16736	1620	0	23015	2179	0
1	2019-06-16	02:00:00	AE	SZZ	S201	TR1	003F	233	57	13091	...	29	14342	1441	0	14545	1057	28	23764	2906	0
2	2019-06-16	03:00:00	AE	SZZ	S201	TR1	003F	236	55	12847	...	29	13543	1381	0	14073	1141	0	22147	2942	0
3	2019-06-16	04:00:00	AE	SZZ	S201	TR1	003F	234	135	30517	...	29	20757	2954	0	22059	2021	0	27317	3701	0
4	2019-06-16	05:00:00	AE	SZZ	S201	TR1	003F	235	102	23069	...	29	29753	5054	0	31259	3121	2	33013	3778	0

5 rows × 38 columns

Tabla 5: Encabezado de Dataframe LVSM_Def

Se obtiene esta vista usando el comando **df.head()** así como la funcionalidad **read_csv()** de Pandas para importar los datos al programa. Una de las virtudes que esta librería tiene son sus métodos integrados para manipular las tablas y, a su vez, que muchas de las otras librerías que son utilizadas en este capítulo admiten *dataframes* (objeto creado en Pandas, popular en el mundo de la Ciencia de Datos) como entrada válida.

Una de las funciones de interés es el método **describe()**, que aporta una narración básica de la distribución de valores dentro de cada atributo.

	date	hour	branch	organization	substation	transformer_code	App SW
count	167531	167531	167531	167531	167531	167531	167531
unique	359	24	1	1	20	1	10
top	2020-03-02	10:00:00	AE	SZZ	S242	TR1	083E
freq	909	7008	167531	167531	8607	167531	87665

4 rows × 38 columns

Tabla 6: Análisis describe() de la tabla LVSM_Def (1)

A modo de prueba de cordura, se confirman las preconcepciones teóricas que fueron mencionadas anteriormente: **“date”** y **“hour”** son columnas extraídas manualmente en el preprocesamiento del archivo a partir de “Reading_Date” y, efectivamente, cuenta con 365 y 24 valores únicos respectivamente. Las columnas Branch, Organization y Transformer contienen un único valor y, como se vio en Listado_Trafos, se distinguen 20 centros de transformación distintos y hasta 10 versiones del *Sofware* de lectura. Llama la atención ver que esta última está muy desbalanceada (la versión 083E es la más frecuente, apareciendo en más del 50% de los casos) pero, sin embargo, los datos de CT se creen uniformemente distribuidos (pues su valor más frecuente, el CT S242, tiene un 5.1% de los registros, es decir la vigésima parte del volumen total).

	V_L1	I_L1	W_L1	QL_L1	QC_L1	cos_L1	angle_L1	V_L2	I_L2
count	164433.000000	164433.000000	164433.000000	164433.000000	164433.000000	164433.000000	164433.000000	164433.000000	164433.000000
mean	233.955271	229.200878	51290.507058	4815.429981	-1102.215942	0.422519	34.157633	234.206260	228.577481
std	4.946956	134.517281	32842.046199	6169.744172	3337.652630	0.493962	68.148608	4.918935	134.815493
min	221.000000	20.000000	-127920.000000	0.000000	-29196.000000	0.000000	0.000000	219.000000	15.000000
25%	231.000000	125.000000	27662.000000	112.000000	-1327.000000	0.000000	4.000000	231.000000	128.000000
50%	234.000000	199.000000	45173.000000	2265.000000	0.000000	0.000000	8.000000	235.000000	200.000000
75%	237.000000	300.000000	69113.000000	7441.000000	0.000000	1.000000	18.000000	238.000000	294.000000
max	248.000000	1089.000000	252547.000000	43387.000000	16371.000000	1.000000	360.000000	248.000000	1240.000000

Tabla 7: Análisis describe() de la tabla LVSM_Def (2)

El método describe() es fundamentalmente distinto para variables numéricas que para categóricas. De esta visualización preliminar se aprende que la tensión es una variable poco variable alrededor de su media (233 V), la intensidad fluctúa de manera más considerable hasta alcanzar extremos más altos, que la potencia activa y potencia reactiva inductiva tienen valores tanto positivos como negativos mientras que la reactiva capacitiva está capada en el cero. Estas características se replican en las columnas análogas del resto de líneas.

De aquí en adelante, se proponen un conjunto de librerías que agilizan y potencian el proceso del Análisis Exploratorio de Datos, ofreciendo visuales de interés para resumir las características más fundamentales de los datos

4.1.2 LIBRERÍA SWEETVIZ [25]

Sweetviz es una librería Python de código libre que genera visualizaciones de alta densidad con los que empezar el E.D.A. Estos forman informes automáticos entregables en el buscador de Internet como archivos HTML. Como cualquier otra librería de Python, se usa la funcionalidad **pip** para importar el módulo en un Jupyter Notebook donde explorar la información.

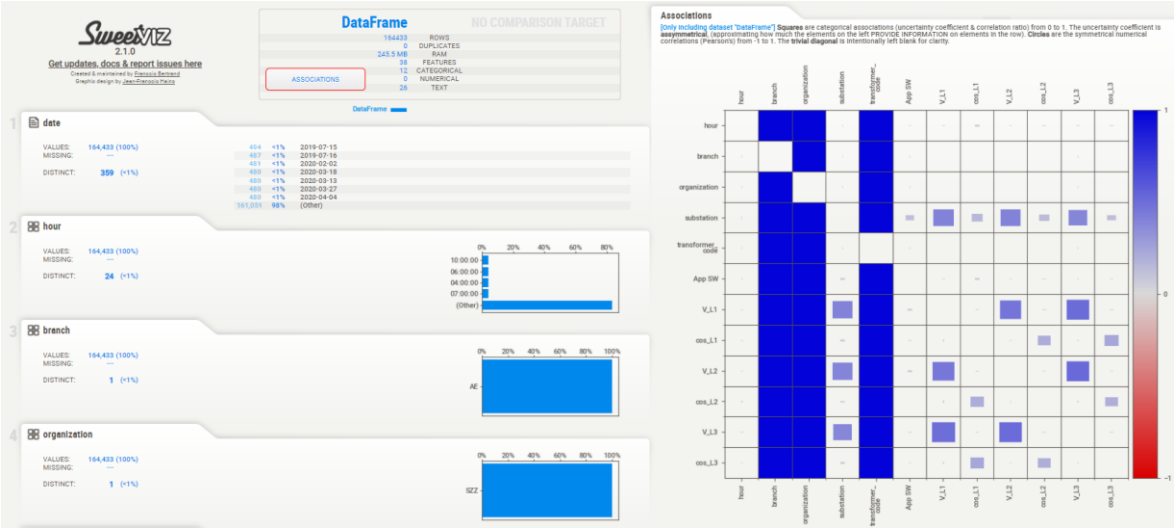


Figura 22: Informe explorador de LVSM_Def según Sweetviz

En la figura anterior, se observa la página de entrada principal. En la parte de la derecha, se muestran las asociaciones subrayadas como las correlaciones entre variables (seleccionadas entre las 38 posibles). Similarmente, en la parte izquierda por medio de la selección es posible evaluar las características de cada atributo, mostrando conclusiones estadísticas sobre cada uno. A continuación, se pasa a ilustrarlos:

- Date, Hour: se reparte uniformemente sobre todas las fechas del año, sobre todas las horas del día.
- Substation: Los CT se reparten uniformemente de manera asombrosamente perfecta, por lo que las medidas son balanceadas a lo largo de todos ellos.
- App_SW: Protagonismo (más del 90% de registros) se acogen a las versiones 083E y 003F. A partir de 12/2019, más del 85% se asocian con el valor 083E.
- V_L1: la tensión se reparte centrados en el 235V, como era de esperar.

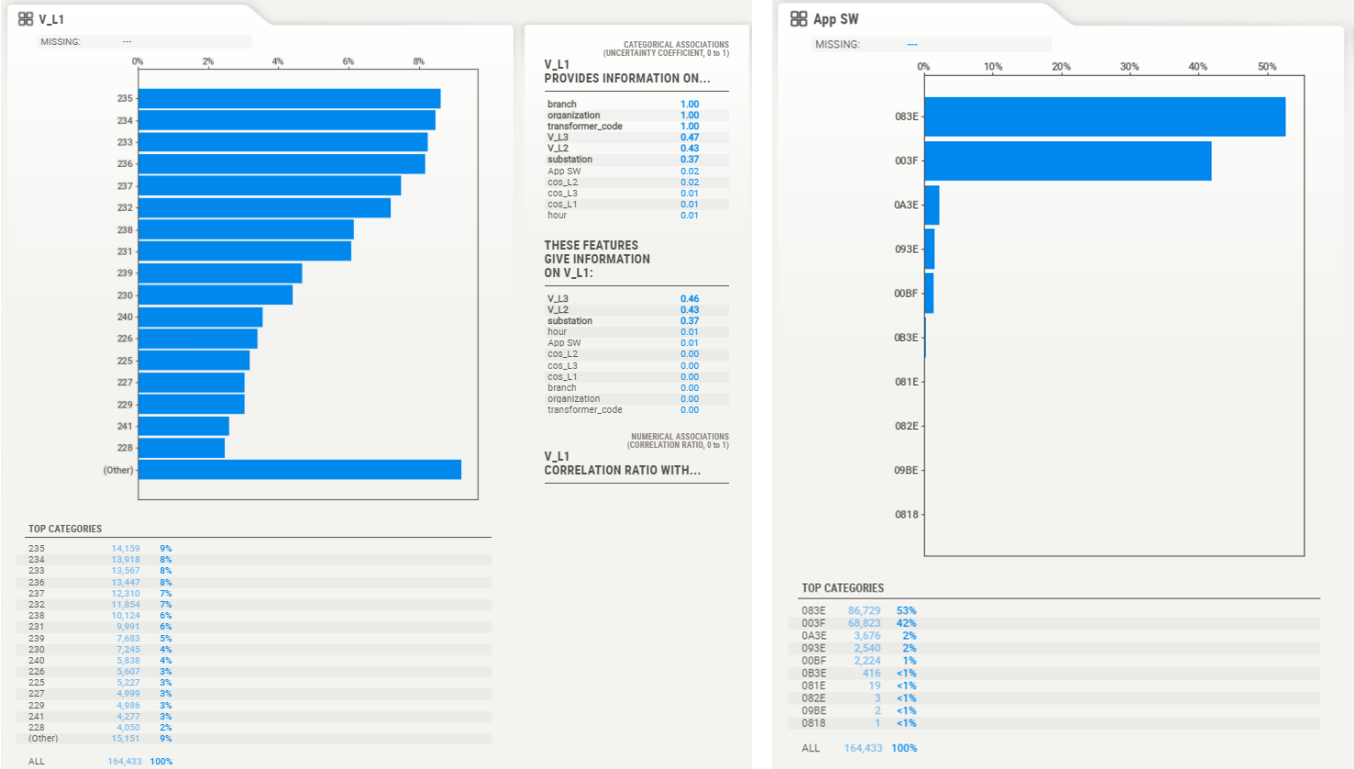


Figura 23: distribución de valores en V_L1 y App_SW

- I_L1: valores muy diseminados, ninguno de ellos alberga más del 1% de los valores.
- La distribución de a_plus están distribuidos de manera muy distinta en las tres líneas, fruto de la naturaleza trifásica frente a desequilibrios.
- Los valores de RplusL y RminusL decaen exponencialmente desde el valor 0 en frecuencia en las tres líneas.

Sin embargo, gracias a Sweetviz también cabe destacar la posibilidad que existe de comparación entre dos subconjuntos de mismos atributos. Para ilustrar esta idea, se comparan a continuación la diferencia existente entre un CT en particular y el resto de ellos.

Figura 24: Comparativa 2D entre datos del CT S201 y el resto

La observación más sorprendente es la desigualdad existente entre valores de tensión que, a pesar de estar centrados en los mismos valores regulados, tienen una distribución muy distinta y pueden marcar la caracterización del CT en particular.

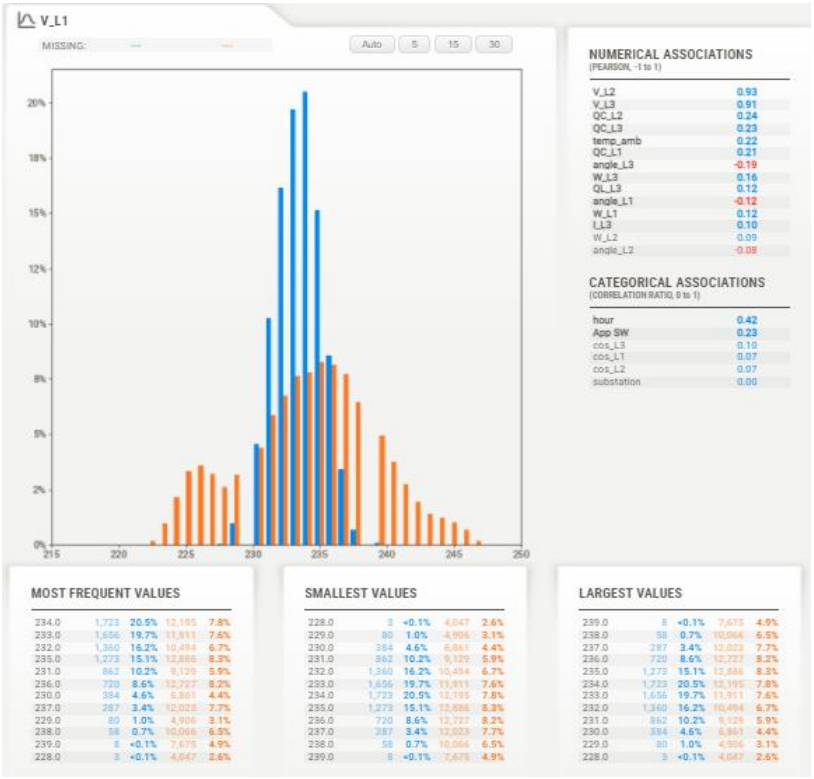


Figura 25: Comparativa entre tensiones en Sweetviz

Por último, este estudio aborda el análisis de correlación que se da entre los valores numéricos de ciertas variables, propias de la electrónica de potencia, basándose en los datos del conjunto anterior excluyente del CT 201.

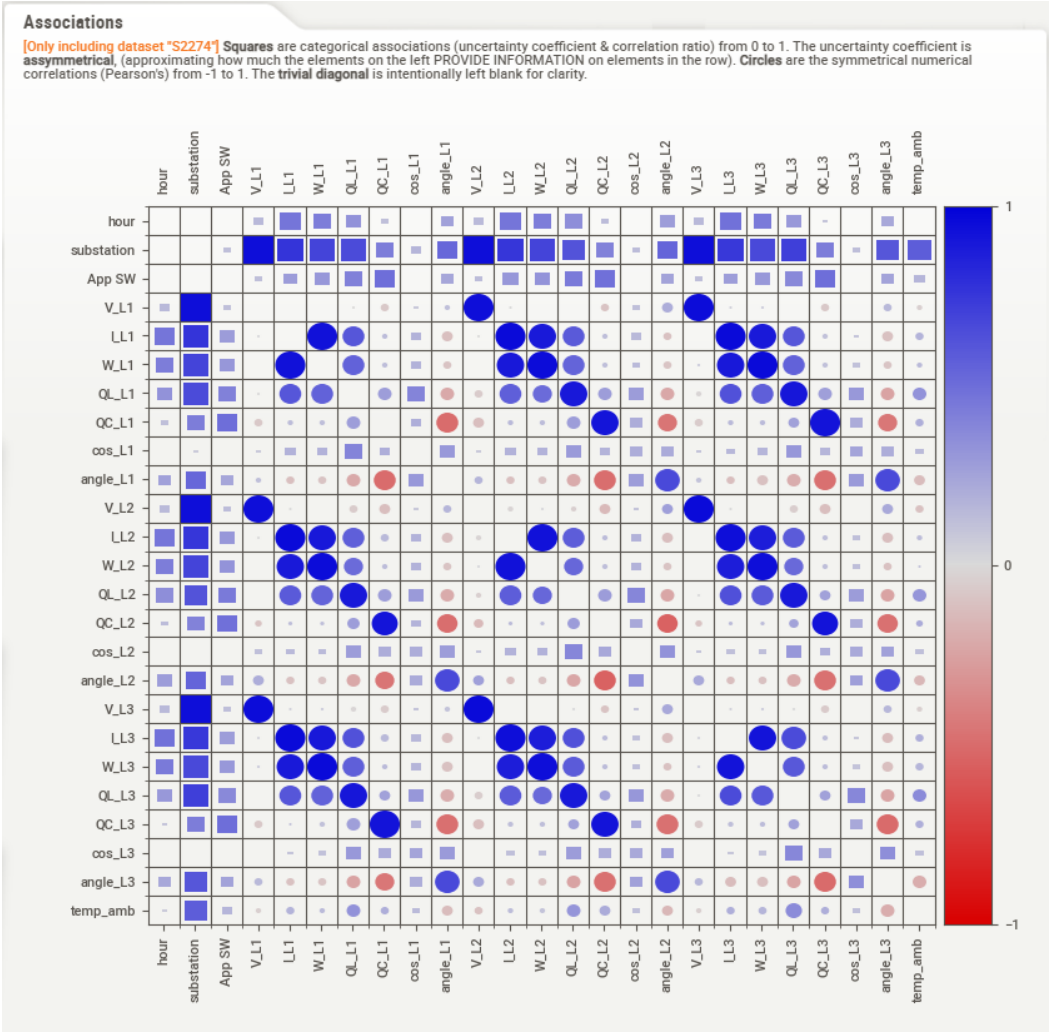


Figura 26: Matriz de asociación entre variables físicas medibles

Existen aquí correlaciones de interés presentes que merecen un comentario:

- Correlación débil entre intensidad y la hora o la temperatura ambiente, constatando la estacionalidad que existe a nivel diario (en funciones como la calefacción, el transporte, la actividad industrial, …)
- Correlación fuerte entre valores de tensión de distintas líneas, siendo esto fruto del reequilibro del sistema llevado a cabo a tiempo real en el sistema.
- Correlación fuerte entre intensidad y potencia activa (los cuales, teóricamente, guardan una relación cuadrática proporcional a la resistencia).

- Asociación muy destacable entre tensión y el código de la subestación (se concluye que existe un potencial de identificación bastante fuerte con respecto al CT donde se enfoque), como también ocurre con otras variables, como las de intensidad y potencia.
- Correlación moderada entre potencia activa y potencia reactiva capacitiva.

Estas observaciones son valoradas en última instancia para la selección de atributos del trabajo.

4.1.3 LIBRERÍA AUTOVIZ [26]

El propósito de esta librería es automatizar la tarea de A.E.D. para su análisis visual de variables, considerando su naturaleza. Basado en la fuerza bruta, esta librería tiene interés en conjuntos de tamaño pequeño pues produce con una sola línea de código gráficos de emparejamiento sobre todas las combinaciones posibles para que el usuario escoja la de mayor interés. En nuestro proyecto, se arrojan cientos de gráficos y por requerimiento computacional no ofrece la mejor prestación. A continuación, se muestran los resultados.

```
Shape of your Data Set: (164433, 38)
############# C L A S S I F Y I N G  V A R I A B L E S  ###################
Classifying variables in data set...
    Number of Numeric Columns =  28
    Number of Integer-Categorical Columns =  0
    Number of String-Categorical Columns =  2
    Number of Factor-Categorical Columns =  0
    Number of String-Boolean Columns =  0
    Number of Numeric-Boolean Columns =  3
    Number of Discrete String Columns =  2
    Number of NLP String Columns =  0
    Number of Date Time Columns =  0
    Number of ID Columns =  0
    Number of Columns to Delete =  3
    38 Predictors classified...
        This does not include the Target column(s)
        5 variables removed since they were ID or low-information variables
Since Number of Rows in data 164433 exceeds maximum, randomly sampling 150000 rows for EDA...
28 numeric variables in data exceeds limit, taking top 30 variables
Number of All Scatter Plots = 406
```

Figura 27: Ejecución del generador de gráficos (n = 406)

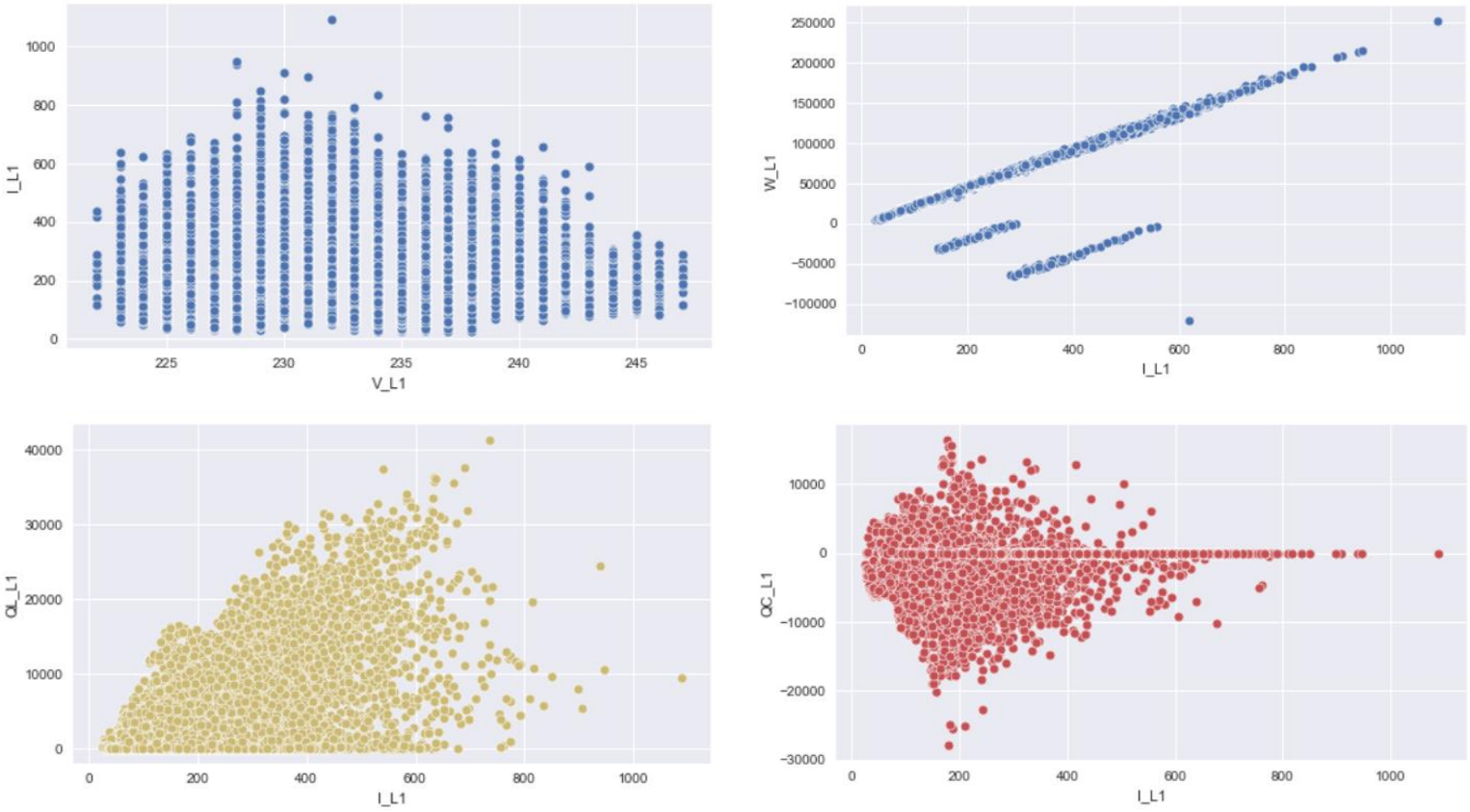


Figura 28: Gráficos Autoviz

Las gráficas de mayor interés son representadas frente a la intensidad. Se observa que las tensiones sufren de un error de cuantización debido a la medida a la escala de un voltio, lo que no evita que sea tratado como una variable numérica en lugar de categórica (pues mantiene la propiedad de ser ordenable), pero será de interés en la visualización.

En términos de asociaciones y de diferencias de escala, la intensidad es mucho más variable (tanto absoluta como relativamente), consecuencia de la regulación de operación que sufre la tensión en la red europea alrededor de 230V. La potencia activa sigue relaciones directamente proporcionales con la intensidad en distintos tramos, dando el salto cuando la potencia se vierte a la red desde la carga. La potencia reactiva inductiva es siempre positiva y creciente en magnitud y variabilidad con la intensidad y la potencia reactiva capacitiva está sobrerrepresentado en el valor nulo, y negativo por entregas de energía por grupos de batería y condensadores en el sistema.

También se obtienen gráficas de distribución de frecuencia dimensión a dimensión como el de la intensidad, que está sesgada a la derecha (la media es superior a la mediana) a consecuencia del peso que tomas las sobre corrientes hacia los valores más extremos (en el *boxplot*, son aquellos situados más allá de la pestaña Q3 + 1.5*IQR).

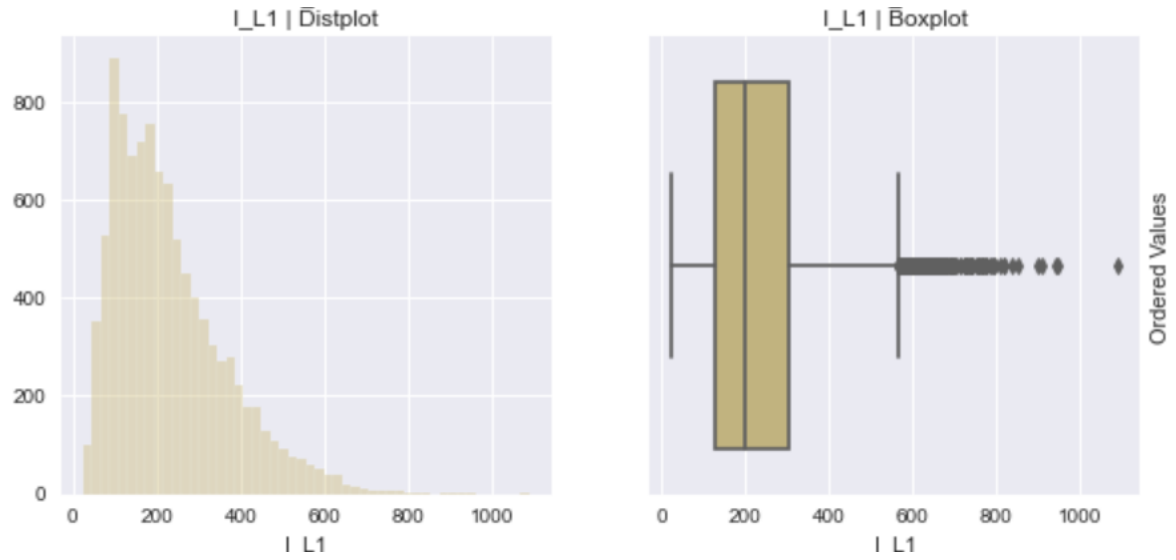


Figura 29: Histograma de Intensidad en Linea 1

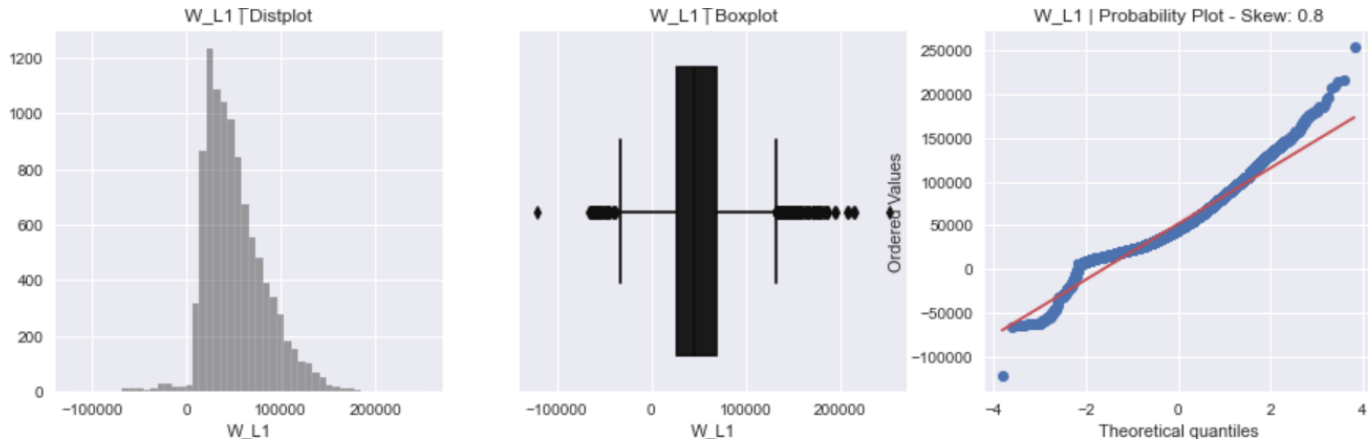


Figura 30: Histograma y Distribución de Potencia Activa en linea 1

Los valores negativos son residuales frente a los positivos, los cuales decaen progresivamente según una regla normalizada que atestigua el valor alto de "Skew".

4.1.4 LIBRERÍA LUX [27]

Lux permite el visionado de gráficos enfrentando variables a través de una Interfaz de Usuario en línea dentro del Notebook con una guía de acciones que encamina a subrayar las figuras más interesantes para el usuario. Soportado por inteligencia artificial, la libraría sigue en desarrollo y sobre las visualizaciones no se obtiene la misma información que en las anteriores, aunque su usabilidad es mucho más notable.

```
• Large dataframe detected: Lux is only visualizing a random sample capped at 30000 rows.
• Large scatterplots detected: Lux is automatically binning scatterplots to heatmaps.
```

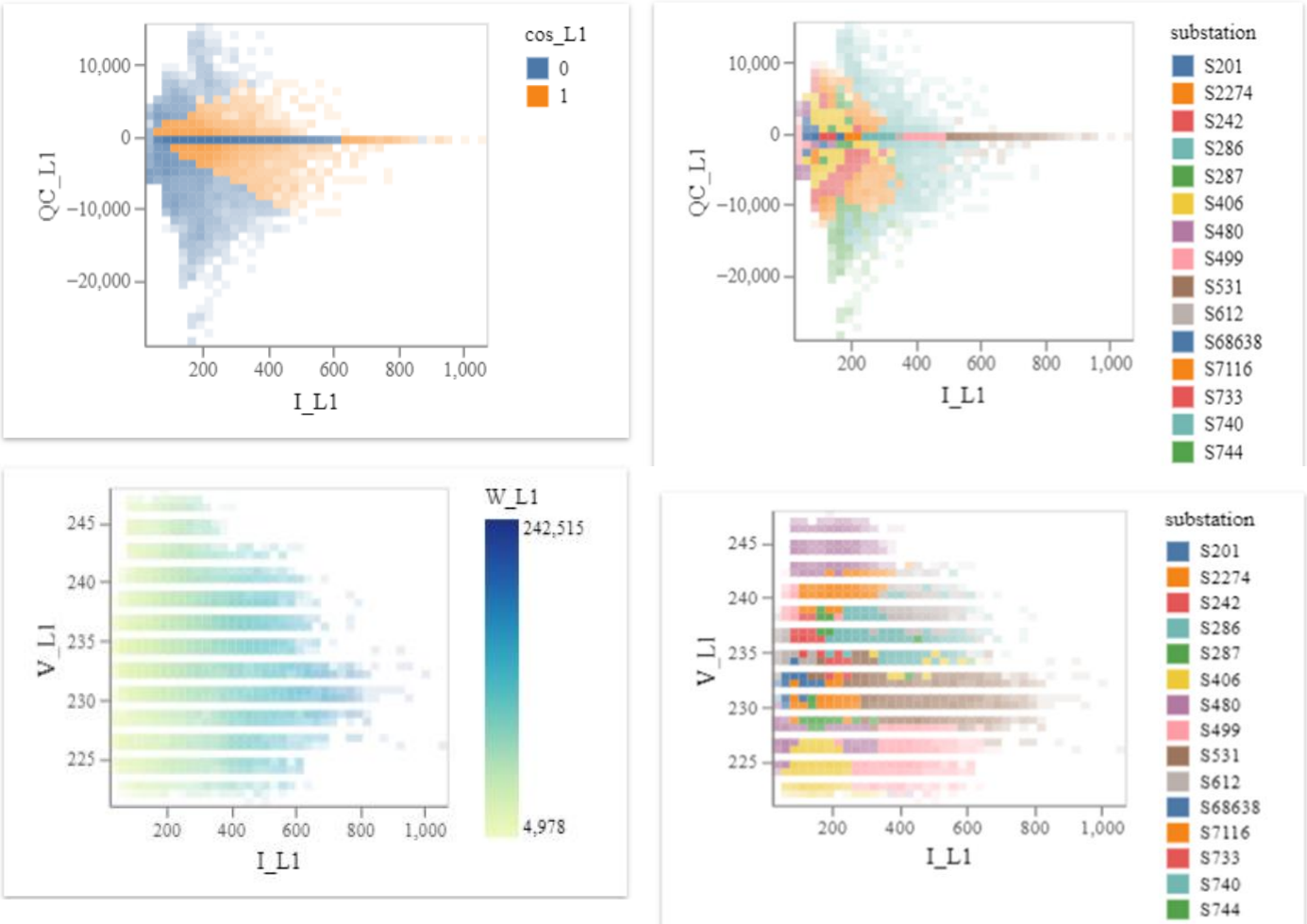


Figura 31: Gráficas Lux

De las gráficas anteriores se aprenden nuevas características no apreciadas en las anteriores librerías. En particular, los valores de cos(phi) se separan claramente frente a los valores de intensidad y de potencia inductiva (lo que implica una relación de causa o bien de consecuencia sobre la potencia, a partir de intensidad y de coseno) y la potencia activa es más dependiente del valor de la magnitud de intensidad, notable por la diferencia en escala que tiene respecto a la de tensión entre el mayor y menor de sus registros.

Además, se observa claramente por primera vez el potencial que tiene la segmentación de lecturas en base al Centro de Transformación. La nube de puntos es diferenciada según los colores de la leyenda y determina las diferencias que existen entre características de cada grupo transformador, y que será utilizado más tarde en el apartado de visualización y análisis.

4.1.5 PREPROCESAMIENTO Y SELECCIÓN DE ATRIBUTOS

En base a las relaciones anteriores, se toman una serie de acciones dentro del código que tendrán como propósito el adaptar, transformar y mejorar la forma y contenido de nuestras bases de datos, para hacerlas interpretables y manejables, pero también para seleccionar aquellos atributos que valgan la pena para encauzar la herramienta de visualización.

El primer paso del refinamiento de la tabla pasa por la limpieza de columnas, manteniendo aquellas que serán conservadas para la aplicación. Se trata aquí de dejar fuera atributos sin interés (por definición sistemática) o sin contenido (valores vacíos, sesgados o muy concentrados en pocos valores discretos) para mantener un estudio consistente de resultados, aunque conservadores, realistas (sin entrar en conocimiento experto del mundo de la electrotecnia). Se eliminan las columnas **A+, A-, R+L, R-L, R+C, R-C y cos(phi)** para las 3 líneas, acabando así con 21 variables de las 38 iniciales. Estas columnas tienen una mala repartición de valores, pésima calidad del dato o mala interpretabilidad.

El segundo paso del preprocesamiento de datos consiste en el formateado y normalización de datos. Según el caso, se explicita la naturaleza de los datos a tipo (Time, Branch, Organization, Substation, Transformer code y App SW) o de tipo **float** (variables de cada línea para tensión, intensidad, potencias y ángulo, así como la temperatura ambiente).

Por último, queda transformar "time" a **datatime, de la librería Pandas**. Sin embargo, no es posible ingresar directamente la columna en el constructor, pues no admite horas del tipo "24:00". Para sortear el problema, se importa en un primer momento como **string** para luego transformar los casos de "24:00" a "00:00" y, para dichos casos, incrementar el valor del día en uno (como objeto tiempo, no como string para evitar valores como podría ser el día 32). Seguidamente, ya sí se ingresa la columna "time" como entrada del constructor de datatime.

Una vez se han tratado las columnas, se tratan los registros. Primero, es necesario evaluar los **elementos vacíos** de la base de datos y valorar la acción a tomar. Si fueran muy numerosos, los métodos deben ser de sustitución por la media del atributo según agrupaciones, o por una estimador a partir del resto de columnas. Afortunadamente, en el archivo LVSM_Def son detectados **227 filas** con algunos de los valores estando vacío (<0.1%). Por lo tanto, se opta por **eliminar los registros vacíos** en alguna de las columnas.

Con la misma lógica, se evalúan los **registros duplicados** en los atributos identificativos de la tabla. Estos atributos para la tabla LVSM_Def son los atributos "date" y "Transformer_code" ya que no tiene sentido realizar múltiples mediciones coincidentes en tiempo y espacio. En total, se detectan **2950 duplicados** (< 1%) que se decide mantener en la forma procesada de la tabla el primero de ellos y **eliminar los siguientes**.

En este punto, se realiza una réplica de la tabla de datos (llamada hasta ahora "data") en una segunda variable denominada "data_new". Este punto marca un salvaguardado del conjunto de datos, para que las operaciones realizadas hasta este punto se realicen en ambos y a partir de donde realizar cambios en la segunda tabla, adaptándola a visualizaciones que requieren distintas columnas. En particular, dentro de la variable data_new se descomponen dos nuevas columnas, llamadas "**date**" y "**hour**" para el posterior análisis y fácil etiquetado. Además, esta copia de variable era especialmente interesante durante el proceso iterativo del proyecto, pues permitía mantener los datos limpios en una copia segura. Se manipula uno mientras se prueba la aplicación sobre data_new y, si se comete un error irreversible en los datos, en lugar de tener que pasar por la importación de datos y preprocesamiento de nuevo, sólo es necesario lanzar la célula de sobreescritura de **data_new.**

4.2 HERRAMIENTA DE VISUALIZACIÓN DESCRIPTIVA

En este apartado se muestra el desarrollo de la dimensión visual que en primera instancia se desarrolló durante este proyecto. Primero en formato Notebook en línea dentro de la plataforma Colab para programar en equipo, más tarde alojando la construcción de gráficas en el programa que hace uso de Dash, este apartado desarrolla el comentario visual del programa en un sentido cronológico de los avances.

Primeramente, se abordó el diseño esquemático del Dashboard. No es posible comenzar la programación de la aplicación sin un plan de trabajo, y es imposible diseñar un plan de trabajo sin conocer cuál es el resultado al que se desea llegar. El primer paso que se dio en la evolución del proyecto fue el intercambio de ideas que se dio sobre el boceto del producto digital. Tras una serie de iteraciones internas al equipo, y después junto al socio de Smart City, se convino una plantilla similar a la mostrada a continuación.

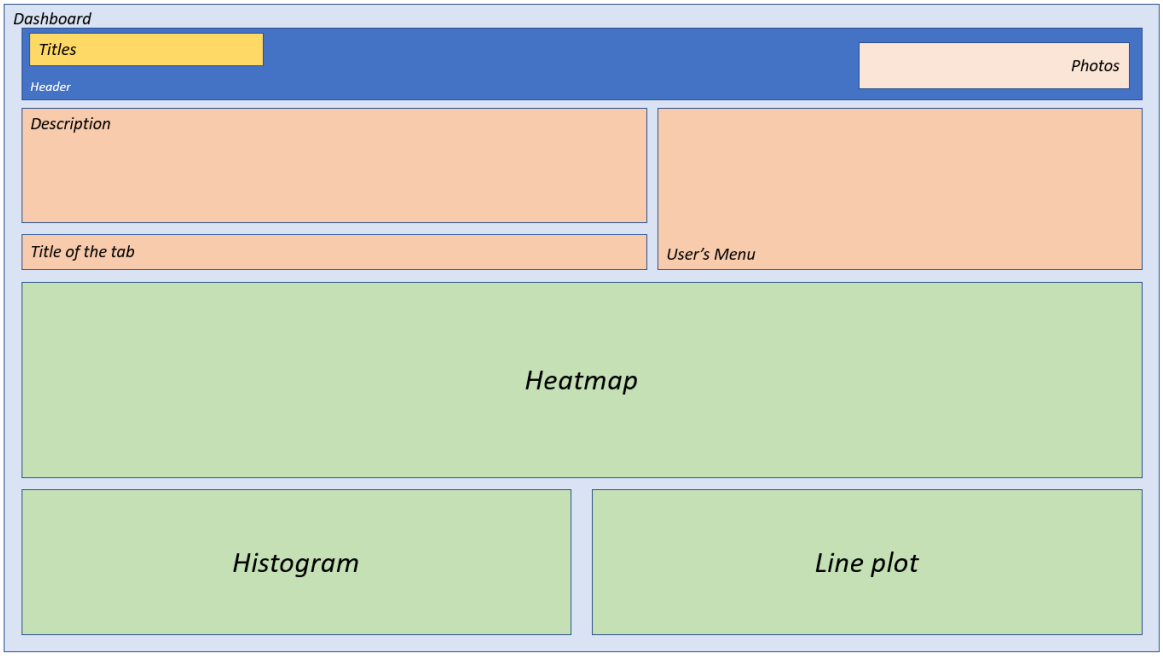


Figura 32: Esquema conceptual del tablero

Dentro del archivo inicial de Colab, se puso sobre código la importación, preprocesamiento, limpieza y formateo de ambas tablas. Una vez esto conseguido, se decidió testear la creación de gráficos uno a uno y, en particular, se probaron distintas versiones de las que poder elegir la mejor versión, con sus pros y contras.

La primera de las versiones se basaba en Plotly. Plotly es la compañía tecnología canadiense propietaria de múltiples herramientas de visualización y analítica, entre las que se encuentra Dash (plataforma donde se construirá la aplicación) o **Plotly Express** (módulo gráfico dedicado a la generación de diagramas). Gracias a su programación de bajo nivel y a la gama de argumentos de entrada que dispone, se realizó la configuración a continuación.

```
fig = px.density_mapbox(obs_values, lat='lat', lon='long', z=variables_dict[obs_var], radius=25,
                        center=dict(lat=36.72016, lon=-4.42034), zoom=12, hover_name='substation', hover_data=['manufacturer','power'],
                        mapbox_style="stamen-terrain")

fig.show()
```

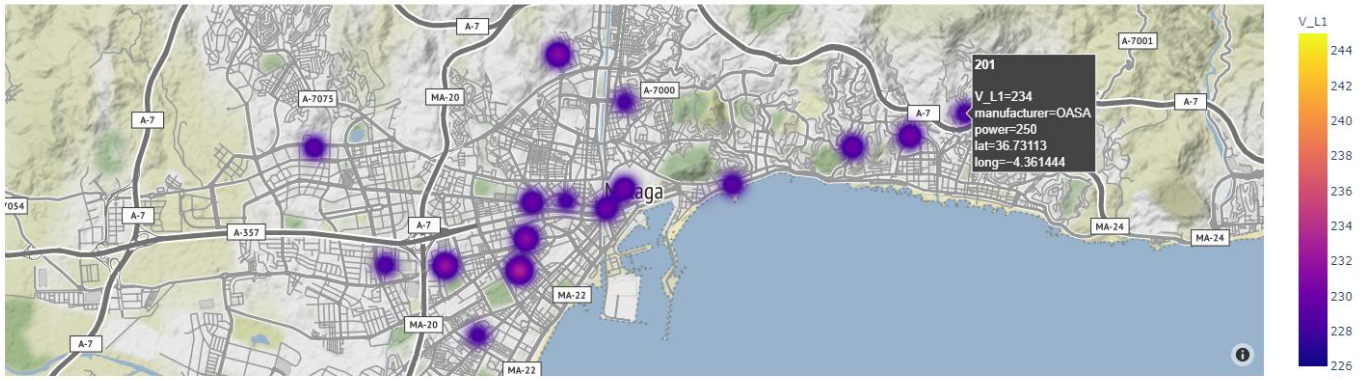


Figura 33: Gráfica Heatmap con Plotly Express

La figura goza de gran interactividad. Como se puede ver en la figura, es posible añadir cualquier información a la etiqueta que señala al punto de calor al pasar el cursor cerca (denominado *hover label*), al igual que existen capacidades en la esquina superior como la captura de pantalla para guardar la imagen como .jpg o el *zoom*.

Esta solución era doblemente interesante al saber con certeza que no tendría problemas de compatibilidad con la propia plataforma Dash, al ser también un método interno.

La segunda opción consistía en pintar todo sobre un lienzo de **Geopandas**. Esta librería es fruto de un proyecto de código libre que pretende conseguir una implementación de datos geoespaciales en Python más sencilla. Inspirado en el artículo [28], la solución es elegante y atractiva y se le da una oportunidad. Sin embargo, apenas antes de empezar el autor indica que es necesario conseguir un mapa vectorial del "telar" sobre el que se desean trazar dibujos y polígonos. Se trató de encontrar uno de la ciudad de Málaga. Sin embargo, esto ya suponía recurrir al servicio de terceros, cuya calidad dependía del precio que estuviera el equipo dispuesto a pagar y, además limitaba el espacio de visualización al perímetro del polígono (detalle que afeaba la aplicación y que no ocurría en Plotly). Por esta razón, **se descartó esta solución**.

La tercera y última opción se hace uso de llamadas API a la aplicación de Google Maps para, haciendo uso de su servicio de geolocalización, superponer los mapas de calor que se desean. De esta manera, se suma a la visualización todas las funcionalidades propias de la herramienta del gigante americano. Sin embargo, se constató que el servicio es de tipo freemium (las funcionalidades costaban más cuanta más capacidad se pidiese y se cobran tasas a partir del número de usos mensual).

```
# Initial model - THIRD VERSION, using gmaps
!pip install gmaps

from IPython.display import display, HTML
from ipywidgets.embed import embed_minimal_html

import gmaps
import gmaps.datasets

my_key = 'AIz
gmaps.configure(api_key=my_key)

# Using heatmap
locations = obs_values[['lat', 'long']]
weights = obs_values[variables_dict[obs_var]]
fig = gmaps.figure()
fig.add_layer(gmaps.heatmap_layer(locations, weights=weights, point_radius=18.0))
fig

embed_minimal_html('solution.html', views=[fig])
display(HTML(filename='solution.html'))
```

Figura 34: Generación Python y llamada a API GMaps

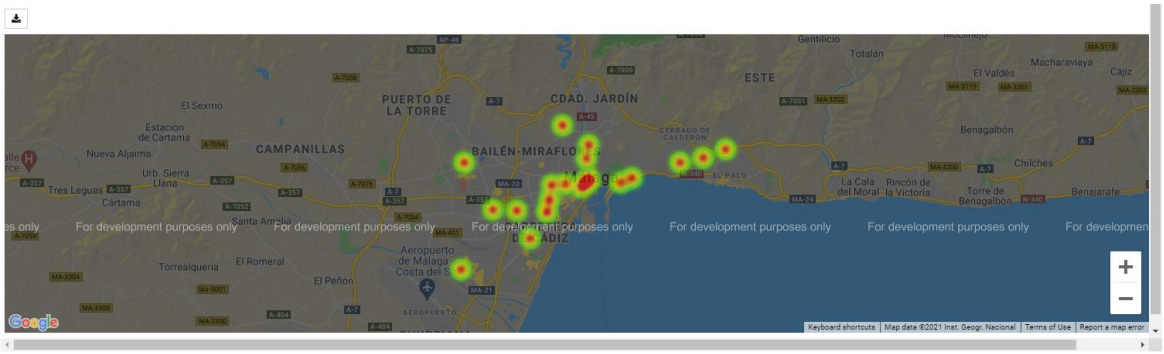


Figura 35: Heatmap GMaps

La solución es satisfactoria visualmente. Sin embargo, los modelos de pago y marcas de agua van en contra de la experiencia de usuario, así como la presencia de una clave privada en el código que liga la aplicación a una cuenta personal, volviéndola vulnerable e inestable. Por lo tanto, esta opción es descartada y se elige **Plotly Express para representar el mapa.**

Por último, en esta sección se desea utilizar al máximo el potencial de la plataforma Plotly, un marco de trabajo para crear paneles de análisis y visualizaciones customizada de datos de entorno profesional y diversas posibilidades, sin coste asociado. Como anuncia su eslogan, este servicio de software es especialmente potente pues tiene como misión la de convertirse en la herramienta estándar de *Front-end* para proyectos basados sobre modelo de ML, y con probar algunas demostraciones de la página es fácil ver lo que es capaz de conseguir. [29]

Dentro de la plataforma Dash, se distinguen **tres familias de componentes fundamentales** sobre las cuales va montada toda la aplicación:

- **Dash_html_components**: Este grupo de objetos forman las acciones atómicas necesarias para generar elementos en la aplicación. Ya sea un título, una imagen, link o botón, esta librería tiene un componente por cada uno de sus análogos en HTML y permite renderizarlo en Python. Cabe destacar también la generación de Divisiones (html.Div), los cuales son contenedores que almacenan elementos en su interior y los distribuye en la página como si fueran bloques (similares a los de la Figura 32: Esquema conceptual del tablero), aparte de aplicar un estilo común a todo lo que quede dentro.

- **Dash_core_components**: Esta sub-librería incluye un conjunto de componentes de nivel superior, más complejos, como menús desplegables, ventanas para gráficos, bloques de contenido… El componente usa Plotly.js para representar gráficos de datos interactivos de alto nivel.
- **Dash_bootstrap_components**: Dash también permite la generación de elementos propios de la librería Bootstrap. Este *framework* CSS de código libre está destinado principalmente al soporte del *Front-End* de desarrollo web para dispositivos móviles. Contiene plantillas de alto nivel sobre parámetros como la tipografía, formas, botones, navegación y otros componentes de interfaz.

Cada uno de estos elementos son descritos enteramente a través de sus atributos (siendo Dash de tipo declarativo) y los de mayor interés en la primera versión serán el uso de *heatmap* para representar y *checkbox* o sliders para escoger el deseo de usuario.

La sintaxis general de cualquier aplicación Dash consiste en la consecución de los siguientes pasos: 1) **Importar las librerías** y frameworks de Python, 2) **Procesar la información** de partida, 3) Construir la **plantilla (*layout*) de la aplicación**, con elementos de entrada y salida en su interior y 4) Especificar en las funciones finales (denominadas ***callbacks***) las **interacciones que existirán en la página web** frente a eventos del usuario. En las divisiones a continuación se especifican como se han realizado cada una de estas partes.

4.2.1 IMPORTACIÓN DE LIBRERÍAS Y PREPARACIÓN DE DATOS

Entre las librerías esenciales, se encuentran **Pandas** y N**umpy** para el manejo y gestión de conjuntos de datos, **matplotlib.pyplot** y **Plotly.express** para la generación de gráficas y figuras, y por último librerías de **Dash** y dependientes (**Dash_html_components**, **Dash_core_components** y **Dash_bootstrap_components**) para manejar los bloques de la aplicación.

En lo relativo al trabajo de adaptación del archivo bruto de datos, se implementan las acciones mencionadas en el apartado 4.1.5 Preprocesamiento y selección de atributos.

4.2.2 CONSTRUCCIÓN DE LA PLANTILLA

Este paso es imperativo en todas las aplicaciones de Dash y usa como punto de partida un esquema conceptual de la herramienta como es la Figura 32: Esquema conceptual del tablero.

La plantilla Dash comienza con la construcción de la aplicación Dash y del servidor.

```
### DASH LAYOUT PREPARATION
# App initialization
app = dash.Dash(__name__,
    title='Malaga Visual Tool',
    external_stylesheets=[dbc.themes.CERULEAN],
    meta_tags=[
        {"name": "viewport", "content": "width=device-width, initial-scale=1"}
    ]
)

server = app.server
app.config.suppress_callback_exceptions = True
```

Figura 36: Construcción de la app y servidor

Dentro del constructor, se incluyen una serie de argumentos que aplican al comportamiento de la página. El atributo ***__main__*** se utiliza para ejecutar el lanzamiento de la aplicación en la función que, como buena práctica, se recomienda poner siempre al final del código.

```
if __name__ == "__main__":
    app.run_server(debug=True)
```

Figura 37: Lanzamiento del servidor que soporta el Dash

En **title** se incluye el nombre que aparecerá en la pestaña de Chrome para reflejar algo más adecuado que el valor por defecto “Dash”. **External_stylesheets** llama a un diccionario de estilos del mencionado nombre – en este caso, CERULEAN – que facilita el trabajo de conseguir armonía dentro del canon de la aplicación y que determinará el estilo de todo aquel objeto del tipo **Dash_bootstrap_components**. Entre las posibilidades, es sencillo ir a la página web de Bootstrap para seleccionar el motivo y paleta de color de la mayoría de

elementos. La alternativa para el resto de los componentes, como indica la documentación de Dash, crear una carpeta llamada "**assets**" en el mismo directorio que el archivo Python de la app, y donde el motor de Dash acude para recuperar las reglas que aparecen en el archivo de tipo **.css** y que modeliza el estilo de los objetos HTML. Dentro de assets también se precisa depositar los archivos gráficos que aparecen estáticos en la aplicación (como son los logos de CIC y Smart City Málaga).

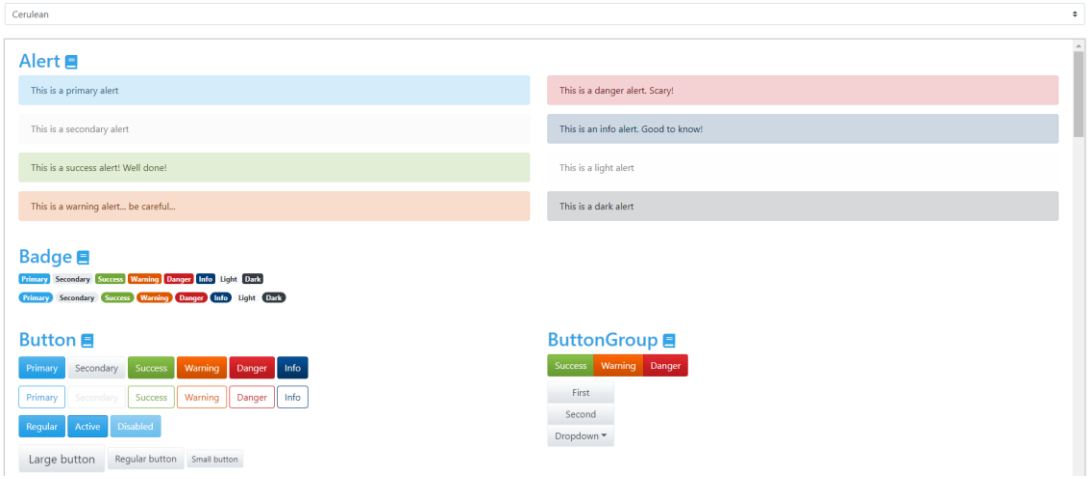


Figura 38: Explorador de temas de Bootstrap y de CELURIAN en el proyecto [30]

```
/* Base Styles
-------------------------------------------------- */
/* NOTE
html is set to 62.5% so that all the REM measurements throughout Skeleton
are based on 10px sizing. So basically 1.5rem = 15px :) */
html {
  font-size: 62.5%; }
body {
  font-size: 1.5em; /* currently ems cause chrome bug misinterpreting rems on body element */
  line-height: 1.6;
  font-weight: 400;
  font-family: "Open Sans", "HelveticaNeue", "Helvetica Neue", Helvetica, Arial, sans-serif;
  color: rgb(50, 50, 50); }


/* Typography
-------------------------------------------------- */
h1, h2, h3, h4, h5, h6 {
  margin-top: 0;
  margin-bottom: 0;
  font-weight: 300; }
h1 { font-size: 4.5rem; line-height: 1.2;  letter-spacing: -.1rem; margin-bottom: 2rem; }
h2 { font-size: 3.6rem; line-height: 1.25; letter-spacing: -.1rem; margin-bottom: 1.8rem; margin-top: 1.8rem;}
```

Figura 39: Extracto del archivo css de la carpeta assets

Por último, los pares llave – valor que se añaden al diccionario **meta-tags** son útiles para la adaptación del contenido a la resolución y tamaño de pantalla del dispositivo que use el usuario. Esto conlleva el uso de nociones css conocidos como "**viewport**" que, aunque opcional, supone una mejora sustancial a favor del carácter multiplataforma del que se quiere dotar la aplicación. El comentario sobre esta mejora continua en 4.5 Tareas de optimización.

Para acabar, la opción de **app.config.suppress_callback_exceptions** es activada para no generar mensajes de error en la aplicación cuando el usuario realice una acción no sea una de las definidas entre las posibles y discretamente explicitadas en la programación (por ejemplo, pulsar en otro punto de la página cuando un menú desplegado se encuentra abierto).

Llegados a este punto, da comienzo la definición de la plantilla como tal. Este bloque se encarga de la apariencia de la herramienta. En la plantilla, se define la app como un **árbol de componentes**, donde se suceden o se anidan según estos se sucedan, o bien se contengan unos a los otros (y hereden en este caso estilos). Estos componentes son de tipo **html.Div()** o **dbc.Row / dbc.Column**. El primero de los casos tienen los mismos atributos que una **división de HTML** (*margin, padding, height, width, display, …*) a los que se suman los estilos css dentro de un argumento llamado **style** con formato diccionario. El segundo utiliza estilos estandarizados que vienen detallados con la importación del tema de Bootstrap y facilita la traducción idea – plantilla. Por convenio, Bootstrap define que una plantilla está ocupada por **12** columnas y si se quiere incluir un elemento que ocupe la mitad de la pantalla, es tan fácil como asociar el valor **"one-half.column"** al atributo **classname** del elemento. Con la figura a continuación, se pretende ilustrar el comentario anterior.

```
html.Div(             # Body block Outlier Detection
    children = [
        html.Div(           # First division: description, title and user's menu
            children=[
                html.Div(           # Left sub-section: Description and title
                    id="intro-text-div-analysis",
                    children = [
                        html.Br(),
                        dcc.Markdown(intro_outlier_tab_esp),
                        html.Br(),
                        dcc.Markdown(intro_outlier_tab_eng),
                        html.Br(),
                        html.H4("Malaga Transformer Clustering", style={'margin-left':'6%'}),   # 30px
                    ],
                    style = {
                        'text-align': 'justify',
                        'display': 'inline-block',
                        'width': '55%',
                        'float': 'left',
                        'margin-left' : '2%',   #10px
                        'padding': '1%'    #10px
                    }
                ),
```

Figura 40: Ejemplo de árbol de divisiones en la plantilla

Como se puede ver, cada una de las divisiones alberga los elementos de su interior en un argumento **children** en formato lista, el cual por convención siempre se encuentra en primer lugar (razón por la que a menudo se omite). A estos se suman un identificador **id**, que se usa para unívocamente para referir los **callbacks** del final, que declaran la interactividad que sufre ese elemento y define cuales elementos son de entrada (**input**), de salida (**output**) o de estado (**state**) – la forma en la que funcionan es comentada en la 4.4 – y por último, el diccionario **style** donde se suceden las propiedades que se quieren definir. En este caso, se define una división donde se incluyen titulo (html.H4) y descripciones (dcc.Markdown), y cuyo cuerpo se justica centradamente en la página, y cuya propiedad **display** se fija en **"inline-block"**. Esta propiedad HTML es necesaria para que otro bloque o bloques puedan ser concatenados en la misma fila. Si en su lugar se diese un valor de tipo **"block"**, el siguiente bloque iría puesto en la siguiente fila de este, sin poder ir a su derecha.

Para ilustrar alguno de los elementos de la estructura de la aplicación, se adjuntan a continuación una serie de capturas describiendo la declaración de algunos elementos principales. Llama la atención la facilidad de uso y programación de alto nivel que, como se viene recordando, es una de las máximas virtudes de la plataforma Dash.

```
html.Img(
    # src="https://drive.google.com/uc?export=view&id=1m17KAS2GEoGND8UsyvYH8BoqpVQ0hIBg",
    src='/assets/smart_malaga_logo.png',
    height='30%',  #55px
    width='6%',   #85px
    style={
        'display': 'inline-block',
        'margin-left': '53%',
    }
),
```

```
dcc.Dropdown(
    id = "variable-dropdown",    # Used to identify the dcc in callbacks
    options=[
        {"label": "Tension", "value": "Tension"},
        {"label": "Intensidad", "value": "Intensidad"},
        {"label": "Potencia activa", "value": "Potencia activa"},
        {"label": "Potencia reactiva capacitiva", "value": "Potencia reactiva cap"},
        {"label": "Temperatura ambiente", "value": "Temperatura ambiente"},
    ],
    value="Intensidad",      # The initial selected value
    placeholder = "Selecciona una variable",
),
```

```
html.Div(
    children=[
        html.Button(
            " Ejecutar ",
            id="btn-updt-map",
            title="Lanzar la presentación de resultados",
            className="button-primary"
        )
    ],
),
```

```
dbc.Row(
    [
        dbc.Col(           # Left sub-section: histogram on variable
            dcc.Graph(id="histogram"),
            width = 6
        ),

        dbc.Col(           # Right sub-section: dropdown CT + line figure
            dcc.Graph(id="line"),
            width = 6
        )
    ],
    no_gutters = True
)
```

Figura 41: Constructores Dash de una imagen (1), de un Menú Desplegable (2), de un Botón (3) y de dos gráficos alineados horizontalmente (4)

El resultado de la herramienta visualizadora se muestra en el Anexo 3, la cual se aconseja consultar para continuar con la lectura.

En el apartado a continuación, el informe focaliza su explicación alrededor de la técnica y evaluación de métodos para diseñar el módulo de análisis para la reducción dimensional y sucesiva detección de valores anómalos.

4.3 ANÁLISIS AGREGATIVO Y DETECCIÓN DE VALORES ANÓMALOS

Este apartado se limita a explicar el progreso y producto fruto de la capacidad de análisis que se pretende implementar en la herramienta. Para no hacer de la aplicación un mero escaparate de datos, aunque interactiva, se pensó que podría ganar mucho valor si extraía información no evidente implícita del conjunto de datos de partida.

En un momento inicial, se valoró la posibilidad de utilizar un **algoritmo de clasificación según aprendizaje supervisado**. La **salida** del modelo correspondería al **Centro de Transformación** de donde emerge un punto de medición en base a los valores de sus variables, y se traduciría por una variable numérica discreta del 1 al 20. El propósito sería valorar la cercanía existente entre distintos CT, valorando la facilidad que tenga el modelo en localizar las nubes de puntos de CT totalmente distinguidas, o la falta de capacidad que tengan cuando las nubes de puntos entre CT solapen en gran medida.

Sin embargo, durante el prototipado del modelo, se observó rápidamente que los agrupamientos (*clusters*) de puntos de cada CT son poco distinguibles cuando son entremezclados, y no se trata de un problema fácilmente separable sin caer en el sobreentrenamiento (es decir, no es bajo ningún caso un problema linealmente separable). Esto en particular es consecuencia en gran medida debido a la oscilación generalizada de todas las observaciones de tensión alrededor del valor de 235V debido a la regulación de la red eléctrica. Además, el atributo del CT es de fácil extracción y no parece ser un valor de dudosa calidad y la compañía no tiene un reto particular frente a ese atributo.

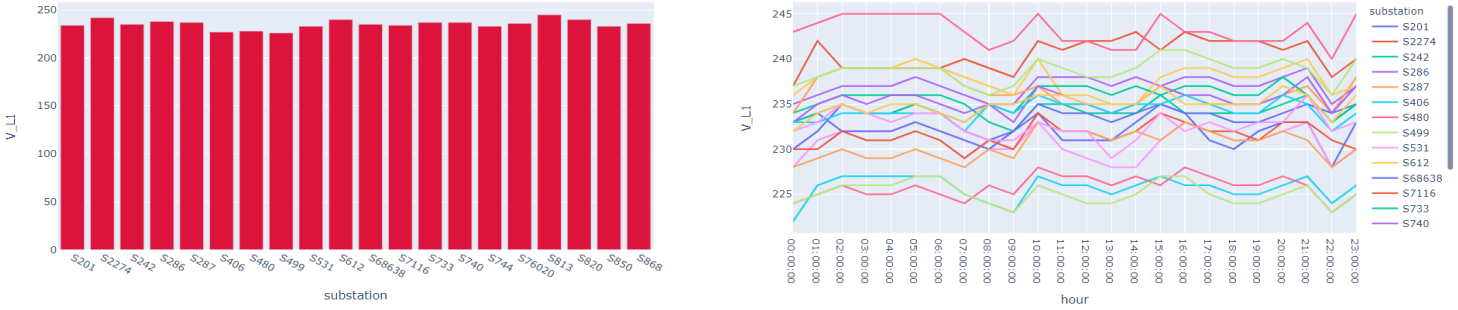


Figura 42: Valores de tensión el 26/06/19 desde la app

En contrapartida, la idea inicial fue modificada para acabar mutando en un **algoritmo no supervisado de detección de valores anómalos**. Partiendo de la selección del CT, es muy interesante para las partes interesadas el conocer lo alejado que se encuentre un punto de operación lejos del resto de los más frecuentes. En términos científicos, esto se corresponde con el problema de **detección de valores anómalos** (*outlier detection*) y sin etiquetar los datos de partida (no se dispone de valor que discrimine entre medida válida y no válida), a partir de la distribución y topología de la nube de puntos. El propósito es dejar al modelo predecir sin restricciones cuando un punto del conjunto de entrenamiento tiene poca probabilidad de pertenecer a la familia del resto de puntos. Estas tienen una diferencia tal con respecto al resto de puntos que podría considerarse como un punto no coherente, perteneciente a otro proceso o mecanismo que lo genera. Estos valores atípicos pueden indicar una variabilidad espontanea del fenómeno, error experimental sobre la medida o un anuncio de error sistemático (como podría ser una carga particularmente inusual con respecto a la esperada, o una próxima rotura del centro de transformación). En definitiva, un valor anómalo se corresponde con una observación que **diverge de un patrón general común a una muestra**.

Son muchas las técnicas existentes para detectar valores atípicos y lo idónea de su aplicación viene muchas veces determinada por la aplicación para lo que se use (de ahí la importancia una vez más de conocer los datos antes de abalanzarse sobre la búsqueda de la solución). De las más básicas (como el **Z-score**, que calcula cuántas desviaciones estándar está un punto separado de la media muestral y que exige transformaciones de los datos para normalizarla a una Gaussiana), estadísticas locales (como el **DBSCAN**, basado en el cálculo de densidad

local de puntos alrededor, denominados **vecinos**, que se encuentren dentro de un radio paramétrico de radio ε) o en base a funciones de distancia (como son los métodos de Árboles de Aislamiento, ***Isolation Forest***, que se basan en la hipótesis que los valores anómalos son pocos y están alejados del resto del conjunto), cada una incluye pros y contras y exige el análisis sobre el resultado para conocer la solución óptima.

Pero a este reto de análisis inteligente se sumaba la principal limitación propia de la aplicación que ocupa a este proyecto: el resultado tiene tanta importancia como fuerza tenga la presentación visual de sus conclusiones. Es aquí donde entra pues la **propuesta de la reducción dimensional**. Este problema se da siempre que sea exigida una reducción de los atributos de entrada para responder a una exigencia de disminución en el conjunto de datos después de ser transformado. Como para el caso anterior, las técnicas son múltiples, pero en este proyecto se realizan ambos análisis para conocer cuál de ellos tiene más sentido.

En primer lugar, se procede a reducir el problema por medio de la selección de atributos. En base a la alta colinealidad existente entre atributos por propiedad sistemática (como ocurre entre variables análogas de distintas líneas), a la pobre calidad de ciertos atributos y a la urgencia por limitar la pérdida de información que se da por la reducción dimensional, se limitan los atributos de entrada a **Tensión, intensidad, Potencia activa, Potencia Reactiva Capacitiva y Potencia Reactiva Inductiva** de la línea 1, sumando un total de cinco variables de entrada para las más de 160.000 observaciones totales.

Una vez conocidos los atributos de inicio, es necesario establecer un sistema objetivo y cuantificable que valore el rendimiento de cada técnica, contando así con un proceso metodológico y objetivo que califique la calidad de cada una de ellas. Como es común realizar en el mundo de la Inteligencia Artificial sobre todo en problemas de análisis supervisado, se divide el conjunto de datos en **subgrupos** de **Entrenamiento** (*Training*), **Validación** (*Validation*) y **Prueba** (*Test*). En la industria, es común valorar el tamaño del conjunto de inicio para decidir el tamaño de cada sección. En proyectos previos al siglo XXI, la escasez de datos tanto en volumen como en diversidad obligaba muchas veces a reservar un porcentaje más grande para los dos últimos grupos. El compromiso se da pues cuantos

más datos de entrenamiento se disponen, mejor es posible ajustar el comportamiento del modelo para que mejore sobre la función objetivo que se decide optimizar. Por el otro lado, cuanto menor sea el tamaño de los últimos dos grupos, menos representativos serán los resultados que estos arrojan para evaluar la calidad del algoritmo. Sin embargo, conforme la disponibilidad en observaciones crece, menos estrecho se vuelve el margen de acción y, en consecuencia de superar los 100.000 puntos de entrenamiento, se decide repartir según la combinación **72% – 18% – 10%.** Para reducir el sobreentrenamiento, optamos por utilizar **validación cruzada**: en lugar de fijar la extracción, recortamos el conjunto de entrenamiento homogéneamente, e iteramos en cada porción para que la media de los resultados en cada configuración ofrezca una visión más fiel y neutral del comportamiento del modelo.

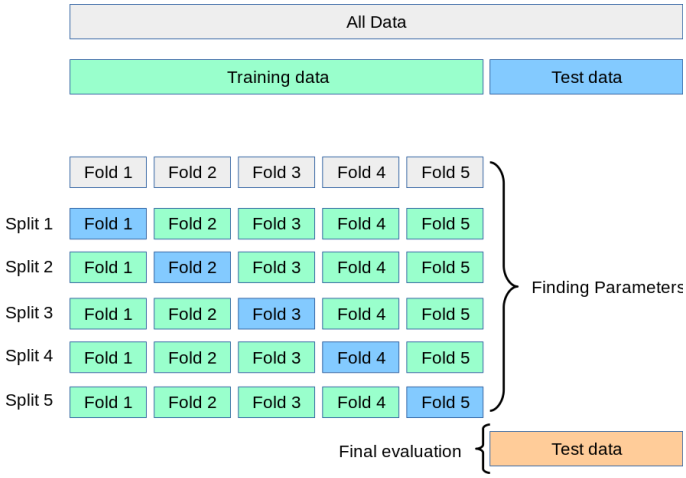


Figura 43: Reparto Train - Validation – Test

Dentro de la operación de división, es importante aleatorizar las observaciones. Si no se realiza, se caería en una diferencia fundamental entre los datos de entrenamiento y de prueba. Considerando que el orden existen ahora mismo es según el orden cronológico, esto supondría evaluar el rendimiento del modelo que ha sido entrenados en fechas anteriores al segundo cuatrimestre de 2020, con un conjunto de fechas del final de la base de datos, que, además, da la casualidad coinciden con el inicio de la pandemia COVID. Esto introduciría un sesgo sistemático inevitable en el segundo y tercer conjunto indeseado que en los proyectos de *Machine Learning* supone una deformación en las distribuciones de

entrenamiento y prueba. En resumidas cuentas, se estaría entrenando en un entorno distinto a aquel donde se quiere destacar.

A esta mezcla se incluye un valor semilla que permite una ejecución semi-aleatoria y durante el desarrollo sea posible un seguimiento del progreso **comparable** (si todo fuera totalmente aleatorio, una mejora de programación podría llevar por mala suerte a un peor resultado, sobre todo en la fase de **calibración de hiperparámetros**, o *hyperparameter tuning*).

El segundo punto sobre el que es necesario ahondar, es sobre la necesidad imperativa de normalizar los valores antes de entrenar los modelos. La explicación consiste en la intuición que se entienden por funciones como distancia, cercanía o densidad y es que es sencillo entender porque la escala de las variables juega un papel importante. Intuitivamente, puntos que en un diagrama 2-D **"cercanos" visualmente** igual están distanciados de unas micras de unidad en un eje y de cientos de unidades en el segundo eje. Un buen estimador para valorar la necesidad de esta transformación consiste en representar las diferencias de escala entre atributos.

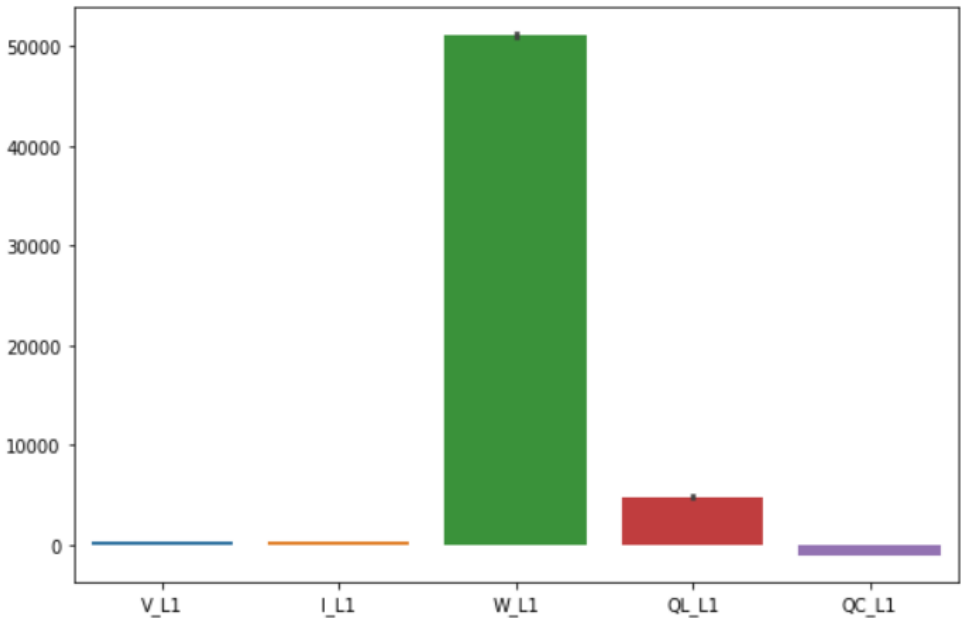


Figura 44: Varianza de atributos de entrada

Si existen variables categóricas o en escala logarítmica, las opciones se reducen y un tratamiento particular es necesario. Sin embargo, sabiendo que todas las variables de entrada son numéricas continuas y que visiblemente se evidencia la necesidad de corrección, se normalizan todos los ejes, restando a todas las componentes la media de los valores y dividiendo por el rango. Esta es la operación realizada por el objeto creado de Scikit-learn que se utiliza en el programa, escala entrenada sobre **X_tr** y aplicado sobre el resto.

```
# Standardize Data
sc = StandardScaler()
X_tr_std = sc.fit_transform(X_tr)
X_valid_std = sc.transform(X_valid)
X_test_std = sc.transform(X_test)
```

Figura 45: Normalización de datos de entrada

Para finalizar el análisis, se procede a evaluar la asociación entre atributos por medio de una matriz de correlación.

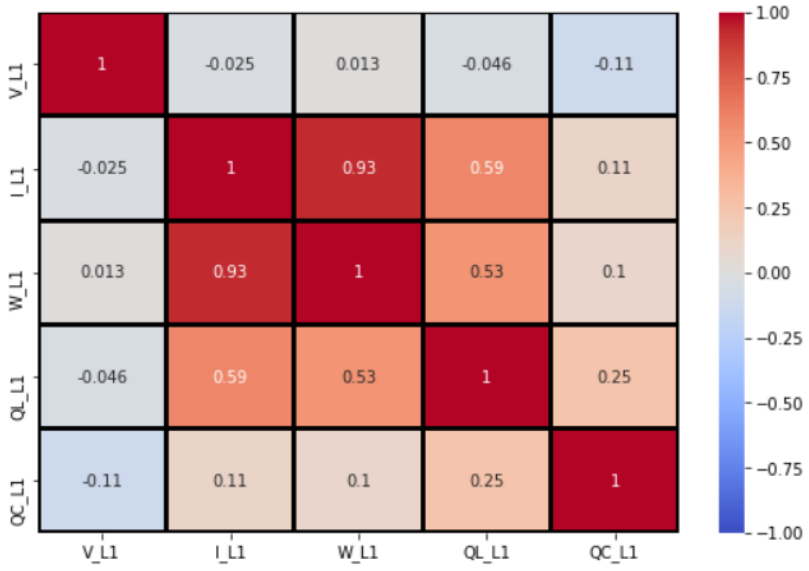


Figura 46: Mapa de Calor de Correlación

Se infiere de esta figura que la correlación destacable entre potencia activa e intensidad no se reproduce en más variables, por lo que puede el estudio puede ser llevado a cabo.

Con los datos procesados, se procede al análisis de modelos que se pasan a molestar. Estas versiones tendrán como base dos modelos canónicos de inicio: **la técnica PCA** (*Principal Component Analysis*) y **la técnica de AE** (*Autoencoder*).

El PCA es uno de los algoritmos de reducción de dimensionalidad por su simplicidad y eficacia. PCA itera encontrando los ejes ortogonales que maximicen la varianza a lo largo de ellos. Trans estandarizar los datos, el modelo obtiene los **valores y vectores propios de la matriz de covarianza** o bien la **Descomposición de Valores Singulares** (SVD) para mejorar la eficiencia computacional. Una vez estos ejes determinados (que son dos o tres para reducir la dimensión para la visualización), se proyectan los datos originales en el nuevo espacio vectorial para conseguir una nueva base de datos de **menor número de atributos,** pero de **menor error de deformación**.

Por otro lado, el AE es un **tipo concreto de Red Neuronal** que se entrena con la intención de reproducir de manera óptima los datos de entrada en salida, pasando por un **espacio latente** intermedio con menor información. Compuesto de un Codificador y de un Descodificador contiguo, se realiza una reducción de la redundancia en un primer tiempo para, a partir de este dato más denso, trate la segunda mitad de replicar de la mejor manera el objeto de entrada.

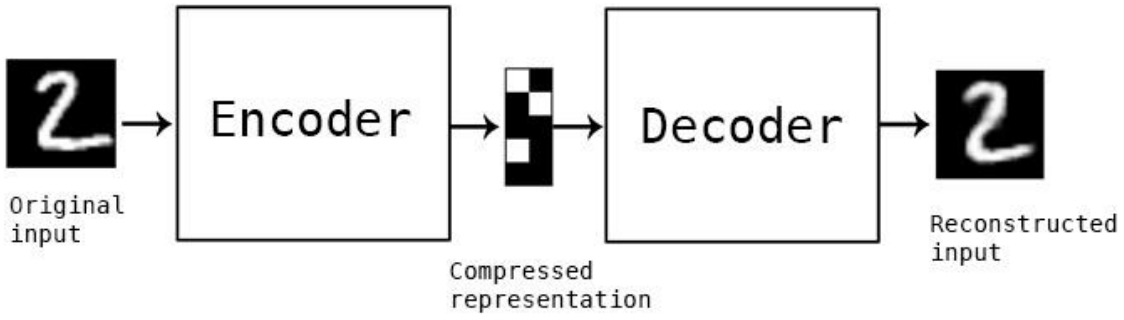


Figura 47: Esquema ilustrando el principio de funcionamiento del AE [32]

En este estudio se comparan los resultados en los conjuntos de Validación y Prueba en estos modelos y sus variaciones.

4.3.1 ESTUDIO DE MODELOS PARA LA REDUCCIÓN DIMENSIONAL

4.3.1.1 Auto – Encoder simple

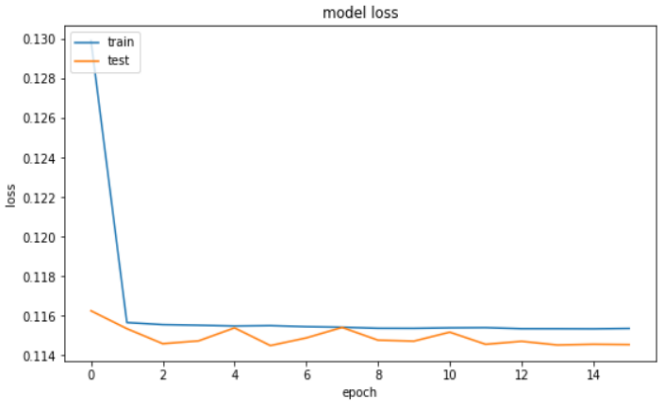
Este autocodificador consta de tres capas, con una intermedia de tres neuronas donde se codifica el espacio latente. Para implementarlo, se usa la programación de alto nivel de **Keras** como se muestra a continuación.

```
encoder = keras.models.Sequential([
    keras.layers.Dense(LATENT_SHAPE, input_shape=[INPUT_SHAPE]),
])

decoder = keras.models.Sequential([
    keras.layers.Dense(INPUT_SHAPE, input_shape=[LATENT_SHAPE]),
])

autoencoder = keras.models.Sequential([encoder, decoder])
autoencoder.compile(loss='mse', optimizer = keras.optimizers.SGD(learning_rate=0.1), metrics=['accuracy'])

history = autoencoder.fit(X_tr_std,X_tr_std, epochs=30, validation_data=(X_valid_std, X_valid_std),
                          callbacks=[keras.callbacks.EarlyStopping(patience=10)])

codings = encoder.predict(X_tr_std)
ae_codings_test = encoder.predict(X_test_std)
```

Figura 48: Entrenamiento y Codificado de datos de entrada

Primero se declara la parte codificadora, codificando la forma de entrada en 5 variables y la de salida en 3 dimensiones. El modelo del descodificador es simétrico y el modelo completo surge de su concatenación (usando el método **Sequential**). Seguidamente, se programa el compilador que declara la variable de "error" como Error Cuadrático Medio ("*mse*" de Mean Square Error) y utilizará para iterar un **Optimizador Estocástico de Descenso del Gradiente** (*SGD Optimizer*) de ratio de aprendizaje fijo de 0.1. En el entrenamiento, se activa el método "Early Stopping" para que el modelo aborte el entrenamiento cuando el error de Validación deje de mejorar (entrando en la zona de sobreentrenamiento).

Usando la variable **history**, es posible evaluar la evolución que la métrica de error y precisión tiene con respecto al número de iteraciones aleatorias (o *epochs*).




```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential (Sequential)      (None, 3)                 18
_________________________________________________________________
sequential_1 (Sequential)    (None, 5)                 20
=================================================================
Total params: 38
Trainable params: 38
Non-trainable params: 0
_________________________________________________________________
None
```

Figura 49: Evolución del Error en X_Tr y X_valid (1) y Modelo del Auto-Encoder (2)

Para tener una intuición sobre la proyección que se realiza de los primeros 20.000 puntos codificados de X_tr_std por el módulo codificador, se muestra a continuación los valores en los dos primeros nodos.

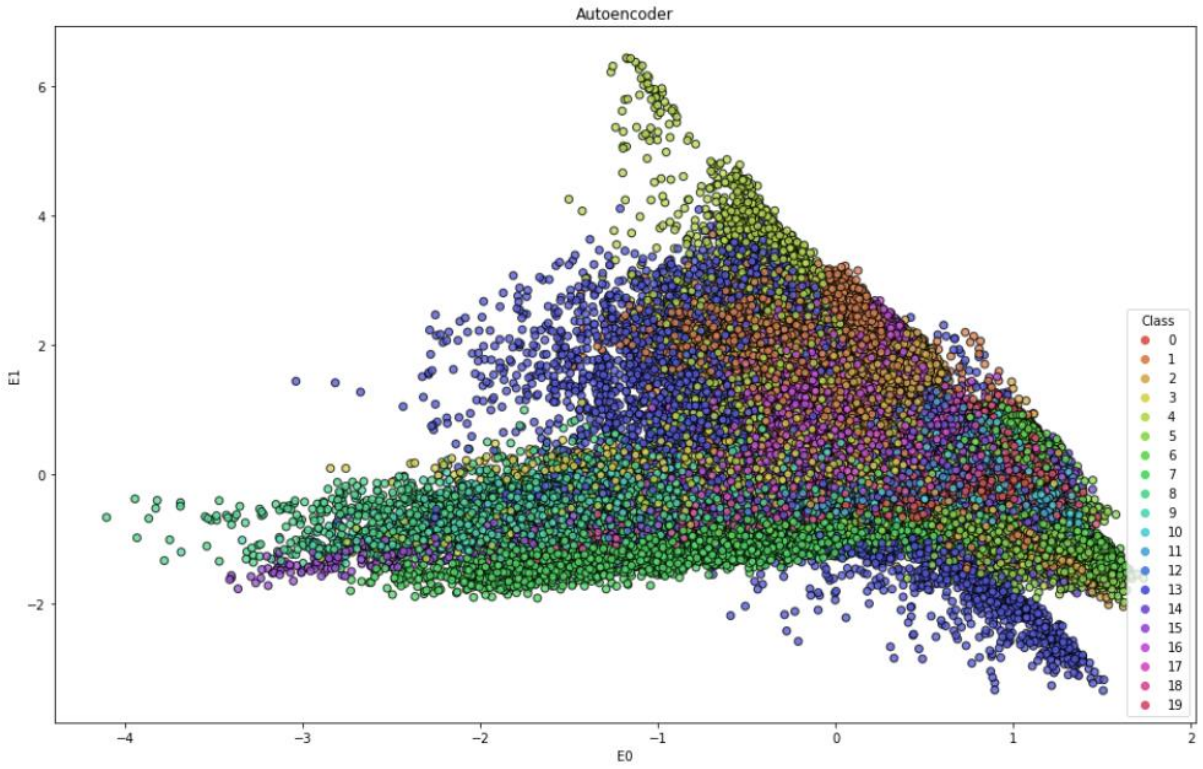


Figura 50: Representación de los dos primeros códigos del espacio latente junto con el CT como leyenda

4.3.1.2 Principal Component Analysis

Su programación es directa usando la librería de Scikit-learn. Se muestran las reproducciones en las primeras tres dimensiones que las proyecciones del PCA generan en un nuevo conjunto de datos (X_new), respecto a las tres primeras dimensiones originales de X_tr_std.

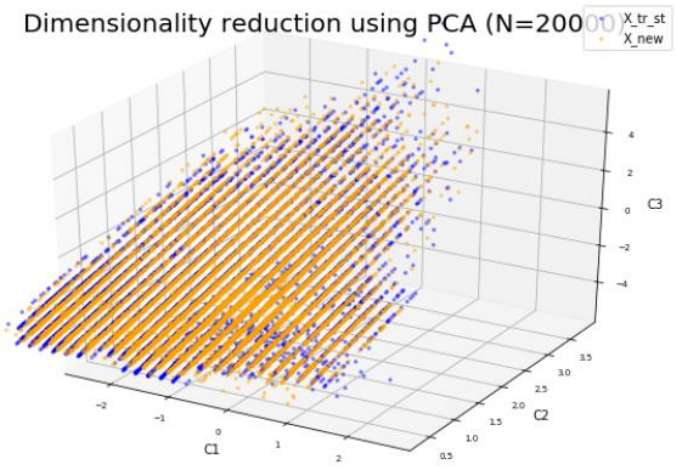


Figura 51: Transformación inversa de las proyecciones frente a originales

Es interesante ver como que, a pesar de la evidente diferencia de las primeras 20.000 observaciones, se discierne fácilmente el perfil característico de la base de datos original y cómo la generación de la nueva base de datos se superpone sobre la misma de manera visible. Comparando los resultados de ambas técnicas, se aprecian que la diferente lógica lleva a codificaciones sustancialmente distintas a la espera de resultados con respecto al objetivo.

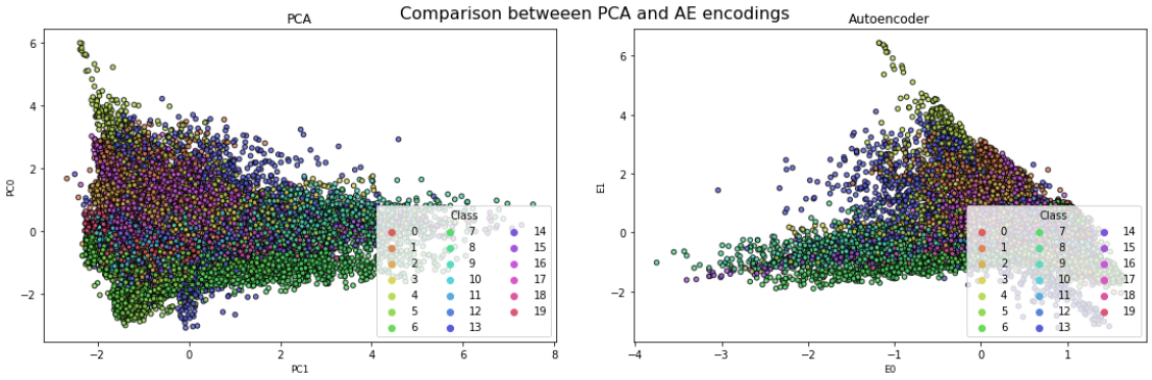


Figura 52: Comparativa entre codificaciones PCA y AE en las primeras componentes

De la gráfica anterior se aprecia que, aunque la diferencia del método lleve a conclusiones distintas, arroja una característica interesante con respecto a las distribuciones de observaciones por CT: **sin ser linealmente separables, sí están localizados en el espacio** (no se distribuyen a lo largo de todo el espacio) cuando, por lo general, al proyectar este fenómeno además se ve reducido (al acumularse puntos de diferente valor en la tercera dimensión en el mismo punto sobre la proyección ortogonal en 2D). Son necesarios más resultados para comparar la eficacia de los métodos.

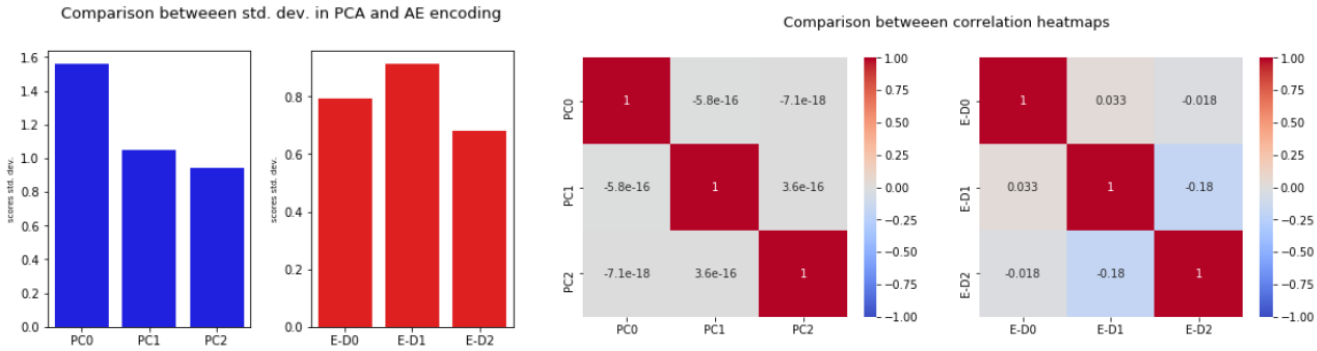


Figura 53: Comparativa de varianza en PCA y AE Simple (1) y Mapa de Correlación entre componentes (2)

Como era de esperar, el ordenamiento de la varianza para los componentes principales del PCA en orden decreciente muestra el **principio de funcionamiento del algoritmo**, así como la falta completa de correlación entre componentes (sinónimo de ortogonalidad de ejes). En el AE, sin embargo, las diferencias no son tales al no estar sometido a dichas restricciones.

4.3.1.3 Versiones de Auto-Encoder

El primero de los modelos modificados es un **Autocodificador Multicapa**, manteniendo la estructura pero incluyendo una capa intermedia que otorgue mayor complejidad potencial al modelo (aunque esto no tenga un impacto positivo sobre los resultados *per se*). Esta variación, como suele ocurrir en Redes Neuronales, dota a la red de más flexibilidad para aprender características más informativas y complejas, abstracciones de las más simples.

El codificador tiene una estructura tipo **5 (entrada) – 4 – 3**. El compilador es equivalente y el descodificador es de nuevo simétrico para generar en salida la misma dimensión.

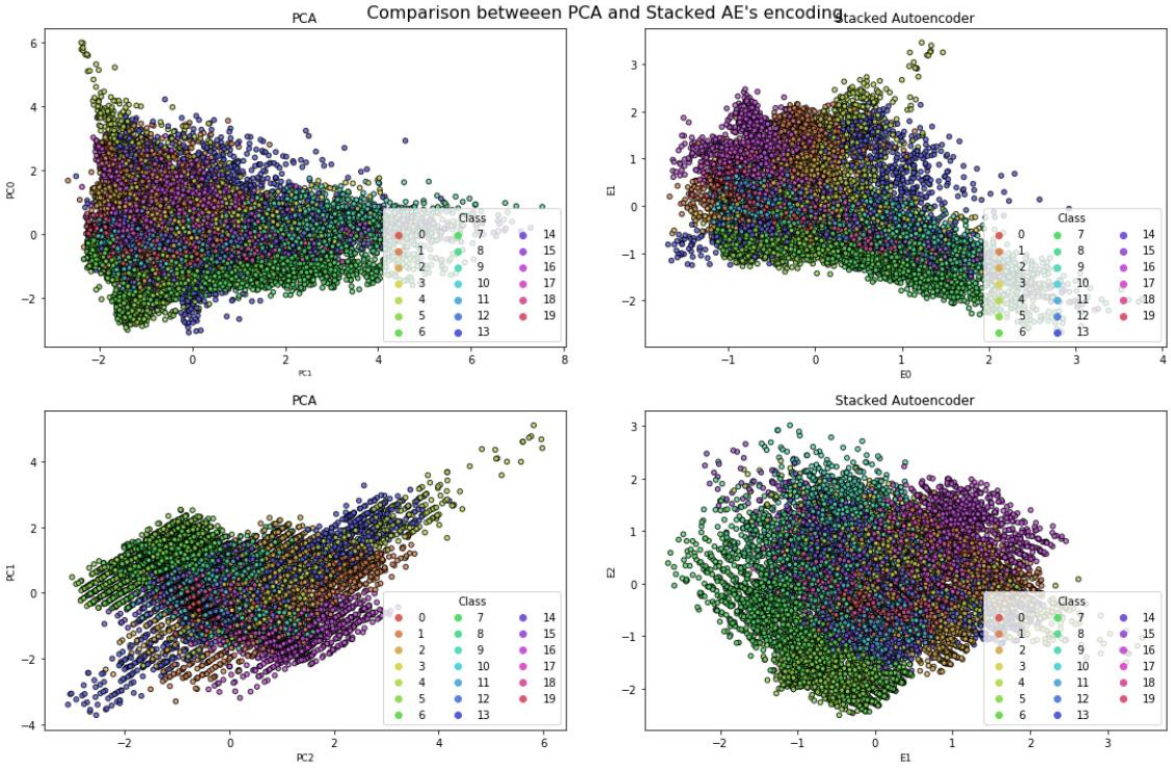


Figura 54: Representación de las codificaciones en PCA y en Stacked_AE

El segundo modelo variacional sobre el AE simple es un AE profundo No Lineal, donde fácilmente es posible implementar una estructura en cuyos nodos internos se dé una **función de activación no lineal**. Se selecciona para ello la función tipo **"selu".**

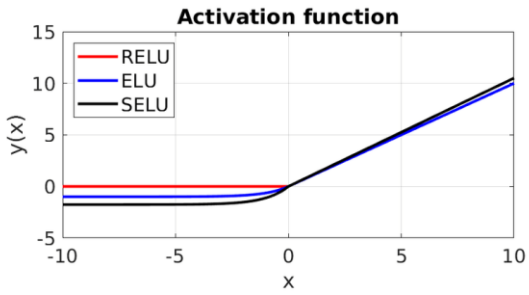


Figura 55: Diferencias en funciones de activación lineal y no lineal

A esto, además se incluye al optimizador **una constante de decaimiento** que varíe el **ratio de aprendizaje** decrezca con las iteraciones y se reduzca conforme se acerca al mínimo de la función objetivo.

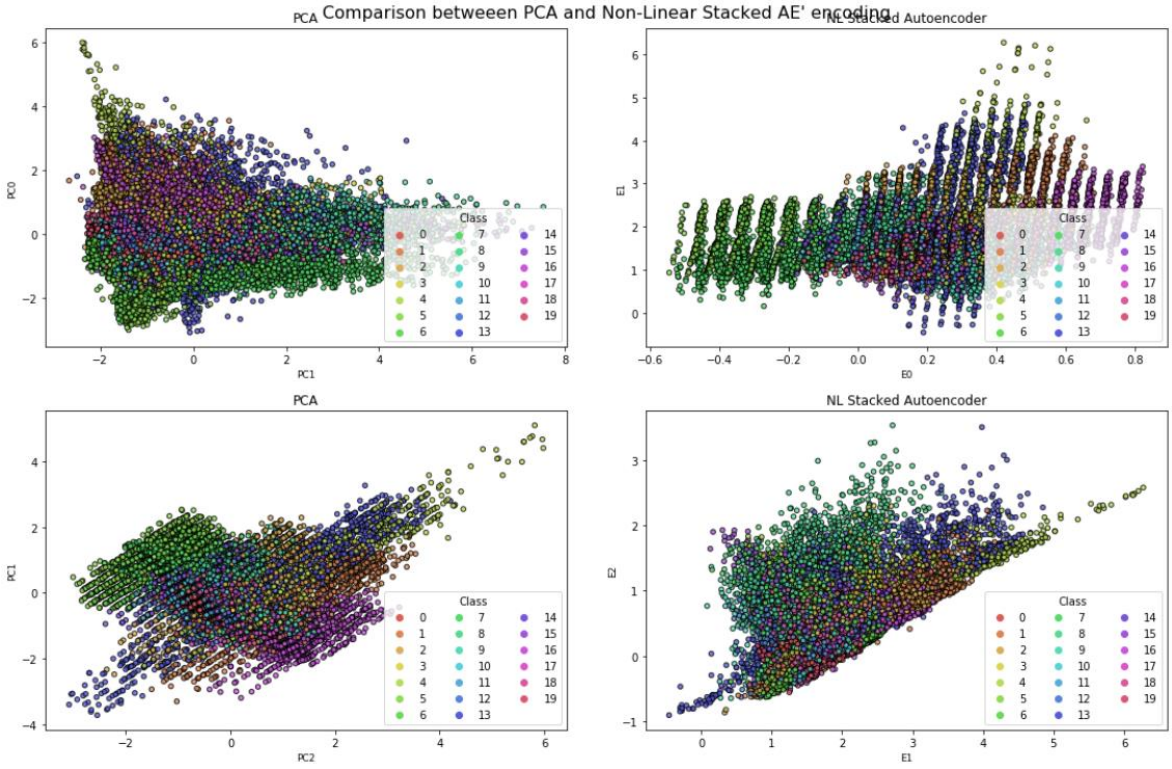


Figura 56: Comparativa de codificaciones entre PCA y NL_St_AE

Llama la atención ver esta vez el comportamiento de los valores discontinuos de tensión mejor reflejados en el modelo NL_St_AE entre el primer y segundo nodo pero más entre PC1 y PC2. Las diferencias se acrecientan, el universo de elección para el ajuste de hiperparámetros crece y con ello, la complejidad del modelo, que prueba nuevos métodos para la reducción dimensional.

4.3.1.4 Enfoque estadístico: modelo t-SNE

Para acabar, se incluye en esta comparativa un modelo esencialmente distinto como es el **t-SNE**. *t-Distributed Stochastic Neighbor Embedding* es una técnica utilizada para la reducción dimensional utilizada particularmente para la visualización de bases de datos de alta dimensionalidad. Al contrario que el PCA, esta técnica no se basa en una formulación matemática sino probabilística. En el artículo original de 2008 [33] se define:

"t-Distributed Stochastic Neighbor Embedding (t-SNE) minimiza la divergencia entre dos distribuciones: una distribución que mide las similitudes dos a dos de los puntos de entrada,

y de una distribución que mide las similitudes dos a dos de los correspondientes puntos de baja dimensionalidad codificados" (traducción propia).

Esencialmente, esto significa que el propósito del modelo es modelizar de la manera posible el conjunto de entrada, con el mapeado tal que la correspondencia entre ambas distribuciones se vean desvirtuada mínimamente. La manera con la que se computa dicha técnica es pesada y por lo tanto cuenta con serias limitaciones de ejecución. Una técnica comúnmente para sobrellevar este problema es, en el caso de datos de muy alta dimensional, realizar una reducción inicial con otro método (PCA por ejemplo) para a partir de un conjunto más manejable (de hasta 20 variables aproximadamente), utilizar el t-SNE

Se utiliza en este trabajo la implementación en Scikit-learn para lo que queda de documento.

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=3, verbose=1, perplexity=40, n_iter=300)
tsne_results = tsne.fit_transform(X_tr_std)
tsne_results_test = tsne.fit_transform(X_test_std)
```

Figura 57: Implementación del t-SNE con Scikit-learn

Es interesante destacar que, al contrario que el resto de métodos, t-SNE no cuenta con un método **transform()** común en las implementaciones de Scikit-learn. Como método no supervisado que es, realmente no trabaja con un set de Entrenamiento y otro de Validación, sino que cada transformación de un conjunto de datos para por entrenarlo sobre ello, pero no es trasladable a otros pues el método es **dependiente del soporte**. No es posible aprender una transformación y aplicarlo sobre otros datos ya que t-SNE no aprende un mapeado estático de unas dimensiones de entrada a unas de salida, sino más bien lanza un proceso iterativo sobre un subespacio para encontrar un equilibrio entre ambas distribuciones que minimice la distancia **sobre un conjunto dado**.

Tras un episodio de entrenamiento particularmente largo (para 20.000 observaciones en entrada, tardando más de 30 minutos con únicamente las cinco variables de entrada), se muestra a continuación el diagrama de dispersión de las dos primeras componentes.

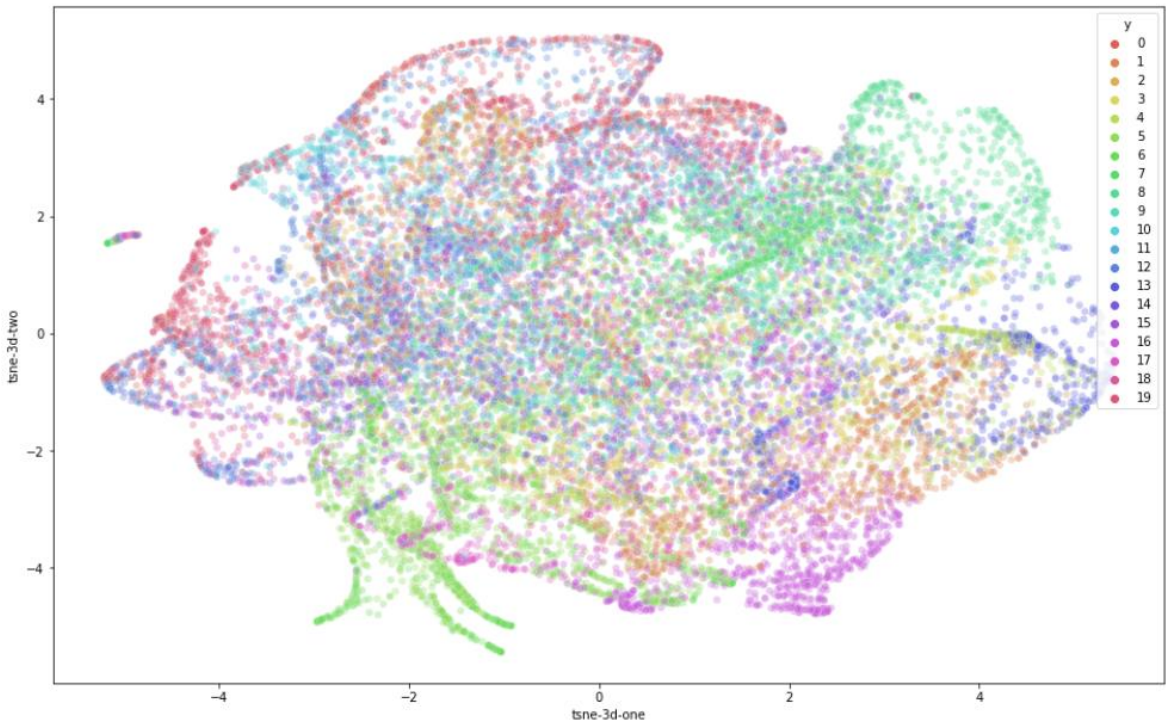


Figura 58: Codificación de t-SNE (N = 20.000)

La comparativa de resultados y defensa sobre la elección del modelo final se desarrolla en la Sección 5. donde cuenta con un apartado reservado a este propósito. Se recomienda encarecidamente al lector que revise dicha sección para interesarse por los resultados finales sobre los conjuntos de Validación (parte del entrenamiento) y de Prueba (valores nunca vistos) y la justificación de la elección en base a rendimiento y eficacia. En esta sección, queda demostrada la capacidad que tiene el Autocodificador para realizar otras tereas que la eliminación de ruido (aplicación común) como es **ejecutar reducción dimensional** como un PCA, desvelando nuevas técnicas para codificar un vector de entrada sobre un espacio menor que deben demostrar su idoneidad según el tipo de problema.

Para el seguimiento del informe, se informa que la elección final de modelo para el módulo analítico pasa por la técnica de PCA sobre las variables de Tensión, Intensidad y potencias de la línea 1 que son proyectados sobre un espacio de tres ejes ortogonales. En la sección siguiente se aborda la parte final del análisis, que consiste en la detección de valores anómalos sobre la proyección 3D fruto del PCA para un Centro de Transformación dado.

4.3.2 ESTUDIOS DE MODELOS PARA LA DETECCIÓN DE VALORES ANÓMALOS

Antes de dar comienzo a la comparativa entre modelos, una justificación sobre el uso del PCA debe ser ejecutada para garantizar la coherencia de los resultados de esta segunda mitad. Como es lógico, no tendría sentido garantizar que una observación se considera atípica a partir de la codificación si no se sabe con certeza que el traspaso de información de cinco a tres variables se realiza **manteniendo la mayor parte de la distribución**. Para ello, se analiza la cantidad de varianza explicada en los datos de partida por las primeras componentes del PCA entrenado sobre la base de datos estandarizada.

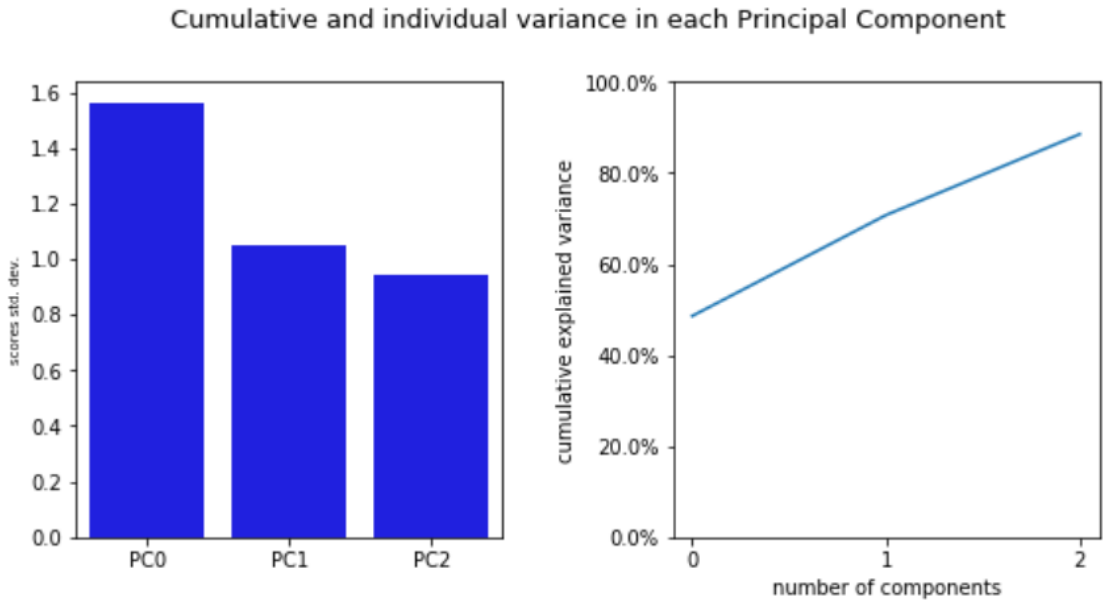


Figura 59: Varianza explicada por PC0, PC1 y PC2

Se observa que dentro de las tres primeras componentes se halla una **varianza explicada acumulada de más del 88%** (0.486, 0.221 y 0.177 respectivamente). La parte restante es el precio a pagar por la proyección y, considerando que suma hasta algo más del 10 %, se considera el estudio viable prosigue con la detección según esta codificación. A pesar de esto, se tendrá en mente esta noción durante el ajuste de hiperparámetros con el fin de ser moderadamente conservador en la identificación de casos límites y así reducir los casos de falsos positivos fruto de la perdida de información en las dos componentes excluidas.

En este apartado del análisis se pretende reconocer los valores atípicos identificados dentro de una ventana temporal para el Centro de Transformación que escoja el usuario. Ya se comentó en el apartado 4.3 pero es necesario recordar la insuficiencia que presentan los métodos de clasificación para este problema ya que no se dispone de una variable o de un estimador que dé información sobre cuales son atípicos, y de hecho siquiera se cuenta con una definición clara sobre qué significa "atípico" en el contexto de este problema (que viene determinado por la **Medida de Disimilitud entre Agrupaciones**). Este problema es uno de aprendizaje no supervisado, donde la separación entre Entrenamiento y Prueba dejan de tener sentido y obliga a buscar otro método para evaluar la calidad de los resultados.

Son muchas las aplicaciones que requieren ser capaces de decidir objetivamente si una observación pertenece a la misma distribución que las existentes en el pasado (como es la detección de fraude, la inspección médica o la ciberseguridad en redes), etiquetándola en consecuencia como valor **típico o normal** (*inlier*), o **atípico o anómalo** (*outlier*). Existe en este momento una distinción central entre dos familias de aplicaciones:

- *Outlier Detection*: Los datos de entrenamiento contienen valores anómalos que se están distanciados del resto. El propósito del análisis es intentar inferir las regiones donde los datos se concentran, ignorando los datos divergentes.
- *Novelty Detection*: La base de entrenamiento no está contaminada de datos anómalos y el interés consiste en detectar si observaciones **nuevas que estén por llegar** son atípicas o no. En este contexto, se denominan valor **novedad** (*novelty*).

Una diferencia fundamental entre ambas familias es que, mientras la primera es no supervisada, la segunda es definida como aprendizaje "semi-supervisado". Sin embargo, ambas parten de hipótesis comunes: los valores anómalos no son capaces de generar agrupaciones densas pues precisamente son anómalas por imperativo.

Por otro lado, los métodos para responder a ambas tareas se agrupan también en dos partes: métodos **aglomerativos** (enfoque "de abajo arriba", donde cada observación comienza como un único grupo y se van uniendo con los más cercanos) y métodos **divisivos** (en el sentido

contrario, las observaciones empiezan como único *cluster* que va sufriendo cismas recursivamente para ser representado generalmente como un **dendograma**).

A continuación, se halla una tabla de las implementaciones disponibles en Scikit-learn [34]

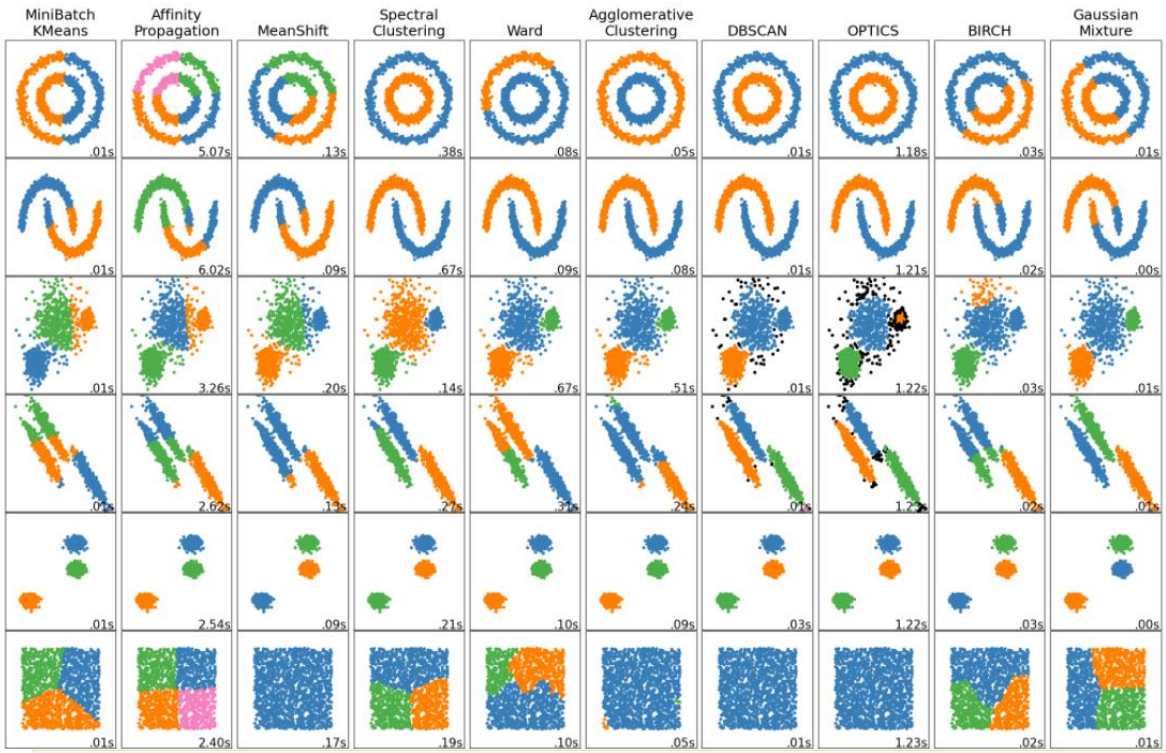


Tabla 8: Tabla recapitulativa de los métodos de Clustering en sklearn

Es apreciable la similitud entre nuestro caso de uso y el descrito en la tercera fila. Entre todos los casos, son **DBSCAN y OPTICS** quienes mejor responden a nuestros requerimientos y demuestran potencial en identificar valores anómalos (que se corresponden con los valores negros). Al estudio se añade por último el algoritmo denominado **Local Outlier Factor**, método estadístico que mide la desviación local de vecinos en una muestra.

Antes de pasar a la comparativa, el autor desea destacar la posibilidad de ampliar las funcionalidades que se van a implementar en la aplicación, ya que aunque se va a enfocar el caso como un problema de *Outlier Detection*, muy pocas modificaciones serían necesarias sobre la estructura actual para ampliarlo en tanto que detector de valores novedad, notificando las apariciones de observaciones nuevas con alta probabilidad de ser anómalos.

4.3.2.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

El algoritmo DBSCAN ve las agrupaciones de datos como áreas de alta densidad separadas de valles de baja densidad. Gracias a la generalidad de este axioma, DBSCAN es capaz de reconocer *clusters* de todas las formas y tamaños (en oposición a *k-means* por ejemplo, el cual está obligado a que las nubes de puntos sean estrictamente convexas).

El componente central de DBSCAN es el concento de **muestras núcleo**, que son las presentes en las áreas de alta densidad. Un *cluster* es así un conjunto de muestras núcleo muy cercanos, más un conjunto de puntos no núcleo que están en la vecindad de una de las anteriores (y que, intuitivamente, se sitúan en el borde del *cluster*). Para definir ambos, se hacen uso de dos parámetros: **min_samples** y **eps**, que albergan juntos lo que entenderá el algoritmo por "denso". Formalmente, se define así una muestra núcleo todo aquel punto que tenga como mínimo "min_samples" vecinos a una distancia menor de "eps". Un agrupamiento se construye así recursivamente, tomando una muestra núcleo de la lista, encontrando sus vecinos y evaluando si ellos lo son a su vez. Mientras que min_samples se encarga principalmente de determinar cómo de tolerante es el modelo frente al ruido, el parámetro eps es crucial para traducir la función distancia implícita en el algoritmo.

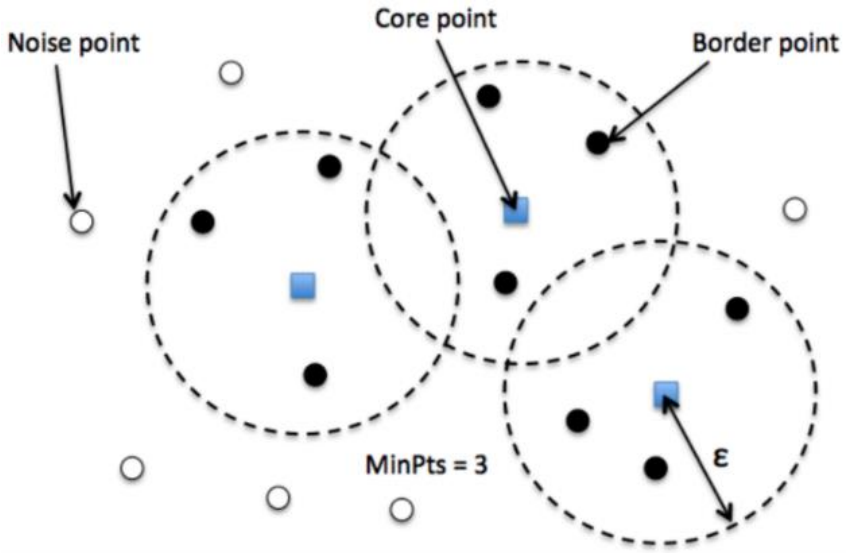


Figura 60: Explicación visual del principio de funcionamiento de DBSCAN [35]

4.3.2.2 OPTICS (Ordering Points To Identify the Clustering Structure)

OPTICS guarda muchas similitudes con el algoritmo DBSCAN, y es considerado como una generalización de este pues el valor de **eps** muta de un valor único a un **rango de valores**. La diferencia fundamental es que OPTICS construye un **Gráfico de Alcanzabilidad** durante el proceso iterativo, el cual asigna una distancia de alcanzabilidad a cada punto así como factor de ordenación en el eje de las abscisas para colocar juntos a puntos adyacentes. Ambos son usados por el modelo para la asignación de agrupamiento o determinación como valor anómalo. A efectos prácticos, esta diferencia juega un papel fundamental pues integra en la función distancia la diferencia en densidad que pueda existir entre distintos grupos de puntos.

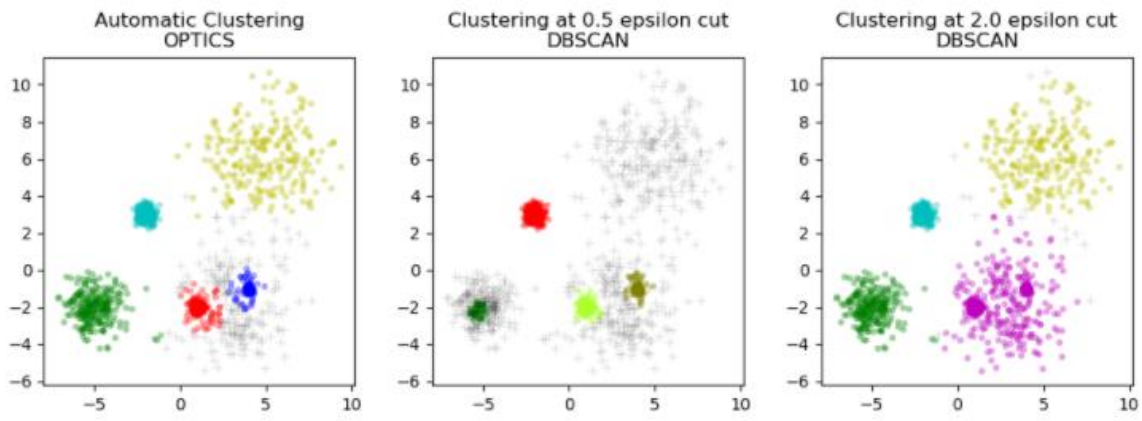


Figura 61: Consecuencia práctica sobre el clustering de DBSCAN y OPTICS para grupos de densidad variable

En un conjunto de datos donde agrupaciones de datos huecas y alargadas conviven con otras densas, DBSCAN encuentra el límite de elección de único valor para eps. Si el valor es demasiado grande, *clusters* distinguidos por su densidad acaban por unirse en uno, pero descenderlo de más provoca no detectar los extendidos y poco densos. OPTICS responde a este problema utilizando un rango de valores amplio.

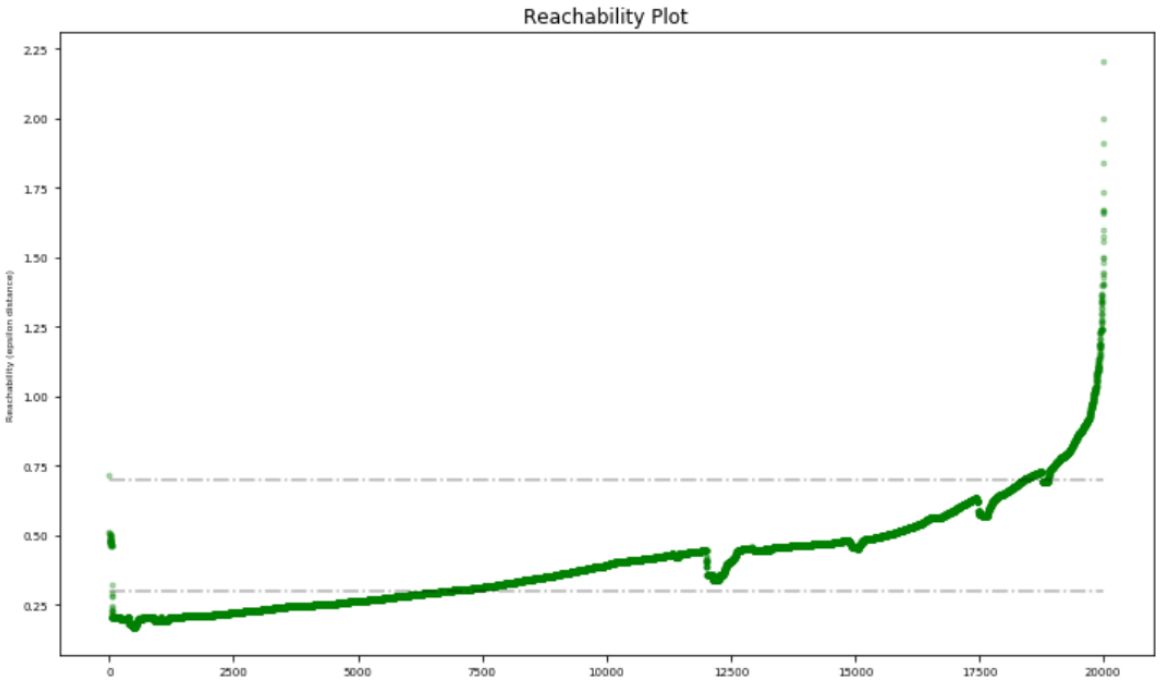


Figura 62: Gráfico de Alcanzabilidad de PCA_codings (N = 20.000)

En la figura anterior se muestra el gráfico que OPTICS genera en la codificación de los datos, con min_samples = 200. A este se suman las líneas con los epsilon escogidos para los modelos DBSCAN que entrarán en la comparativa (de valores eps 0.3 y 0.7). OPTICS reconoce una única agrupación y aparentemente ningún valor atípico, aunque los resultados finales son desarrollados en la Sección 5.

4.3.2.3 LOF (Local Outlier Factor)

LOF es un algoritmo de detección no supervisado encargado en detectar valores atípicos locales. Basado en la estimación de densidad, su tarea es la de asociar a cada punto un *score* LOF y, a partir de un umbral que es parámetro ajustado del algoritmo, discrimina entre valores con un LOF demasiado alto (punto en área densamente poblada) o demasiado bajo (*outlier*). Este valor es computado a partir de las distancias de alcanzabilidad de cada punto y pretende capturar la información respecto a su densidad local.

En su implementación, se indica el número de vecinos que se espera evaluar para cada punto con lo que calcular la distancia de alcanzabilidad, y *contamination*, valor entre 0 y 1 que

indica la parte de puntos que se deben limpiar por ser considerados anómalos. De esta función, los puntos que tengan etiqueta asociada de -1 son declarados anómalos.

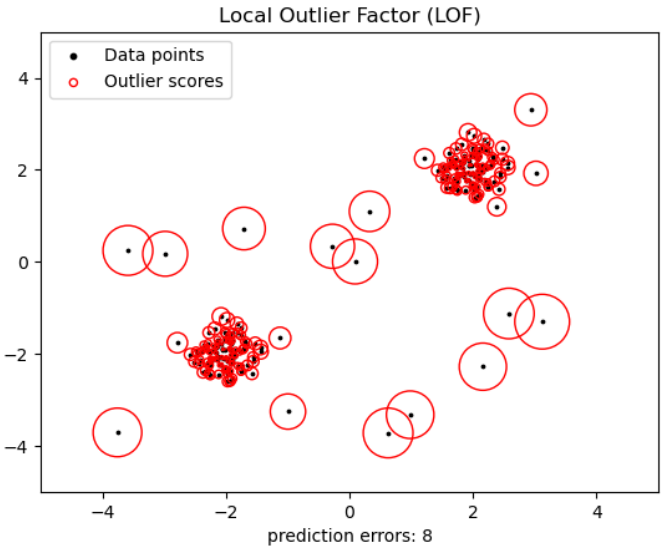


Figura 63: Representación de LOF scores según estimador de densidad [37]

En la Sección 5. se reserva una sección donde se describen y analizan los resultados pero, para permitir la continuación de la lectura del informe, se aclara que el modelo finalmente seleccionado por su eficacia de entrenamiento, capacidad de detección y relatividad de sus parámetros (siendo siempre referidos al tamaño de datos de entrada, el cual será cambiante según el CT y sobre todo la ventana temporal seleccionada), **se escoge para la detección el modelo DBSCAN**.

4.4 INTEGRACIÓN E INTERACCIÓN CON LA APLICACIÓN

En el apartado 4.2 Herramienta de visualización descriptiva, se ha comentado las virtudes y potenciales usos que puede tomar la librería Dash para el despliegue de resultados. En el apartado 4.3 Análisis agregativo y detección de valores anómalos, se han evaluado entre múltiples candidatos los mejores métodos que permiten 1) Reducir las informaciones

tratables para generar resultados visualizables y 2) Detectar valores atípicos dentro de Centros de Transformación específicos. En este apartado presente, la ambición es traer lo mejor de ambos mundos para generar una sinergia funcional entre ellos y dar soporte a la que se convertirá la versión final de la aplicación.

El punto inicial y más crítico es el papel de la interactividad en la herramienta y en Dash, este rol se le otorga a un único protagonista: los **callbacks.**

Los callbacks son **funciones Python** que son automáticamente llamadas por Dash cada vez que la propiedad de algún componente cambia. A continuación se muestra el ejemplo de un callback presente en la primera versión de la aplicación (ANEXO III. Malaga Visual Tool, Versión inicial).

```
# Variable selected
@app.callback(
    Output(component_id='variable-selected', component_property='children'),
    [Input(component_id='variable-dropdown', component_property='value')]
)
def update_output_variable(input_value1):
    return 'La variable seleccionada es: {}'.format(input_value1)
```

Figura 64: Ejecución simple de un callback

Como se puede ver, las entradas y salidas de la interfaz de la aplicación se declaran como argumentos del decorador **@app.callback**. En este caso, un cambio en la selección del valor del menú desplegable identificado como “variable dropdown” hace cambiar la información desplegada en los atributos dependientes (“children”) del elemento “variable selected”. La manera en la que se procesa esta información para presentar la salida viene explícitamente programada en la función que va directamente después. El *input* del decorador es el argumento de la función y lo que devuelve irá alojado en aquello que va asociado a salida del decorador.

Este ejemplo es el más básico de todos, donde se representa la elección de un menú desplegable como cadena de caracteres de un elemento de tipo párrafo en otra parte de la aplicación.

En conclusión, dentro de Dash las entradas y salidas de la aplicación son simplemente propiedades de componentes particulares. Cuando la propiedad de entrada cambie, la función del decorador del callback es llamada automáticamente y Dash lanza la ejecución de la función con el nuevo valor de entrada para devolver el nuevo valor de salida. Esto es denominado ***Reactive Programming*** y, a excepción que el valor se sobrescriba manualmente, permite la interactividad de la aplicación. Como se comentó, en Dash todo componente viene descrito por el conjunto de sus propiedades, y ahora con los callbacks es posible actualizarlo, ya sea el contenido de un gráfico o de un párrafo hasta su estilo o incluso las opciones disponibles de un menú.

Los callbacks más complejos son ahora capaces de tomar varias entradas o salidas en una misma función, incluir transiciones suaves a los cambios en gráficos… Sin embargo, hay una serie de buenas prácticas que es conveniente respetar. Entre ellas, cuando la tarea de guardado de datos es muy intensiva en memoria, se recomienda cargar el *dataframe* al inicializar la aplicación y, una vez cargado, segmentar los datos que interesen a la ejecución de un callback. Esto evita tener que cargar los datos cada vez que se llama y evita que el callback modifique los datos originales, creando sólo copias después de hacer el filtro. Como lógica de programación, como ocurre en la programación orientada a objetos, un callback nunca debería ser capaz de mutar las variables globales que quedan fuera de su alcance y así no interferir en una misma ejecución de la aplicación o incluso alterar sesiones cruzadas de usuarios distintos corriendo en distintos procesos.

Otro concepto interesante que es usado en esta aplicación son las nociones de **callbacks encadenados** y de **carga del estado** antes de solicitud del callback. Efectivamente, según como se configuren los callbacks a veces es necesario que la salida de un callback sea la entrada del otro, por mantener la coherencia de la web. Si se filtran los datos de Estados Unidos en lugar de Europa, no se permite dejar que, si luego hay una serie de capitales de país seleccionables, estas no cambien con la primera selección. De igual manera, hay veces que la aplicación no debería modificar los gráficos hasta que el usuario no acabe de completar todos los argumentos de entrada que desea. Esto viene marcado por la carga de

un estado, que una vez completado se ejecuta al seleccionar el verdadero *input* (ejemplo, un botón).

```
# Generate heat map
@app.callback(
    Output('map-graph','figure'),
    [Input('btn-updt-map','n_clicks')],
    [State('date-picker','date'),
     State('hour-selector','value'),
     State('variable-dropdown','value')]
)
def update_map(n_clicks, date_picker, hour_selector, variable_items):
    # Decompose the values of the date
```

Figura 65: Ejecución de callback con Estado

En la figura superior es posible ver el callback encargado de generar el mapa de calor del ANEXO III. Malaga Visual Tool, Versión inicial a partir de los datos seleccionados por el usuario. Si no fuera por el objeto *State*, haría falta un solo cambio en alguno de los parámetros para que el gráfico cambiase (gastando recursos inútilmente) aunque este no sea el deseo del interesado. En su lugar, se pone a su disposición un botón que alberga el mensaje de "Ejecutar" para que, una vez el menú tenga las características convenientes para su investigación, sí se ejecuten los gráficos que desea.

Después de explicar los callbacks, se reserva a continuación un comentario de la implementación que se hacen de estos para la aplicación en cuestión. En particular, el mayor reto existente en este momento fue el de ligar las dos páginas que se deseaba desplegar dentro de la herramienta: por un lado, la visualización descriptiva de la primera versión, y por otro lado la plataforma de análisis de valores anómalos. Afortunadamente, este problema cuenta con soluciones alto nivel que proponen en el github [36] de Faculty AI, mantenedores de los *Dash-Bootstrap-components* que ya se comentó en la Sección 4.). Dentro de las posibles opciones, se distinguen dos familias esencialmente: o por el uso de objetos **NavLinks**, o por el uso de **Pestañas**. El primero de ellos autoriza la navegación entre distintas páginas de una misma aplicación, de la misma manera que los hipervínculos permiten la navegación entre páginas HTML de Internet. Aunque útil para plataformas de gran tamaño, se pensó desde un primer momento en las tareas extra de comunicación requeridas entre dos aplicaciones Dash,

además de todas las tareas duplicadas que requieren las dos funciones que parten de los mismos datos, y por último se confirmó más tarde que la diferencia entre montar una o dos aplicaciones sobre un servidor encarece el servicio.

En consecuencia, se decide utilizar el uso de dos pestañas, que aparecen en el encabezado de la herramienta. Este método interpela la propiedad de activación de ambas pestañas para intervenir en la propiedad de *display* del cuerpo de contenido necesario.

```
html.Div(           # Tabs for window selection
    dbc.Tabs(
        [
            dbc.Tab(label="Heatmap", tab_id="heatmap", tab_style = {'width': '11%', 'textAlign': 'center'}),
            dbc.Tab(label="Outlier Detector", tab_id="outlier_detect", tab_style = {'width': '11%',
            'textAlign': 'center'}),
        ],
        id="body_tabs",
        active_tab="heatmap",
    ),
```

```
### APP CALLBACKS
# Display one of the two tab contents
@app.callback(
   Output(component_id='body_heatmap', component_property='style'),
   Output(component_id='body_outlier_detect', component_property='style'),
   [Input(component_id='body_tabs', component_property='active_tab')])

def update_tab_content(visibility_state):
    if visibility_state == 'heatmap':
        return {'display': 'block'}, {'display': 'none'}
    if visibility_state == 'outlier_detect':
        return {'display': 'none'}, {'display': 'block'}
```

Figura 66: Definición y callback del sistema de pestañas

Una vez asegurada la dicotomía entre pestañas, se montan los elementos y callbacks necesarios hasta completar la aplicación interactiva que finalmente se presenta como producto final del proyecto. Visiblemente, destaca la información condensada que aparece en la pestaña inicial, a lo cual se suma la segunda pestaña que permite seleccionar el CT y ventana temporal para acabar presentando una frase de evaluación junto a un gráfico Plotly 3D que permite la rotación, filtrado de familias entre *not selected*, *inliers* o *outliers* y en definitiva una herramienta multiplataforma, ágil y adaptada para trabajadores de todos los contextos.

Se ruega al lector que consulte el resultado estático de la plataforma en el ANEXO IV. Malaga Visual Tool, Versión Final donde se observan las dos vistas de la aplicación, así como el repositorio con el proyecto y recursos de instalación en [38].

4.5 TAREAS DE OPTIMIZACIÓN

A pesar de la completa funcionalidad de la aplicación llegados a este punto, se encontraron a lo largo del proyecto una serie de mejoras que mejoraron de una manera u otra la utilización de recursos, presentación de resultados o experiencia de usuario.

La primera fue el uso de Graph_Objects en lugar de una llamada a Plotly para la generación del mapa de calor. Los objetos de gráfico son estructura de datos con forma de árbol de relaciones que automáticamente son serializados en formato JSON para ser renderizados por la librería JavaScript Plotly.js. Estos árboles están compuestos de nodos llamados "atributos", y el módulo **plotly.graph_objects** contiene una multitud de clases Python que generan objetos visuales en Python, siendo estos "graph objects" instancias de las clases.

Por lo general, en Dash es común el uso de la librería plotly.express para usar funciones de más alto nivel que generan gráficos visualmente eficaces. Sin embargo, para el caso del mapa del calor convenía ser capaces de poder establecer parámetros internos inalcanzables si no fuera por los graph_objects, como por ejemplo son la altura, anchura o mensaje de etiqueta en los nodos al pasar el cursor por ellos (llamado *hover_label).*

```python
# FIRST VERSION: usando density_mapbox de Plotly Express - Pro: lo hace todo bien, Contra: muy delgado
fig = px.density_mapbox(obs_values, lat='lat', lon='long', z=variables_dict[variable_items], radius=25,
                        center=dict(lat=36.72016, lon=-4.42034), zoom=12, hover_name='substation', hover_data=
                        ['manufacturer','power'],
                        mapbox_style="stamen-terrain")

# SECOND VERSION: usando Density_mapbox de graph_objects - Pro: Customisable, the scope is adaptable to the
window size
fig = go.Figure(
    go.Densitymapbox(
        lat = obs_values['lat'], lon=obs_values['long'],
        z = obs_values[variables_dict[variable_items]],
        customdata = obs_values['substation'],
        hovertemplate = '<b>CT code: S%{customdata}<br>Value: %{z}</b><extra></extra>',
    )
)
```

Figura 67: Implementación del mapa de calor con plotly.express y plotly.graph_objects

La segunda de las mejoras es más lógica que de plataforma, y tiene que ver con la selección simple de CT en la ventana de análisis. En un momento dado, se reflexionó sobre el interés que tendría la selección múltiple de centros para la detección de valores atípicos, siendo el usuario capaz de ver simultáneamente los de varios. Sin embargo, se deja a propósito la selección simple pues de lo contrario el conjunto de entrenamiento del DBSCAN sería multimodal y las observaciones de uno podrían interferir con los de otro, causando falsos negativos (especialmente si los datos atípicos de un centro caen en una región densa de otro).

En lo relativo a mejoras operacionales, se han implementado métodos óptimos para la programación en Python (como es el uso de las ***comprehension lists*** para crear eficazmente listas sin pasar por bucles o el uso de la **vectorización** para computar gracias a la librería Numpy) o mejoras en la importación de datos (pd.read_csv en lugar del pd.read_excel obliga a convertir el archivo de partida pero tiene grandes beneficios en tiempos, **pasando de 2 minutos 11 segundos a 1.49 segundos**).

Para permitir el uso multiplataforma, se prestó atención a programar todos los tamaños (ya sean de elemento o de márgenes) en unidades relativas del tamaño de pantalla del visor (denominado ***viewport*** en nomenclatura css). Ejemplos de esto son la declaración de márgenes como porcentaje de su división padre en la que se encuentren.

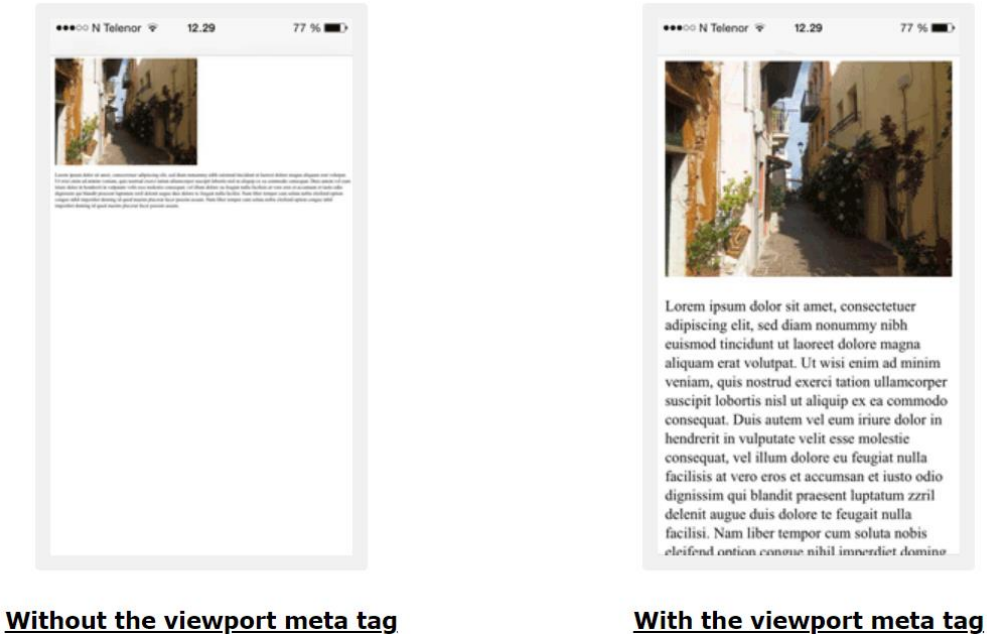


Figura 68: Diferencia por activación de viewport en la creación de la aplicación [39]

Sección 5. RESULTADOS Y ANÁLISIS

En este capítulo se analizan los resultados de los estudios llevados en la Sección 4. , tanto para la comparativa de modelos que intervienen en la reducción dimensional como en los candidatos para la identificación de valores atípicos. En base a pruebas medibles y comparables, se extraen enseñanzas clave de cada tipo de técnica y así se concluye sobre la idoneidad de uno u otro para la aplicación.

5.1 ANÁLISIS DE LA REDUCCIÓN DIMENSIONAL

Para comprobar la calidad de la operación, es necesario partir de un método simple que aplique a las transformaciones hechas por todas las técnicas. Considerando que las observaciones son etiquetadas por su valor de Centro de Transformación del que vengan, se entrena un modelo de clasificación tipo **Random Forest Classifier (RFC)** que se encargue de predecir el Centro de Transformación al que con mayor probabilidad parecen pertenecer los datos de Validación (que estaban incluidos en la base de entrenamiento) y de Prueba (observaciones no usadas). El hecho de valorar ambas puntuaciones tiene interés para evaluar el **compromiso sesgo – varianza de cada modelo**, y así conocer **si se da un caso de infra o sobreentrenamiento**. En *Machine Learning*, un modelo que se entrena con demasiadas iteraciones sobre un conjunto de entrenamiento puede luego no generalizar bien a la hora de solucionar el problema en otros conjuntos. Si se considera por ejemplo un problema de clasificación, un entrenamiento exagerado sobre un conjunto de entrenamiento haría mejorar más y más los resultados en los *scores* en el conjunto de Validación, pero en paralelo deterioraría los resultados en la base de Prueba, remontando del mínimo donde habría que haber abandonado el entrenamiento (de hecho, este es el principio teórico detrás del *Early Stopping* que se ha implementado).

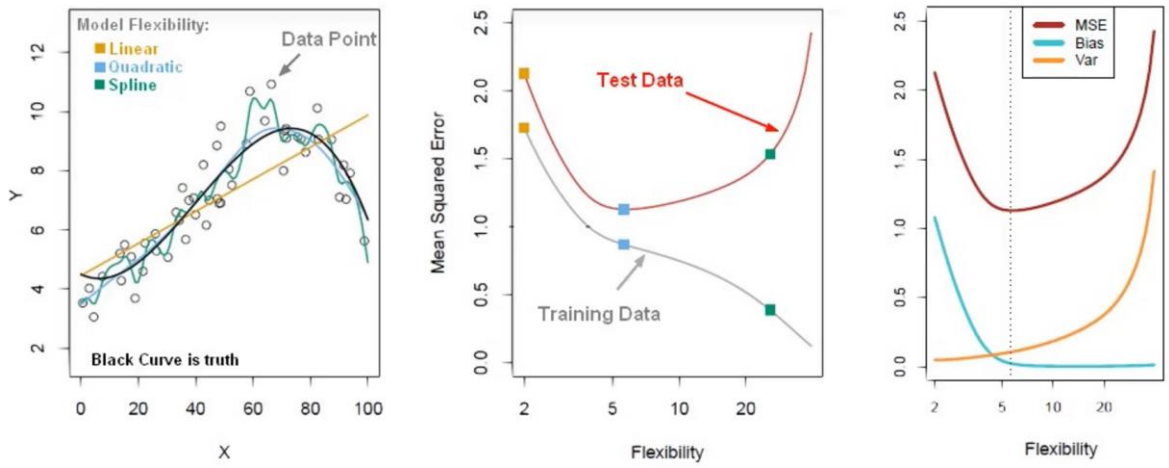


Figura 69: Problema de Generalización del modelo y diferencia en errores de Entrenamiento y Prueba [40]

Este fenómeno explica la necesidad de separar los datos en parte de entrenamiento y prueba, y la prueba que evaluará de manera justa a todos los modelos corresponderá con **la eficacia que tiene el modelo RFC en clasificar las observaciones** según la base de datos original, y después de las reducciones hechas por cada uno. Ante esta declaración, dos fenómenos son anticipables: en primer lugar, ya se ha comentado que **el problema no es linealmente separable en 2D o 3D**, lo que significa que los porcentajes de acierto del RFC no serán ni mucho menos óptimos. El RFC es un meta-estimador que entrena una serie de árboles de clasificación en varios subconjuntos de la base de entrenamiento y, sobre nuevos datos, agrega las predicciones de cada uno de ellos para mejorar la precisión de la meta-predicción, a la vez que controla mejor el sobreentrenamiento (el tamaño al que se ve enfrentado cada árbol es menor que el total). Pero a pesar de esto, el modelo no es infalible y será visto a continuación, pero afortunadamente este no es el propósito de este estudio, sino el de poder elegir el modelo que otorgue al RFC el mejor resultado. En segundo lugar, es posible adelantar que los resultados de las predicciones del RFC serán siempre peores basándose en las codificaciones de las observaciones que en el *dataset* original. Evidentemente, al disponer **la tabla original mayor información no redundante de partida** (cinco variables, frente a tres dimensiones), su resultado será **cota superior del resto**. Sin embargo, esto no evita que haya algunos más cercanos que otros de este límite, y es de este ordenamiento de donde emergen las conclusiones.

```
### Part 5: Test scores
rfc = RandomForestClassifier(n_estimators=200, max_depth=5)

labels = ['Original', 'PCA', 'AE', 'Stacked_AE', 'NL_St_AE', 'tSNE']
scores_benchmark_test = pd.DataFrame(columns=labels)
scores_benchmark_test['Original'] = cross_val_score(rfc, X_test_std, y_test, cv=5)
scores_benchmark_test['PCA'] = cross_val_score(rfc, scores_test, y_test, cv=5)
scores_benchmark_test['AE'] = cross_val_score(rfc, ae_codings_test, y_test, cv=5)
scores_benchmark_test['Stacked_AE'] = cross_val_score(rfc, codings_hidden_test, y_test, cv=5)
scores_benchmark_test['NL_St_AE'] = cross_val_score(rfc, nl_st_codings_test, y_test, cv=5)
scores_benchmark_test['tSNE'] = cross_val_score(rfc, tsne_results_test, y_test, cv=5)
```

Figura 70: Ejecución de Benchmark sobre el conjunto Prueba

Scores Benchmark:

	Original	PCA	AE	Stacked_AE	NL_St_AE	tSNE
0	0.540316	0.512190	0.489222	0.455014	0.509366	0.491502
1	0.543194	0.522832	0.495520	0.452137	0.502416	0.494869
2	0.545993	0.515530	0.489466	0.455474	0.509666	0.484361
3	0.532635	0.513684	0.489846	0.455093	0.502226	0.487402
4	0.551857	0.523349	0.490715	0.460904	0.507819	0.486045

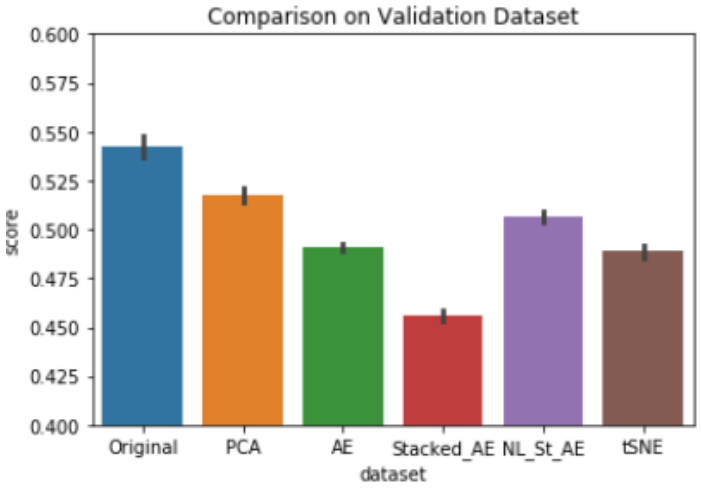


Figura 71: Resultados del RFC sobre Validation

Score Benchmarks Test:

	Original	PCA	AE	Stacked_AE	NL_St_AE	tSNE
0	0.556603	0.527820	0.489510	0.460728	0.506132	0.512922
1	0.550071	0.519562	0.487026	0.451956	0.513177	0.507399
2	0.551186	0.520981	0.479424	0.447496	0.513379	0.511555
3	0.551591	0.517231	0.477296	0.449726	0.506994	0.502331
4	0.545814	0.524731	0.491283	0.447192	0.506588	0.500405

Figura 72: Resultados del RFC sobre Test

Viendo las diferencias en rendimiento de la clasificación entre el espacio original y los diferentes espacios, se observan resultados bastante concluyentes. En base a la validación con k = 5 (descomposición del grupo *Training* en k grupos para contar con tantas evaluaciones), se aprecia que la **clasificación no consigue superar apenas un 55% de éxito** para el caso de los datos originales. A partir de este valor, **los modelos de PCA y de Autocodificador Multicapa No Lineal** rivalizan por colocarse como técnica ideal. Les siguen los modelos de Autocodificador simple y t-SNE le siguen y por último acaba el Autocodificador Multicapa Lineal (que probablemente no ha podido explotar las bondades de las capas intermedias debido al débil margen existente entre entrada y codificación). En términos de eficiencia, los modelos más rápidos fueron el PCA (tiempo de ajuste = 3.13 s) y el AE simple (tiempo de ajuste = 2.5, optimizado con el paquete Keras).

Si se acuden a los resultados de Test, el orden queda intacta a excepción del t-SNE que consigue superar el 50% en un conjunto considerablemente más ligero (x10 de disminución de volumen).

La conclusión es que, para mantener fielmente la distribución de puntos observados a la vez que se posibilita su representación visual, la mejor opción es **reducir las dimensiones de cinco a tres por medio de un PCA.**

5.1.1 INTERPRETABILIDAD DEL PCA

En base a la generación de la base sobre los tres ejes ortogonales óptimos, las observaciones sufren una transformación de tipo proyectiva. Sin embargo, fruto de esta maximización de la varianza puede producirse una distorsión sobre la interpretabilidad y una desconexión de la medición con la relación que la ata a variables físicas. Comúnmente, se realiza pues un estudio sobre los pesos (*loadings*) que tienen los vectores propios sobre la base de variables originales. Cuanto más diagonal es está matriz (o pseudo diagonal en el caso donde la matriz es rectangular), mejor interpretabilidad tienen los vectores y por lo tanto, más fáciles son las semejanzas que se pueden trazar entre valores extremos en los Componentes Principales y su correspondencia con la realidad.

Indice	Tensión	Intensidad	Potencia Activa	Potencia inductiva	Potencia Capacitiva	Varianza Capturada
PC1	-0,04	**0.60**	**0.59**	0.5	0.18	*0.49*
PC2	**0.71**	0.15	0.19	-0,11	**-0,66**	*0.22*
PC3	**-0,71**	0.14	0.12	-0,13	**-0,67**	*0.18*

Tabla 9: Pesos de los vectores propios del PCA en la base original

Los Componentes Principales se separan en las variables originales según coeficientes desiguales, pero destacan siempre parejas de variables que extrae el significado de cada uno:

- PC1: Fuerte presencia en Intensidad y Potencia Activa. Este eje alberga la mayoría de la varianza acumulada (casi 50%) y define los casos de **sobreintensidad ante desbalances fuertes de carga**. Con la intensidad, la potencia activa crece en correspondencia y a causa de factores de potencia bajo, arrastra consigo la potencia reactiva.
- PC2: Esta componente acentúa los altos valores de Tensión y bajos de Potencia Capacitiva. Este eje (que acoge el 22% de la varianza) fija su atención en los casos donde **la tensión se encuentra en valores altos superando el de regulación**. Para casos donde la red se vea descompensada por falta de potencia capacitiva, la tensión tiende al aumento y queda registrado a lo largo de este eje.
- PC3: Tercera componente perpendicular. Centrado en valores bajos de tensión o de capacitiva, esta componente se fija en las **bajadas de tensión fuerte** (causados por fallas de equipos, encendido de grandes cargas, …).

5.2 ANÁLISIS DEL DETECTOR DE VALORES ANÓMALOS

Para agilizar el entrenamiento sin romper la coherencia del estudio, se decidió tomar un fragmento menor de la base de codificaciones para proceder a la evaluación de resultados.

```
# Reduce the training data after shuffling
indices = np.arange(X_std.shape[0])
np.random.shuffle(indices)

X_std_tr = X_std[indices][:N]
codings_tr = codings[indices][:N]
```

Se entrena cada uno de los modelos y, para capturar la viabilidad de su implementación dentro de una *app* de tiempo real, se miden los tiempos de ejecución correspondientes.

MODELO	DBSCAN eps = 0.3	DBSCAN eps = 0.7	OPTICS	LOF
Tiempo entrenamiento	*1'037*	*1'911*	*1"15'60*	*00'5514*

Tabla 10: Tiempos de entrenamiento para DBSCAN, OPTICS y LOF

A la vista de los resultados, DBSCAN es un modelo válido que supera el filtro de la latencia para responder a la aplicación. Sin embargo, a falta de conocer su eficacia en la detección, OPTICS supondría un problema grande, alcanzando los dos dígitos para lograr entrenarse.

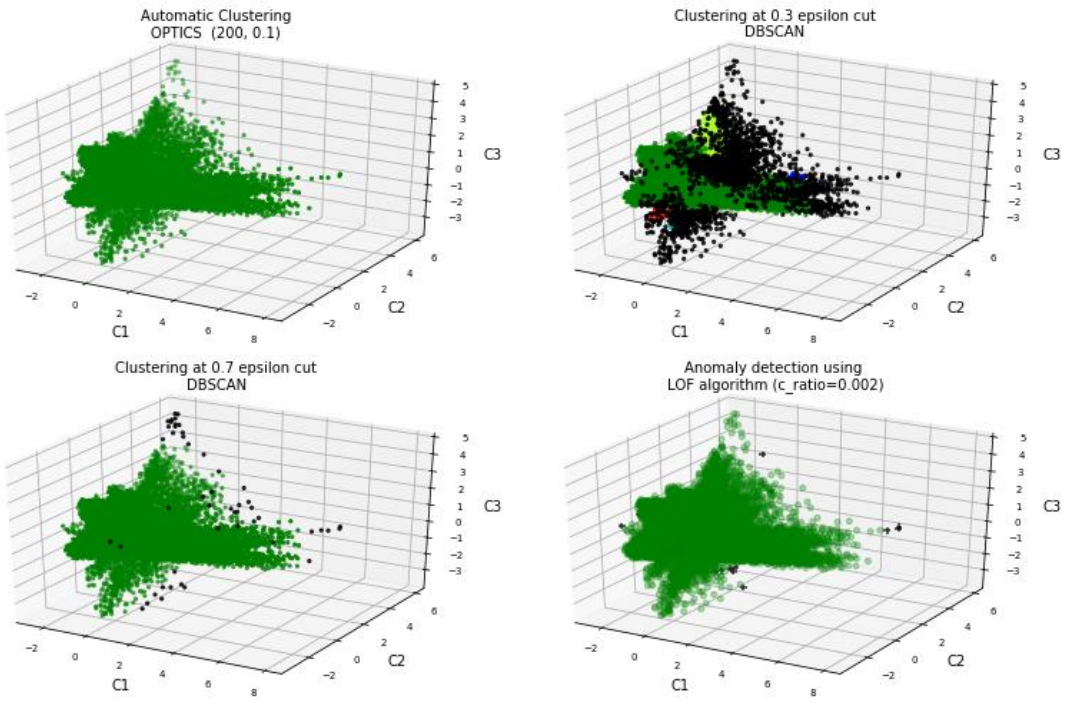


Figura 73: Resultados de detección de valores anómalos con los cuatro modelos

```
Algorithm used for outlier detection: OPTICS
Number of counted outliers out of 20000:
0

Algorithm used for outlier detection: DBSCAN
Number of counted outliers out of 20000:
SMALL_EPS: 1998, HIGH_EPS: 50

Algorithm used for outlier detection: Local Outlier Factor
Number of counted outliers out of 20000:
40
```

Tabla 11: Volumen de detección de observaciones atípicas

Basándose en el estudio y visualización de **20.000** registros (12.5%):

- El algoritmo de OPTICS, a pesar de ser el más dinámico y autorregulador de hiperparámetros según una optimización interna, es también **causa de su lentitud**, y juega en contra de la baja latencia que se busca en esta página. Además, su detección automática no detecta ningún valor anómalo y esto **impide calibrar** lo conservador o permisivo de nuestros resultados.

- DBSCAN es método **eficaz** en recursos, **relativo al tamaño** y con resultados coherentes a los esperados. Se selecciona este método por conseguir un buen **compromiso entre agilidad y rendimiento**. Además, a pesar de que los datos anómalos del futuro se reagrupen en corpúsculos lejanos al bloque central (lo cual podría llegar a ocurrir, fruto de un error sistemático), DBSCAN lo detecta de igual manera (cuestión a la que el LOF es vulnerable).

- LOF es un algoritmo particularmente bueno para la búsqueda de nuevos valores extremos que no incluya el conjunto de entrenamiento que, sin embargo, se espera que lleguen de la misma distribución de partida. Sin embargo, debido a la elección del hiperparámetros *contamination*, **se estaría condicionando al algoritmo a encontrar siempre valores que descartar**, según un ratio fijo con respecto al total de registros, lo cual invalida el resultado (aunque si tendrá potencial si se pretende extender la predicción de *outliers* en datos futuros, lo cual queda fuera del alcance del proyecto).

La solución final es por lo tanto escoger un modelo **DBSCAN** por su facilidad y agilidad, con un valor de eps = 0.55 (para mantenerlo conservador frente a la reducción dimensional y siendo coherentes con el tamaño de una subselección de CT).

El gráfico de la visualización final que es utilizada para la aplicación es similar a la siguiente.

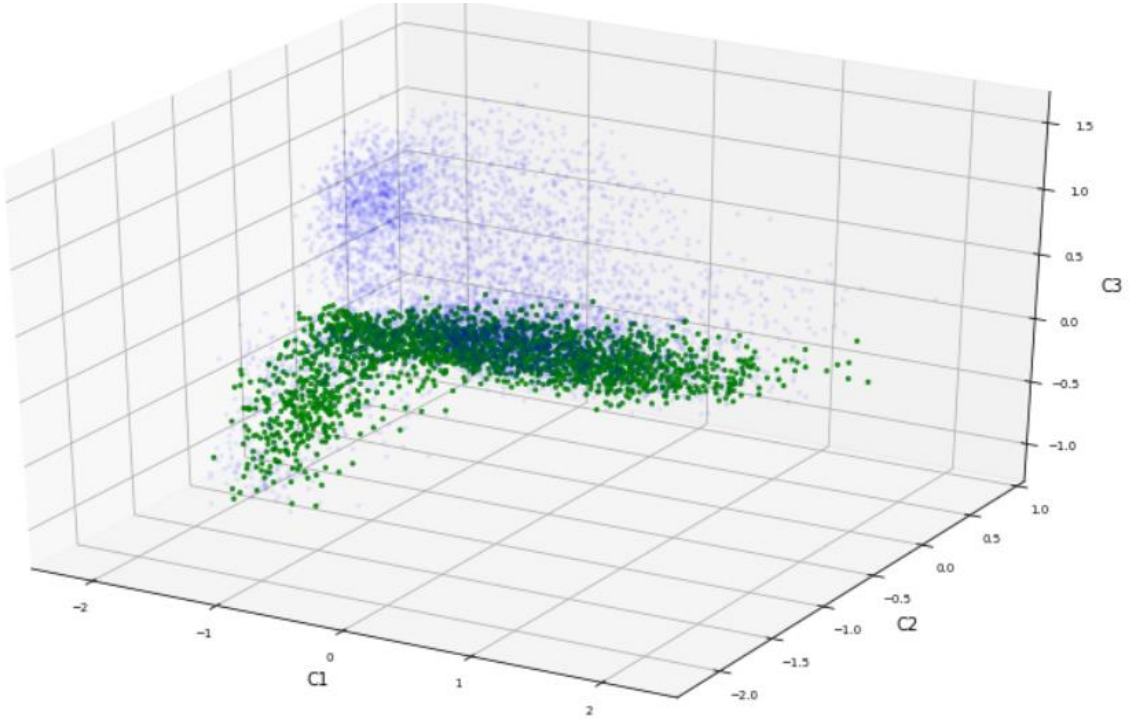


Figura 74: Visualización del detector de valores anómalos

En este gráfico, los valores azules se corresponden con el conjunto de observaciones pertenecientes a un CT que no se producen dentro del rango temporal seleccionado. Si es el caso contrario, aparecerán en verde si son valores considerados típicos y como cruz roja si se califican de anómalos.

Es importante precisar que, una vez el usuario introduce su entrada, el DBSCAN sí es entrenado con todas las observaciones de ese CT, pues el comportamiento es interno a él a pesar de la estacionalidad que pueda existir y, sobre todo, comprende que en el caso de seleccionar ventanas temporales muy estrechas, entrenar con una base de entrenamiento reducida de pocas observaciones podría ser igual de fatal que no entrenarlo siquiera.

Sección 6. CONCLUSIONES Y FUTUROS PASOS

6.1 RESULTADOS DEL PROYECTO

A lo largo de este proyecto de casi un año, se ha conseguido procesar, explorar, extraer información y presentarla dentro de una aplicación multiplataforma, ágil y distribuible para el soporte y apoyo de tareas de mantenimiento de Centros de Transformación de la ciudad de Málaga.

La aplicación Dash es holística frente a datos y dispositivos, y consigue generar una herramienta analítica de presentación de datos descriptivos y predictivos alrededor de observaciones medidas sobre las líneas de centros de la red eléctrica. La latencia de la página web es inferior al segundo, presenta una interfaz intuitiva al usuario y que responde a las problemáticas que hoy en día la compañía encuentra a la hora de explotar las extensas bases de datos históricas que poseen de las variables físicas que son capaces de medir.

Retomando los objetivos establecidos en un inicio y que se encuentran en 1.4 Objetivos, el proyecto ha conseguido una reponsividad e interacción a la altura de un servicio web, facilita la generación automática de informes de salud y congestión de la red, se basa en un modelo robusto y agnóstico al dato y dispositivo y, por último, efectivamente apalanca el análisis y conclusiones del estudio sobre herramientas de reducción dimensional que contienen menos del 15% del espacio original.

De cara al CIC, esta herramienta se encuentra ahora en un punto dulce donde, una vez superada la cumbre de aprendizaje que era obligatorio superar al principio, admite miles de ampliaciones y variaciones de adaptación para nuevos colaboradores de la Cátedra que quieran extraer información de valor con el poder de la Ciencia de Datos, muy necesario en el siglo XXI y en el futuro que queda por delante.

6.2 DISTRIBUCIÓN DE LA HERRAMIENTA

En este apartado se aborda, una vez acabado el desarrollo de la primera versión completamente funcional de la aplicación, la distribución de esta al resto de partes interesadas. Gracias a los entornos de **conda** y **pip**, fue posible generar en base a las librerías usadas en el entorno virtual de desarrollo dos archivos: el **environment.yml** y **requirements.txt.** Si una persona ajena del grupo quiere visualizar la aplicación dentro del *host* local como se ha venido haciendo hasta ahora (127.0.0.1), basta con descargar el archivo py, organizar las carpetas "DATA" (o dejar el código como está para rescatar el archivo de Github) y "assets" a su alrededor para replicar el directorio y, dentro de Visual Studio o cualquier otro IDE, generar el entorno ejecutando unos de estos dos comandos:

```
conda env create -f environment.yml     //     pip install requirements.txt
```

En ambos casos, dentro del directorio donde se haya dirigido el usuario, y tras darle el nombre al entorno virtual, será capaz de lanzar la aplicación y de ver el resultado en tiempo real tras unos minutos.

También se trató de generar un objeto .exe que permitiese ejecutar la aplicación solo una selección en el archivo, para ello haciendo uso de la librería **pyinstaller**. Sin embargo, se dan problemas con la librería Flask.js que permite la interactividad de la página.

Por último, a Smart City Málaga u otros grupos se les explica el procedimiento para poder construir esta aplicación como servicio usando terceras partes especializadas en ofrecer servidores para aplicaciones de este tipo: **PythonAnywhere.com**, y **Heroku**.

Heroku es originalmente la plataforma como servicio (PaaS) que ayudó en origen a desarrolladores Ruby a desplegar sus aplicaciones. Apenas en 2012 abrieron una rama para soportar lenguajes interpretados como Python. PythonAnywhere sin embargo comienza con el objetivo de desplegar un entorno Python completamente configurable. Todos los paquetes y librerías se encuentran en su última versión y han sido probados. Mientras tanto, Heroku es un espacio en blanco donde los requerimientos son definidos por el archivo requirements.txt

PythonAnywhere se comporta más como un servidor tradicional donde el almacenamiento local persiste y es posible conectar por SSH, usar la consola web, editar los archivos de configuración… Con la diferencia de no ser realmente un servidor tradicional (pues la máquina física se abstrae para que la aplicación pueda correr en servidores en paralelo).

En comparación, Heroku internamente crear un contenedor distinto para cada aplicación web que se lanza como nueva instancia. Para expertos del sector, PA es visto como un servidor de desarrollo. Aunque, Heroku es recomendado desde la página web oficial de Dash, PA cuenta con funcionalidades muy interesantes, como la posibilidad de poder forzar el protocolo HTTPS, o incluir control de accesos a la web incluyendo un registro de usuarios con contraseña. En ambos es posible crear una cuenta de servicio gratuito inicial y proveen de una URL distribuible.

Desgraciadamente, ambos incluyen limitaciones de tamaño que para el programa se queda corto por poco. Para solventarlo, se creó un entorno virtual distinto al de desarrollo y así descargar las librerías mínimas, además de recuperar los datos a través del **link Raw que ofrece github**. Sin embargo, como se puede ver en la foto el proyecto se queda muy cerca de poder entrar en la plataforma:

```
2021-07-07T13:02:39.862346+00:00 app[web.1]: [2021-07-07 13:02:39 +0000] [194] [INFO] Booting worker with pid: 194
2021-07-07T13:02:39.862347+00:00 app[web.1]: [2021-07-07 13:02:39 +0000] [193] [INFO] Booting worker with pid: 193
2021-07-07T13:02:55.089776+00:00 heroku[web.1]: Process running mem=543M(106.2%)
2021-07-07T13:02:55.105476+00:00 heroku[web.1]: Error R14 (Memory quota exceeded)
```

Si se consultan los medios de pago, ambos tienen un coste de alrededor de 5$ y de 25$ respectivamente para la ampliación hasta 1 Gb. Como opinión personal, se aconseja mejor el uso de PA más que Heroku, al disponer de una interfaz más amigable (alejada de los comandos en Bash de git para modificar un archivo) además de disponer de control de acceso por contraseña y de protocolo seguro internet (https).

Para no dejar el trabajo de documentación en balde, se adjunta en la sección de Anexos dos documentos explicando paso a paso el despliegue de la aplicación en ambas páginas, y dejar la decisión final al sponsor del proyecto.

6.3 FUTUROS PASOS Y AMPLIACIONES

Más allá de la limitación del alcance llevada a cabo en este proyecto, se proponen a continuación una serie de ampliaciones que tienen justificación por el caso de uso y encajarían con la aplicación actual:

- **Crecimiento orgánico**: integración de nuevos atributos, arquitecturas más complejas, …
- Lectura de **información en tiempo real** y montar la aplicación en la web de operaciones internas de la página web de Endesa, ambos muy dependientes de la gestión de datos interna que existan, tiempos de actualización, políticas de comunicación y privacidad, …
- **Problema de escalabilidad** en términos de potencia de computación en el entrenamiento del DBSCAN si los datos se vuelven muy voluminosos. Este defecto se ve atenuado por la obligatoriedad de la selección simple, pero también podría evitarse recuperando los parámetros de un modelo preentrenado para cada CT, o bien seleccionando aleatoriamente una muestra menor para entrenar nuevamente el DBSCAN según la lógica actual.
- **Integración con el trabajo humano en planta**: hipótesis y casos de negocio.
- **Distribución como página web**: https://dash.plotly.com/deployment y 6.2 Distribución de la herramienta.

Sección 7. BIBLIOGRAFÍA

[1] Grigory A. Trestman "Primary and Secondary Sources of Electrical Energy", 2017, https://doi.org/10.1117/3.2268619.ch5

[2] Sistema del suministro eléctrico, Wikipedia.org: https://es.wikipedia.org/wiki/Sistema_de_suministro_el%C3%A9ctrico

[3] Battistelli, E. "El futuro de la energía eléctrica: todos seremos prosumidores". Junio 2018. https://www.pwc.com.ar/es/publicaciones/eye-to-eye/future-in-sight/futuro-energia-electrica-prosumidores.html

[4] Blanco, S. "Mercado Eléctrico: Los servicios de ajuste del sistema eléctrico peninsular". Febrero 2018. https://www.aeqenergia.com/index.php/blog/mercado-electrico-iii-los-servicios-de-ajuste-del-sistema-electrico-peninsular

[5] B. S. Tatera and H. L. Smith, "The evolution of monitoring and controlling in electric power substations," 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008, pp. 1-5, doi: 10.1109/PES.2008.4596842: https://ieeexplore.ieee.org/document/4596842

[6] C. Yao, Z. Zhao, Y. Mi, C. Li, Y. Liao and G. Qian, "Improved Online Monitoring Method for Transformer Winding Deformations Based on the Lissajous Graphical Analysis of Voltage and Current," in IEEE Transactions on Power Delivery, vol. 30, no. 4, pp. 1965-1973, Aug. 2015, doi: 10.1109/TPWRD.2015.2418344. https://ieeexplore.ieee.org/document/7086082

[7] A. Singh and P. Verma, "A review of intelligent diagnostic methods for condition assessment of insulation system in power transformers", 2008 International Conference on Condition Monitoring and Diagnosis, 2008, pp. 1354-1357, doi: 10.1109/CMD.2008.4580520.: https://ieeexplore.ieee.org/document/4580520

[8] United States Department of the Interior, Bureau of Reclamation, "Transformer Diagnostics", Facilities Instructions Standards and Techniques, 2003, colum3 3-31: https://www.usbr.gov/power/data/fist/fist3_31/fist3-31.pdf

[9] P. H. Mukti, F. A. Pamuji, B. S. Munir, "Implementation of Artificial Neural Networks for Determining Power Transfomer Condition", in ADCONP 2014 Hiroshima: https://folk.ntnu.no/skoge/prost/proceedings/adconip-2014/pdf/SUBS101TO128/0109/0109_FI.pdf

[10] C. Yao, Z. Zhao, Y. Mi, C. Li, Y. Liao and G. Qian, "Improved Online Monitoring Method for Transformer Winding Deformations Based on the Lissajous Graphical Analysis of Voltage and Current," in IEEE Transactions on Power Delivery, vol. 30, no. 4, pp. 1965-1973, Aug. 2015, doi: 10.1109/TPWRD.2015.2418344.: https://ieeexplore.ieee.org/document/7086082

[11] A. Jahromi, R. Piercy, S. Cress, J. Service and W. Fan, "An approach to power transformer asset management using health index," in IEEE Electrical Insulation Magazine, vol. 25, no. 2, pp. 20-34, March-April 2009, doi: 10.1109/MEI.2009.4802595.

[12] M. Dong, A. B. Nassif and B. Li, "A Data-Driven Residential Transformer Overloading Risk Assessment Method," in IEEE Transactions on Power Delivery, vol. 34, no. 1, pp. 387-396, Feb. 2019, doi: 10.1109/TPWRD.2018.2882215.

[13] Kunlong Chen, Jiuchun Jiang, Fangdan Zheng, Kunjin Chen, "A novel data-driven approach for residential electricity consumption prediction based on ensemble learning", Volume 150, 2018, Pages 49-60: https://www.sciencedirect.com/science/article/abs/pii/S0360544218302561

[14] Javier Leiva, José A. Aguado, Ángel Paredes, Pablo Arboleya, "Data-driven flexibility prediction in low voltage power networks," International Journal of Electrical Power & Energy Systems, Volume 123, 2020, 106242, ISSN 0142-0615: https://www.sciencedirect.com/science/article/abs/pii/S0142061520306293

[15] Brinch, Sara, "Charles-Joseph Minard's map of Napoleon's flawed Russian campaign: An ever-current classic", sciencenorway.no, Mars 2019: https://sciencenorway.no/blog-blog-from-numbers-to-graphics-statistics/charles-joseph-minards-map-of-napoleons-flawed-russian-campaign-an-ever-current-classic/1618695

[16] Libin Yang. An Application of Principal Component Analysis to Stock Portfolio Management. Department of Economics and Finance, University of Canterbury, January 2015: https://ir.canterbury.ac.nz/bitstream/handle/10092/10293/thesis.pdf?sequence=1

[17] Autoencoder, march 2020, Wikipedia.org: https://en.wikipedia.org/wiki/Autoencoder

[18] Página de Introducción a Dash: https://dash.plotly.com/introduction

[19] "Compare Dash and Qlik Sense", g2.com: https://www.g2.com/compare/plotly-dash-vs-qlik-sense

[20] Página Introducción a Scikit-learn: https://scikit-learn.org/stable/tutorial/basic/tutorial.html

[21] Imagen de archivo OARSIS, Sept 2015 : http://www.oarsis.com/portfolio/ybvr/

[22] Josh Allard, Andrew Roskuski, and Mark Claypool. 2020. Measuring and Modeling the Impact of Buffering and Interrupts on Streaming Video Quality of Experience. In The 18th International Conference on Advances in Mobile Computing and Multimedia (MoMM '20), November 30-December 2, 2020, Chiang Mai, Thailand. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3428690.3429173

[23] Link a la base de datos, restringido a perfiles admitidos por el autor: https://drive.google.com/file/d/1YQQ3t3KQId_5vHH4yi3ez-y5sS2ellfl/view?usp=sharing

[24] Jhalora Surendra, "Import and export electrical energy", Sept. 2020 : https://www.securemeters.com/uk/white_papers/import-and-export-of-electrical-energy/

[25] Himanshu Sharma, "Automating Exploratory Data Analysis – Part 1", Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-1-f5f2b7d548e5

[26] Himanshu Sharma, "Automating Exploratory Data Analysis – Part 2", Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-2-f03083f42ecf

[27] Himanshu Sharma, "Automating Exploratory Data Analysis – Part 3", Feb 2021, medium.com: https://medium.com/swlh/automating-exploratory-data-analysis-part-3-d04352b83072

[28] How safe are the streets of Santiago?, Marcelo Rovai, towardsdatascience.com: https://towardsdatascience.com/how-safe-are-the-streets-of-santiago-e01ba483ce4b

[29] Dash Enterprise App Gallery: https://dash-gallery.plotly.host/Portal/

[30] Theme Explorer, facultyAI: https://dash-bootstrap-components.opensource.faculty.ai/docs/themes/explorer/

[31] "Cross-validation: evaluating estimator performance": https://scikit-learn.org/stable/modules/cross_validation.html

[32] Chollet, F., "Building Autoencoders in Keras", May 2016: https://blog.keras.io/building-autoencoders-in-keras.html

[33] van der Maaten, Laurens & Hinton, Geoffrey. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research. 9. 2579-2605.: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

[34] Métodos de *Clustering* disponibles en Scikit-learn, scikit-learn.org : https://scikit-learn.org/stable/modules/clustering.html

[35] Chauhan Singh, Nagesh, "DBSCAN Clustering Algorithm in Machine Learning", April 2020: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

[36] Página Github de recursos de Dash-Bootstrap-Callbacks: https://github.com/facultyai/dash-bootstrap-components/tree/main/examples

[37] "Outlier detection with Local Outlier Factor (LOF)", Sphinx-Gallery: https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html

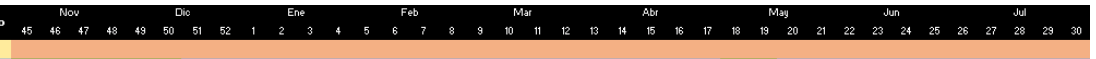
[38] Página Github del proyecto Malaga Visual Tool: https://github.com/andres-garcia97/tfm_cic/tree/main

[39] Responsive Web Design – The Viewport: https://www.w3schools.com/css/css_rwd_viewport.asp

[40] Tripathi, M., "Underfitting and Overfitting in Machine Learning", June 2020: https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning

ANEXO I. DIAGRAMA DE GANTT

Grupo	Indice	Tarea	Duración (semanas)	Estado de Riesgo
PROYECTO TFM			145	In Progress
INTRODUCCION AL PROYECTO	1		19	
	1.1	Presentación de la temática	2	Completed
	1.2	Reuniones junto a coordinadores	6	Completed
	1.3	Lectura de informes previos (Becas CIC verano 2020)	2	Completed
	1.4	Sensibilización del tema (papers académicos)	4	Completed
	1.5	Discusión sobre requerimientos y objetivos	5	Completed
ESTUDIO PRELIMINAR	2		14	
	2.1	Evaluación calidad del dato	3	Completed
	2.2	Trato de casos límites	2	Completed
	2.3	Datos no válidos	3	Completed
	2.4	Técnicas de sustitución y preprocesamiento	2	Completed
	2.5	Viabilidad de potenciales herramientas de visualización	4	Completed
ANALISIS E INTEGRACION DE DATOS	3		20	
	3.1	Análisis Exploratorio de Datos	10	Completed
	3.2	Análisis comparativo entre plataformas	5	Completed
	3.3	Preparación y limpiado del dataframe	3	Completed
	3.4	*Slicing* de datos de entrada	2	Completed
GENERACION DE FIGURAS	4		21	
	4.1	Primeros prototipos	8	Completed
	4.2	Construcción de narrativa ilustrativa	7	Completed
	4.3	Optimización de ventana y reducción de espacios vacíos	6	Completed
DESARROLLO DE PRIMER MODELO	5		32	
	5.1	Definición de cuadrícula y secciones de la herramienta	3	Completed
	5.2	Integración de cuadro de mandos del usuario	3	Completed
	5.3	Interactividad estable	4	Completed
	5.4	Desarrollo de métodos *Dash*	12	Completed
	5.5	Definición de *callbacks*	8	Completed
	5.6	Selección de *palette* y fuentes	2	Completed
INTEGRACION Y DOCUMENTACION	6		4	
	6.1	Test multiplataforma	1	Completed
	6.2	Generación de archivo metadata .yml	2	Completed
	6.3	Demo inicial y feedback	1	Completed
MODELOS INTELIGENTES	7		17	
	7.1	Replanificación del proyecto	1	Completed
	7.2	Compresión de datos PCA vs AE	9	Completed
	7.3	Clusterización de puntos con modelos: DBSCAN, OPTICS y LOF	5	Completed
	7.4	Comparativa de perdida de informacion entre original y reconstruccion	2	Completed
ANÁLISIS DE RESULTADOS	8		7	
	8.1	Casos de uso y utilidad de la herramienta	4	Completed
	8.2	Análisis inteligente de la fuente	2	Completed
	8.3	Conclusiones y siguientes pasos	1	Completed
ENTREGA Y CORRESPONDENCIA CON ICAI	9		11	
	9.1	Redacción de Anexo B	2	Completed
	9.2	Presentación final	1	In Progress
	9.3	Actualización de archivos y *Delivery*	2	In Progress
	9.4	Redacción de memoria final	4	In Progress
	9.5	Redacción de artículo	2	In Progress



ANEXO II. REFLEXIÓN SOBRE LOS O.D.S.

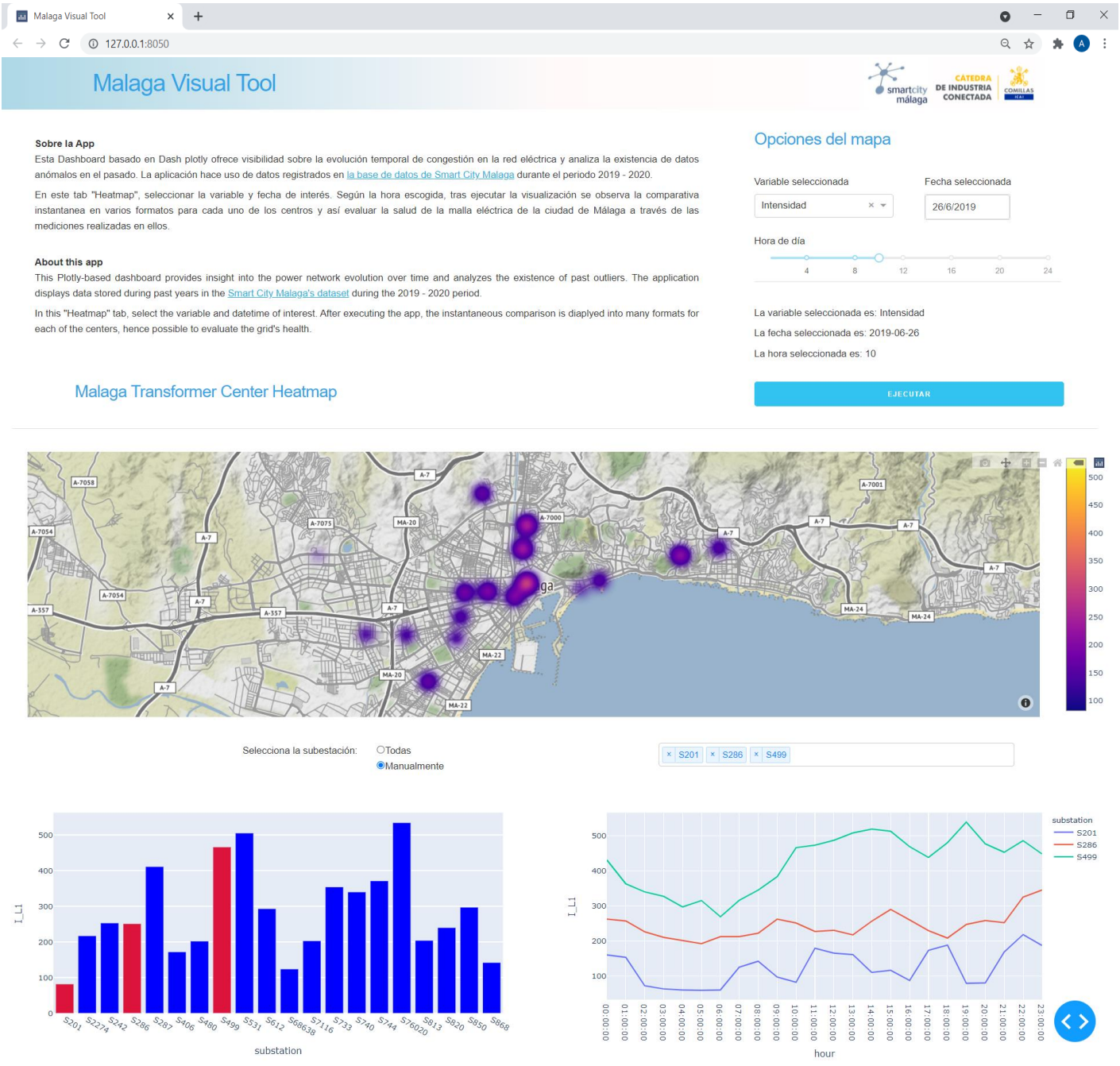
El mundo vive un proceso transitorio en aceleración hacia nuevos retos demográficos, las crisis energéticas, y los problemas de eficiencia que todo ello conlleva. Esto ha incrementado la preocupación por el modo en el que hacemos uso de dichos recursos a la hora de abastecer las necesidades de las sociedades modernas. Es por este motivo por el que el desarrollo sostenible ha pasado a un primer plano, marcando el ritmo con la que muchas empresas crecen y desarrollan sus proyectos. En concreto, este proyecto se alinea con los objetivos de desarrollo sostenible en los siguientes términos:

- 7 – Energía asequible y no contaminante: la sostenibilidad del sistema energético no será sostenible ni escalable si no viene acompañada de una modernización de la infraestructura a la altura de las exigencias. Una herramienta analítica extrae la profundidad de conocimiento necesaria para el control e intervención manual para un sistema tan complejo.

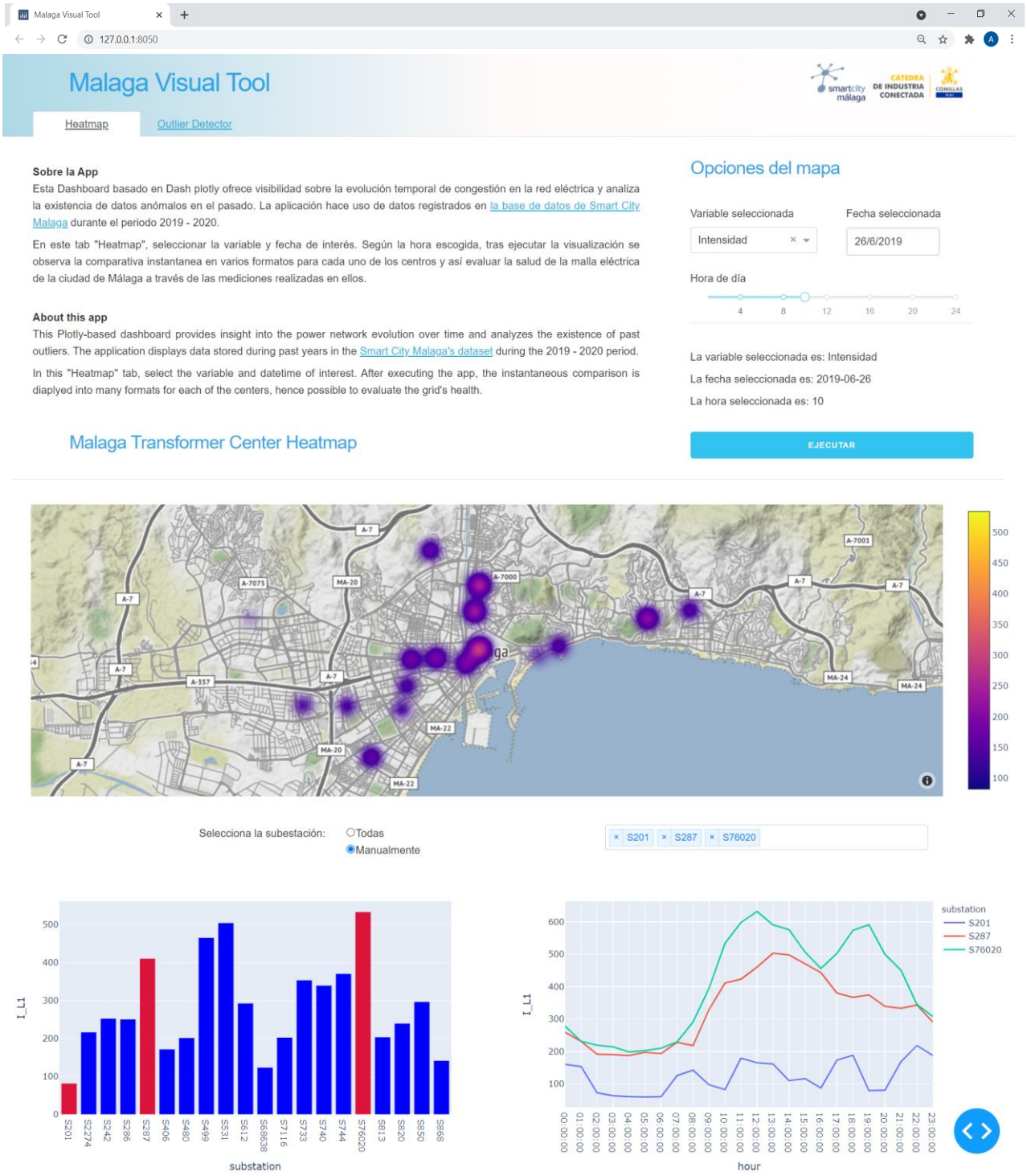
- 9 – Industria, Innovación e Infraestructuras: este proyecto fomenta la innovación y participa en la mejora de la capacidad tecnológica ofreciendo una solución para el mantenimiento de activos en diferentes entornos. La solución no es única de esta aplicación, sino que abre un abanico de posibilidades muy interesante para empresas industriales, y tecnológicas en general.

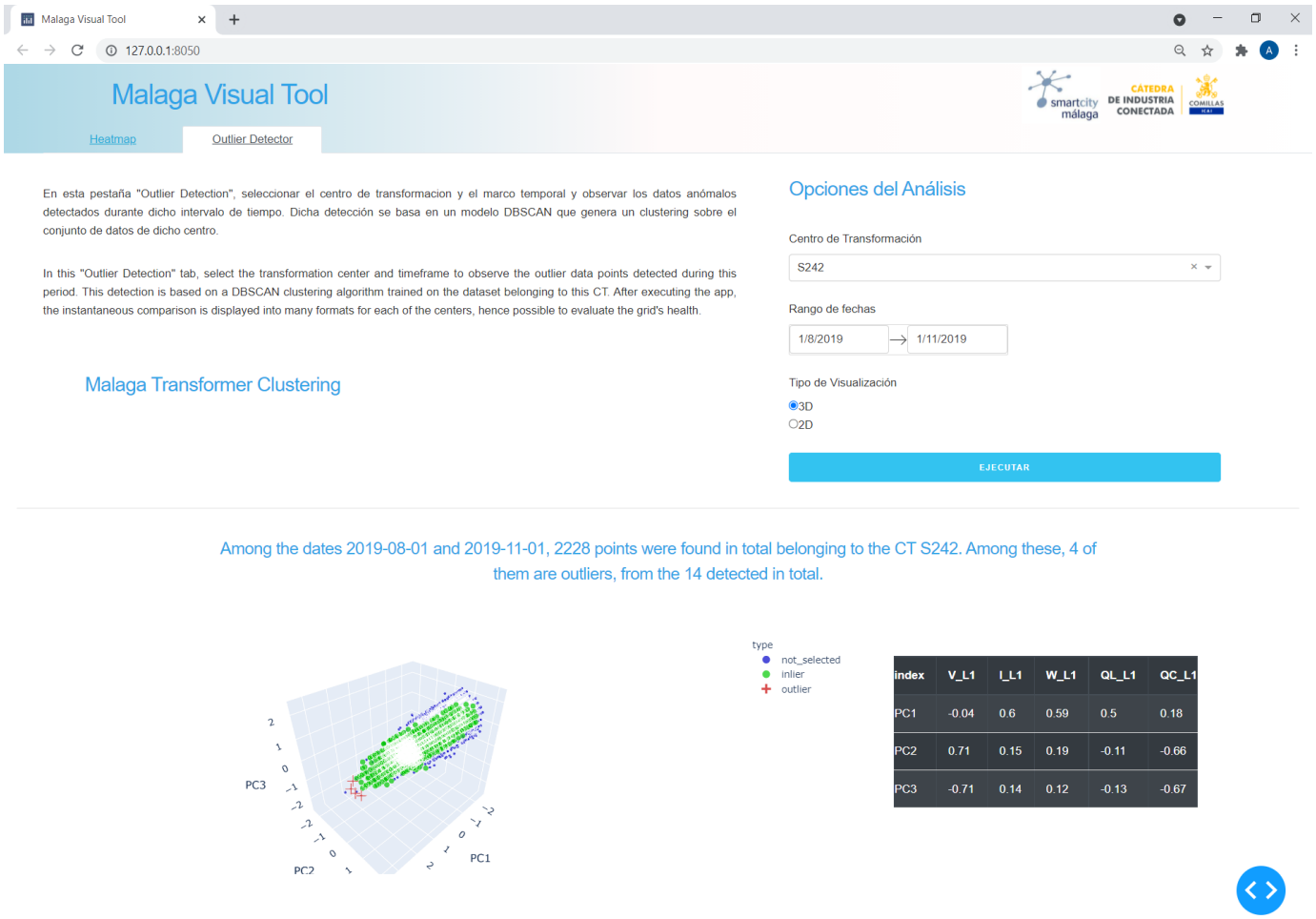
- 11 – Ciudades y comunidades sostenibles: La red eléctrica de Media y Baja Tensión son puntos particularmente críticos del sistema de distribución energética y colinda con el medio residencial, el cual se prevé que mute fuertemente en los próximos años. El cambio de paradigma obliga a evolucionar los medios no sólo en volumen, sino también en fondo y lógica de actuación.

ANEXO III. MALAGA VISUAL TOOL, VERSIÓN INICIAL



ANEXO IV. MALAGA VISUAL TOOL, VERSIÓN FINAL





index	V_L1	I_L1	W_L1	QL_L1	QC_L1
PC1	-0.04	0.6	0.59	0.5	0.18
PC2	0.71	0.15	0.19	-0.11	-0.66
PC3	-0.71	0.14	0.12	-0.13	-0.67

Figura 75: Pestaña Outlier Detection - Vista 1

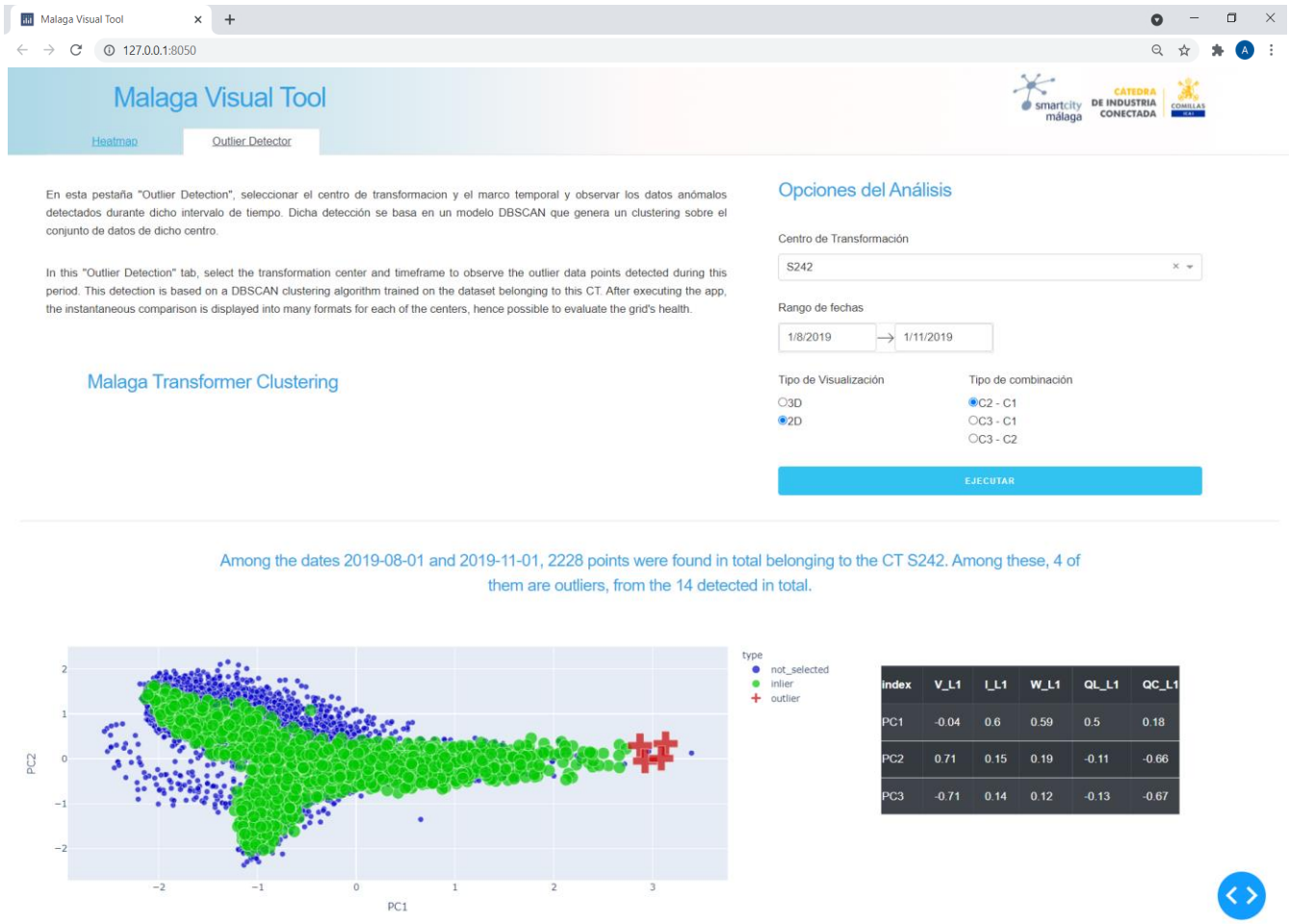


Figura 76: Pestaña Outlier Detection - Vista 2

ANEXO V. INSTRUCCIONES PARA EL DESPLIEGUE DE LA APLICACIÓN DASH

- Implementación en Heroku (desde VS Code)
  - Crear cuenta en Heroku: https://signup.heroku.com/
  - Crear un nombre de aplicación (este aparecerá en la URL)
  - Descargar e instalar Heroku CLI (este permite crear y gestionar las apps Heroku desde el terminal)
  - Crear una nueva carpeta en el directorio que se desee (donde la app y los archivos se localizarán)
    - Crear un nuevo entorno virtual
    - Seleccionar un interpretador de Python (versión 3.7.10 recomendada)
  - Copiar el archivo app_Malaga_Visual_Tool.py en la carpeta, así como el requirements.txt
  - Abrir un terminal y colocarse en la carpeta del proyecto (cd path) y activar el entorno virtual
  - Instalar las librerías con **pip install requirements.txt**
  - Crear un archivo llamado .gitignore. En este, se escribirían archivos innecesarios en el espacio Git.
  - Crear un archivo llamado **Procfile** dentro de la misma carpeta. Añadir en él la línea "**web: gunicorn web_Malaga_Visual_Tool: server**"
  - Dentro del terminal, escribir **heroku login**
  - Comprobar que git está instalado en el ordenador y escribir **git init**
  - **heroku git:remote -a NombreDeLaApp** (del paso 2)
  - **git add .**
  - **git commit -am "first commit"**
  - **git push heroku master**

- Implementación en PythonAnywhere (desde VS Code)
  - Crear una nueva carpeta en el directorio que se desee (donde la app y los archivos se localizarán)
    - Crear un nuevo entorno virtual
    - Seleccionar un interpretador de Python (versión 3.7.10 recomendada)
  - Copiar el archivo app_Malaga_Visual_Tool.py en la carpeta, así como el requirements.txt
  - Abrir un terminal y colocarse en la carpeta del proyecto (cd path) y activar el entorno virtual
  - Instalar las librerías con **pip install requirements.txt**
  - Llegado a este punto, lanzar el script para comprobar que la aplicación corre bien
  - Crear un repositorio Git local para preparar el Github
    - git init
    - git add .
    - git commit -m “first commit”
  - Crear el repositorio en la web de Github
  - Hacer push a este repositorio desde terminal. Para ello, ejecutar la serie de comandos indicados en la portada del repositorio vacío de la web en la consola de VS.
  - Transferir el repositorio a PythonAnywhere:
    - Crear una nueva cuenta: https://www.pythonanywhere.com/login/
    - Abrir una consola Bash desde la página web (Consoles > Bash)
    - **git clone [https del repositorio para clonar]**
  - Crear un entorno virtual en PythonAnywhere e instalar las librerías:
    - **mkvirtualenv [nombre_entorno_virtual] -p python3.7.10**
    - **cd [git repository name]**
    - **pip install -r requirements.txt**
  - Añadir la aplicación a PythonAnywhere
    - **Apps > New > Next > Flask > Python 3.7 > Next**
  - Una vez las librerías instaladas, ir a la pestaña Web y cambiar el “Source Code”
    - **home/[nombre_usuario]/[repositorio de git]**

- Bajo la sección de “Virtualenv”, escribir el nombre del entorno virtual
- Ir al “WSGI configuration file” y actualizar el archivo
    - Actualizar el final del camino del proyecto para coincidir con el nombre de la app
    - Cambiar las últimas líneas del código a

      from elections import app

      application = app.server
    - Guardar e archivo
- Volver a la pestaña “Web” y refrescar la página
- Ya es posible abrir la app

Extras:

- Las cuentas gratuitas sólo admiten un límite de memoria de 512 Mb. Si se borra la aplicación, el entorno virtual debe eliminarse antes de borrar los archivos:
    - **rm -rf /home/[nombre_usuario]/.virtualenvs/[nombre_entorno_virtual]**
    - Para comprobar el estado de la memoria: **du -hs /tmp ~/.[!.]* ~/* | sort -h**
- Si la aplicación continua sin abrirse, comprobar el “Error log”
- Activar el protocolo HTTPS y la contraseña en la pestaña “Web”