ESCUELA TÉCNICA SUPERIOR DE INGENERÍA (ICAI)

Máster en Big Data. Tecnología y Analítica Avanzada

ESTRATEGIA PARA LA CONCEPTUALIZACIÓN DE MODELOS DE IA EN EL CONTEXTO DE GESTIÓN DE LA ENERGÍA

Autor

Thomas Bustos

Dirigido por

Alejandro Macho Aroca

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

ESTRATEGIA PARA LA CONCEPTUALIZACIÓN DE MODELOS DE IA EN EL CONTEXTO DE GESTIÓN DE LA ENERGÍA

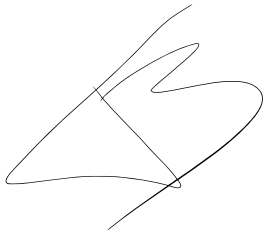
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2020/21 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.: Thomas Bustos Fecha: 09/ 07/ 2021



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Alejandro Macho Aroca Fecha: 09/ 07/ 2021

Vº Bº del Coordinador de Proyectos

Fdo.: Carlos Morrás Ruiz-Falcó Fecha: ...../ ...../ .....

Quisiera agradecer a mi tutor, Alejandro Macho Aroca, por su gran ayuda y apoyo a lo largo de este Trabajo de Fin de Máster.

Además, quiero dar las gracias los compañeros de trabajo de Iberdrola y a los profesores de la Escuela Pontificia Comillas ICAI por darme una oportunidad única de desarrollar mi formación.

RESUMEN

En este Trabajo Final de Máster se aborda el funcionamiento del mercado de la electricidad en España, diferentes técnicas de adquisición de datos, la implementación de diferentes modelos de predicción para predecir la evolución de la demanda eléctrica mensual en la industria, el uso de herramientas de autoML, métricas de optimizacion claves al momento de comparar modelos y herramientas de visualización de datos para comunicar resultados o predicciones dentro de la empresa.

El objetivo del trabajo es de entender si un modelo realizado con autoML, sin conocimientos específicos del mercado eléctrico y con datos públicos, puede resultar de utilidad. Para conseguir esto se han utilizado diferentes fuentes de datos, muchos dataset distintos con el objetivo de obtener un modelo de predicción que tenga valor para Iberdrola. Además, en el presente trabajo se pueden observar los límites del autoML y de este enfoque con datos públicos al momento de realizar modelos predictivos.

En el desarrollo del trabajo se explica el funcionamiento del mercado de la electricidad y nociones fundamentales de Machine Learning para contextualizar conceptos que serán utilizado durante la ejecución de este Trabajo Fin de Máster. En una segunda parte, se desarrolla la estrategia seguida para entrenar el modelo, desde la adquisición de los datos hasta la productivización de ese modelo mediante herramientas de visualización de datos, pasando por las diferentes métricas para su optimización utilizadas para comparar los diferentes modelos obtenidos. Para ilustrar esa parte mas teórica, se hará un desarrollo de prueba de concepto en el cual seguiremos todos los pasos anteriores para llegar a un modelo que coincide con los objetivos planteados en primer lugar.

Para el correcto desarrollo de estos modelos se han definido diferentes métricas y resultados objetivo para garantizar la utilidad de éstos. Dado que se utiliza autoML se han entrenado numerosos modelos distintos de forma rápida y en paralelo, lo cual permite enfocarse en reunir los datos que mejor pueden predecir la evolución de la demanda eléctrica en España.

Tras la obtención de un modelo satisfactorio, se pretende ilustrar su productivización con el desarrollo de un ejemplo de dashboard mediante la herramienta Tableau. El objetivo es hacer de la información extraíble del modelo datos útiles para la toma de decisiones del departamento de trading de Iberdrola al momento de planificar u operar en el mercado eléctrico.

Finalmente se presentan las lecciones aprendidas y conclusiones del trabajo sobre herramientas de autoML, sobre el enfoque de la construcción de modelos con datos públicos o de forma mas general, el potencial de la IA en las empresas.

ABSTRACT

This Master's thesis is about the electricity market, different data acquisition techniques, the implementation of different forecasting models to predict the evolution of monthly electricity demand, the use of autoML tools, key optimization metrics when comparing models, and data visualization tools to communicate results or predictions within the company.

The objective of this work is to understand if a model made with autoML, without specific knowledge of the electricity market and with public data can be useful. To achieve this, many data sources and many datasets are used with the approach of obtaining a prediction model that has value for Iberdrola and that matches the different goals of this work. In this paper, we can observe the limits of autoML as well as of public data when making predictive models.

In the development of the work, the functioning of the electricity market and fundamental notions of Machine Learning are explained to contextualize concepts that will be used later on. In a second part, we review the strategy followed to train a model, from data acquisition to data visualization, including the different optimization metrics used to compare models with each other. To illustrate this more theoretical part, a proof-of-concept development will be made in which we will follow all the previous steps to arrive at a model that matches the objectives set at the beginning.

For the correct development of these models, we will define what is considered to be a good model as well as the different inputs that have been used to arrive at the best possible model. Since autoML is used, many models can be trained in a short time, which allows focusing on gathering the data that can best predict the evolution of electricity demand in Spain.

After this, we will proceed to illustrate with the example of a Dashboard using the Tableau. It will show the predictions together with additional information that allows improving the decision making of the Trading department of Iberdrola at the moment of planning or operating in the electricity market. Finally, the lessons learned and conclusions of the work on autoML tools, modeling with public data or more generally, the potential of AI in companies will be determined.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

AEMET: Agencia Estatal de Meteorología

API: Application Programming Interfaces

AutoML: Automated Machine Learning

BI: Business Intelligence

FD: Forecast Distance

FDW: Feature Derivation Window

GEM: Global Energy Management

GWh: Gigavatio-hora

IA: Inteligencia Artificial

IRE: Índice Red Electrica

KPI: Key Performance Indicator

kWh: kilovatio-hora

MAPE: Mean absolute percentage error

MIBEL: Mercado Ibérico De La Electricidad

MIT: Massachusetts Institute of Technology

ML: Machine Learning

MLOps: Machine Learning Operations

MWh: Megavatio-hora

OMEL: Operador del Mercado Eléctrico

OTV: Out of Time Validation

PPA: Power Purchase Agreements

PVPC: Precio Voluntario para el Pequeño Consumidor

REE: Red Eléctrica España

RMSE: Root-mean-square error

Subasta CESUR: Contrato de Energía para Suministro de Último Recurso

SVM: Support Vector Machine

TSDB: Time Series Database

TUR: Tarifa de Último Recurso

URI: Uniform Resource Identifier

INTRODUCCION E OBJETIVOS

INTRODUCCIÓN

Hoy en día la energía se hace imprescindible en un mundo en el cual las tecnologías crecen cada vez más rápido. Para darse cuenta de este cambio enorme solo hace falta observar día a día las tecnologías que utilizamos y ponerlas en contraste con las que

utilizábamos en la década anterior. La energía es un bien de consumo de primera necesidad en la sociedad y existen muchas aplicaciones tanto como en los hogares, en la industria, los comercios y otros muchos ámbitos.

El grupo **Iberdrola** es hoy un líder energético global, el primer productor eólico y una de las mayores compañías eléctricas por capitalización bursátil del mundo. Se han adelantado dos décadas a la transición energética para combatir el cambio climático y ofrecer un modelo de negocio limpio, confiable e inteligente. El departamento de **Estrategia Digital** se encarga de la parte de gobierno y arquitectura en Gestión de la Energía Global (GEM, por sus siglas en inglés). El equipo del que depende "Estrategia Digital" es "Digital". que engloba la parte de estrategia, desarrollo y ciberseguridad. Sus principales actividades son:

- Realizar, estudiar y mejorar la estrategia tecnológica del departamento de gestión de la energía.
- Encargarse de la arquitectura tecnológica del área, estudiando las diferentes opciones y participando en las decisiones.
- Estudiar y aplicar las iniciativas en diferentes países dado que el equipo y la posición son globales dentro de la compañía.

El área de Gestión Global de la Energía (GEM), se encarga de gestionar los activos de generación y acudir al mercado para optimizar su operación, garantizando las mejores condiciones a los clientes. GEM es un equipo global dentro de Iberdrola, presente en diferentes países como España, Reino Unido, Brasil, México o Australia, entre otros. Es dentro de este departamento donde se ha realizado este Trabajo Fin de Máster.

OBJETIVOS

Los objetivos de este Trabajo de Fin de Máster, realizado durante las prácticas en Iberdrola y más concretamente en el área de Gestión Global de la energía, son:

1. Entender si un modelo realizado con autoML, sin conocimientos específicos del mercado eléctrico y con datos públicos puede resultar de utilidad
2. Conseguir modelos precisos con información pública y herramientas de autoML enfocados al mercado eléctrico con un MAPE inferior a 3%.
3. Entender el funcionamiento del departamento de Gestión de la energía.
4. Estudiar la productivización de estos modelos, especialmente a través de herramientas de BI.

METODOLOGÍA DEL TRABAJO

En una primera parte desarrollaremos el Estado del Arte para explicar el mercado de la electricidad o términos claves de ML. En segundo se tratará de explicar la estrategia para conceptualización del trabajo. Esa parte consiste en recordar los objetivos y desarrollar el proceso seguido para cumplirlos. En tercero será el ejemplo del desarrollo del mejor

modelo predictivo que se ha conseguido a lo largo del trabajo, siguiendo los pasos vistos en la parte estrategia para conceptualización. La siguiente **Ilustración 1** resume la metodología del trabajo que se pretende seguir, mostrando el tiempo aproximado que se empleara en cada tarea:

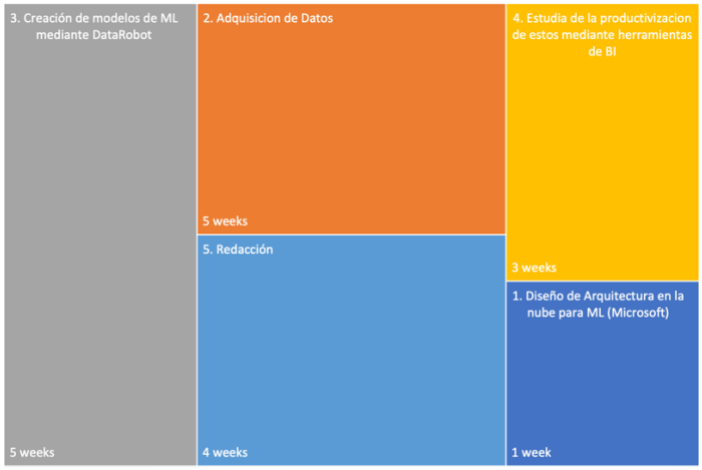


Ilustración 1: Metodología del trabajo

A lo largo del proyecto, se emplearán:

- 1 semana para el diseño de la Arquitectura en la nube para ML (Microsoft)
- 5 semanas de adquisición de datos contactando con ministerios y realizando scrapping
- 5 semanas de creación de modelos y análisis de resultados, utilizando DataRobot
- 3 semanas para estudiar la productivización y su aplicación a través de herramientas BI (visualización)
- 4 semanas para desarrollar este documento de tesis, los resultados obtenidos y sus conclusiones.

En la parte de Adquisición de los datos, se ha contactado con el Ministerios de Industria, Comercio y Turismo y el Ministerio para la Transición Ecológica y el Reto Demográfico, buscando y utilizando diferentes fuentes para encontrar datos relevantes para un buen modelo. Además, se ha utilizado Python para hacer “scrapping” mediante las librerías Beautiful Soup y Selenium y se ha utilizado la librería Pandas para crear los Dataset, limpiar los datos y juntarlos.

Para la construcción de los modelos se ha utilizado DataRobot. Gracias a esta herramienta es posible observar qué variables, indicadores económicos o indicadores relacionados con industrias especificas, tienen un mayor impacto en la demanda, creando así el modelo que mejor prediga la demanda a corto plazo en un primer lugar; y a largo plazo en un posteriormente.

En el estudio de la productivización, se utiliza Tableau o Power BI para ilustrar los resultados de los modelos, utilizando un dashboard para mostrar la información más importante de una forma sencilla y útil.

ESTADO DEL ARTE

MERCADO DE LA ELECTRICIDAD EN ESPAÑA

Para empezar, se quiere explicar el funcionamiento del mercado de la electricidad en España con el objetivo de plantear los conceptos claves al momento del desarrollo del trabajo realizado en Iberdrola. Primeramente, se desarrollará la estructura del mercado eléctrico en
España, cual ha sufrido muchos cambios desde 1997 cambiando hasta una liberalización del
mercado en cuento a las generaciones y las comercializadoras. A continuación, trataremos de explicar el sistema eléctrico y el Equilibrio entre generación de electricidad y demanda, cuales partes nos permitirá entender mejor el contexto en lo cual se ha realizado el trabajo y la importancia de poder predecir la demanda de la electricidad. Por otra parte, trataremos de cubrir los diferentes tipos de mercados y entender mejor a continuación la posición que ocupa Iberdrola en el Mercado de la electricidad en España.

Estructura del sector

Antes del año 1998, el sector eléctrico constituye un monopolio en todo España y se enfocaba sobre empresas con e importante crece vertical. Desde ese momento hasta entonces ese sector ha sufrido muchos cambios cuyos han resultado en la liberalización progresiva del sector.
El 27 de noviembre se aprobé la Ley 54/1997 que consiste en menos intervención publica al momento de gestionar el sistema, la creación del mercado para negociar la energía y terceros pueden acceder a las redes. Ley 24/2013 de 26 de diciembre es la que regula la estructura y el funcionamiento del mercado. Hace una distinción entre lo que es regulado y lo que no, aumente la competencia entre las comercializadoras para que el consumidor tiene una mejor información de lo que se propone tanto como para que pueda cambiar de suministrador de manera mas fácil.

“El suministro de energía eléctrica se define como la entrega de energía a través de las redes de transporte y distribución mediante contraprestación económica en las condiciones de regularidad y calidad que resulten exigibles.” [1]

La generación, el transporte, la distribución, la recarga energética, la comercialización, los intercambios intracomunitarios e internacionales y representan todas las actividades en relación con el suministro de la energía eléctrica. A continuación, se describen mas en detalle las mas importantes.

Generación: Consiste en la producción de energía eléctrica.

Transporte: Tiene por objeto la transmisión de energía eléctrica por la red de transporte, utilizada con el fin de suministrarla a los distintos sujetos y para la realización de intercambios internacionales.

La red de transporte de energía eléctrica está constituida por la red de transporte primario (instalaciones de tensión mayor o igual a 380 kV) y la red de transporte secundario (hasta 220 kV).

Distribución: Tiene por objeto la transmisión de energía eléctrica desde las redes de transporte, o en su caso desde otras redes de distribución o desde la generación conectada a la propia red de distribución, hasta los puntos de consumo u otras redes de distribución en las adecuadas condiciones de calidad con el fin último de suministrarla a los consumidores.

Comercialización: La actividad de comercialización será desarrollada por las empresas comercializadoras de energía eléctrica que, accediendo a las redes de transporte o distribución, tienen como función la venta de energía eléctrica a los consumidores y a otros sujetos según la normativa vigente.
[1]

Sistema eléctrico

Un sistema eléctrico es el conjunto de elementos que operan de forma coordinada en un determinado territorio para satisfacer la demanda de energía eléctrica de los consumidores. Se puede observar en la **Ilustración 2** los diferentes elementos que componen el sistema eléctrico, los cuales desarrollamos a continuación.

1) Los centros o plantas de generación donde se produce la electricidad (centrales nucleares, hidroeléctricas, de ciclo combinado, parques eólicos, etc.),
2) Las líneas de transporte de la energía eléctrica de alta tensión (AT),
3) Las estaciones transformadoras (subestaciones) que reducen la tensión o el voltaje de la línea (Alta tensión/Media tensión, Media tensión/Baja tensión),
4) Las líneas de distribución de media y baja tensión que llevan la electricidad hasta los puntos de consumo,
5) Un centro de control eléctrico desde el que se gestiona y opera el sistema de generación y transporte de energía.

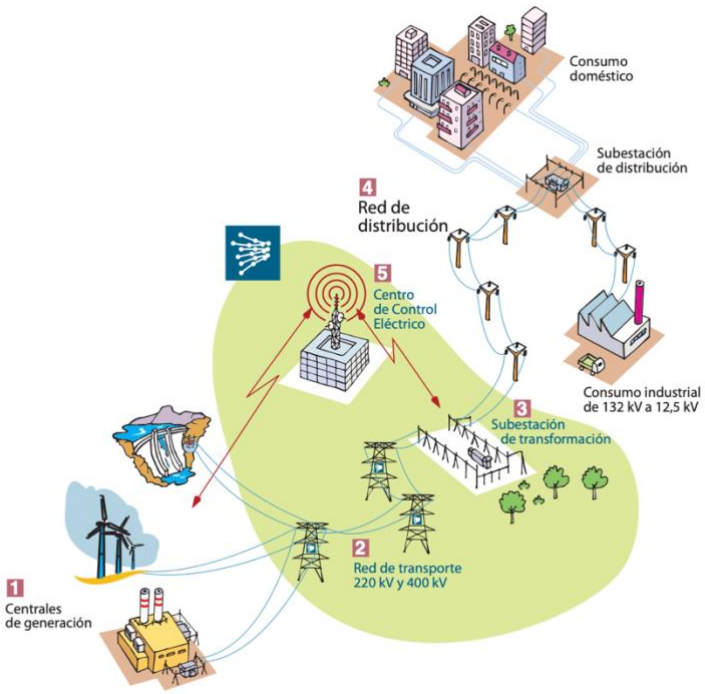


Ilustración 2: Esquema del Sistema Eléctrico Peninsular [2]

Equilibrio entre generación y demanda

Cuando utilizamos electricidad para cargar nuestro teléfono o encendemos la televisión, hay todo un sofisticado sistema que comienza en las centrales de producción, donde se genera la energía eléctrica. Luego se transporta esta energía transformada en alta tensión a través de las instalaciones eléctricas hasta los centros de distribución. Desde allí se transforma de nuevo la energía al nivel de tensión necesario según cada tipo de consumo (residencial, industrial o servicios) se realiza la distribución final a los consumidores

Posteriormente, esta energía transformada en alta tensión se transporta a través de las instalaciones eléctricas hasta los centros de distribución. Y desde allí, de nuevo transformada al nivel de tensión necesario para cada tipo de consumo (ya sea residencial, industrial o servicios) se realiza la distribución final a los consumidores.
Para que este proceso funcione y que la electricidad sigue llegando hasta nuestras casas al momento preciso en el que hacemos uso de ella, se tiene que operar el sistema en tiempo real, todos los días del año, las 24 horas del día, y mantener en constante equilibrio la generación y el consumo. Esto es debido a que la electricidad no se puede almacenar en grandes cantidades, por lo cual requiere una generación que corresponde a la cantidad precisa que se necesita en cada momento.

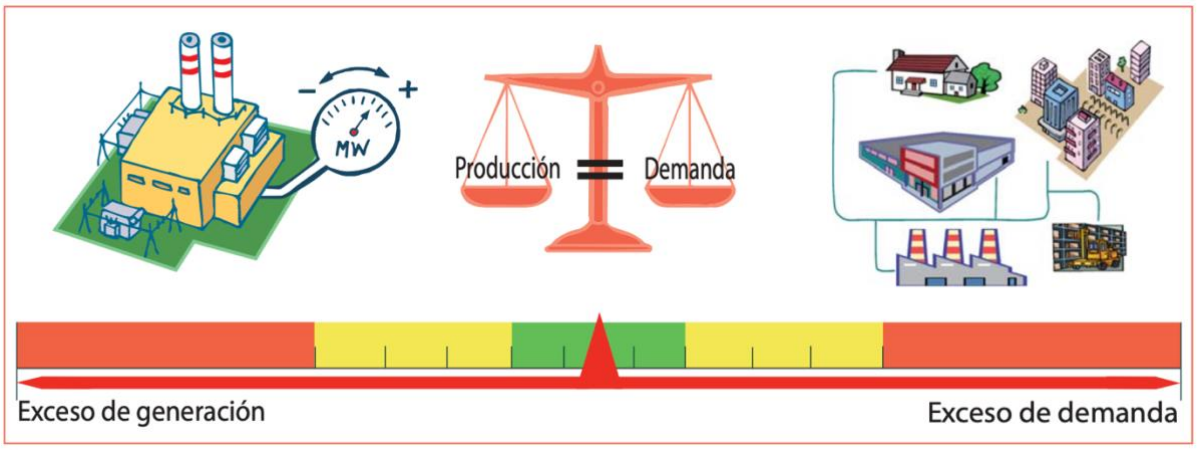


Ilustración 3. El Equilibrio Generación / Demanda [2]

El principal problema de la energía eléctrica reside en almacenarla. La energía eléctrica puede ser fácilmente generada, transportada y transformada. Sin embargo, hasta ahora no se ha logrado almacenarla de forma práctica, fácil y barata. Para acumular energía eléctrica debemos transformarla en otros tipos de energía y volver a transformarla en electricidad para su consumo (bombeo hidroeléctrico, aire comprimido o baterías son ejemplos de tecnologías que permiten transformar y almacenar energía de manera eficiente). En los últimos años, el sector de las energías renovables ha visto en las baterías de ion de litio una solución a su problema: el almacenamiento de la energía generada. No se ha convertido todavía en la principal tecnología de almacenamiento de renovables porque tiene un coste elevado.

Aunque el almacenamiento de energía sea posible, de momento, solo se aplica a ciertos tipos de generación de electricidad, sigue siendo complicado y caro y las centrales eléctricas (unidades de compra) no almacenan energía. Lo cual obliga a las “unidades

de venta" (empresas que venden energía eléctrica como Iberdrola) de comprar a "unidades de compra" sobre diferentes mercados (Apalancamiento, diario, intradiario). Por tanto, la que no es consumida en el momento se pierde. Los diferentes mercados de la electricidad se desarrollarán a lo largo del trabajo.

Tipos de Mercados

En ese apartado se quiere representar los tipos de mercados para entender mejor como se definen los precios y como se va ajustando el precio a lo largo del día para mantener ese equilibrio entre generación y demanda.

El mercado actual este compuesto de un mercado Mayorista y un mercado minorista. Con la liberalización progresiva de 1997 hasta 2009 para llegar al Mercado Ibérico de la Electricidad (MIBEL), se pretendía que decisiones con el antiguo marco correspondían al estado pasaran a ser gestionadas por mecanismos de libre mercado. Básicamente continua la regulación del estado en el transporte y mantenimiento de la rede y se liberalizan la generación y la comercialización. Esto significa que en la generación una empresa puede decidir a su cuenta y riesgo instalar una central que después será retribuida por mecanismos de mercado.

El mercado minorista permite a los consumidores elegir una entidad comercializadora de la electricidad. Las comercializadoras pueden proveer a sus clientes con la electricidad que han comprado a los generadores en el mercado mayorista.
En la comercialización los consumidores elegirán una comercializadora en el llamado mercado minorista y estas comercializadoras a su vez compraron la electricidad a los generadores en el mercado mayorista.

Para estos consumidores finales la estructura de costes de la energía eléctrica que consumen consta de varios componentes que forman parte de la cadena de valor de la electricidad. Podemos estudiar los costes a partir del tipo de mecanismo que fija las retribuciones asociadas: costes regulados frente a no regulados. Los costes regulados se establecen de manera administrativa y se aplican a negocios de carácter monopolístico como la transmisión, gestionada por Red Eléctrica de España, y la distribución, en manos de diversas distribuidoras como Iberdrola o Endesa. Además, también están regulados los costes asociados a incentivos que han sido decisión del estado como primas al régimen especial, costes de transición a la competencia, etc. Los costes regulados se corresponden con la componente de mercado mayorista. En estos mercados están organizados en régimen de libre competencia, en el que las

empresas comercializadoras y los consumidores directos pujan por la energía que requieren para sus clientes o para sus propias actividades respectivamente.
La **Ilustración 4** permite entender mejor como se relacionan el mercado Mayorista, Minorista así como la regulación de las actividades destinadas al suministro de energía eléctrica.

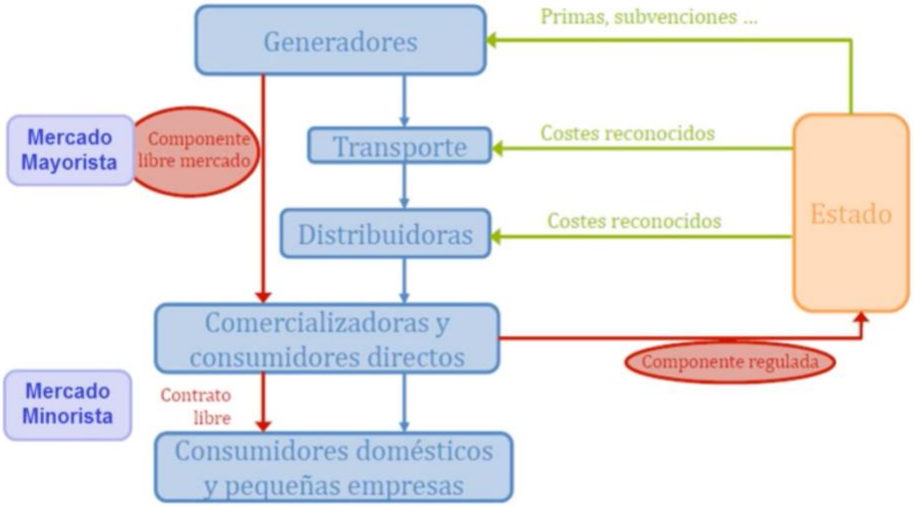


Ilustración 4. El Mercado Ibérico de la Electricidad [3]

Ese mercado mayorista en el que se subasta la electricidad se divide a su vez en mercados más pequeños que son gestionados por OMEL y Red Eléctrica España (REE).

OMEL: OMEL realiza la operación del mercado mayorista. Su labor es organizar subastas en las que, de facto, se decide la mayor parte de la energía eléctrica vendida en la península ibérica para cada hora del día. Organiza el mercado diario, también llamado *spot* y el intradiario.

REE: La labor de REE consiste en garantizar la calidad del suministro eléctrico en la red de alta tensión, atendiendo a estándares regulados, asegurando en todo momento la seguridad del suministro y su continuidad. Además gestiona los mercados de servicios que prestan los diferentes gestores de infraestructura conectada a la red, como generadores de electricidad, almacenamiento o compensación de la demanda. Estos son los mercados de restricciones técnicas, servicios complementarios y compensación de desvíos.

Para llegar al precio final de la electricidad para cada hora del día se hace a las 10:00 de la mañana para el día siguiente la subasta del mercado diario. En esa subasta los generadores y los consumidores, organizados por OMEL, alcanzan un precio de la electricidad para cada hora del día siguiente y cuánta energía se va a intercambiar. A continuación REE es responsable de comprobar el programa de generación y consumo resultante y factibilidad técnica. En caso de que el sistema no pueda soportar los flujos de potencia resultantes en líneas de alta tensión REE tiene la potestad para modificar los programas. REE acomete esta labor realizando contraofertas a los participantes cuyos programas quiere modificar, realizándose así de la manera con menor coste posible. La responsabilidad de REE como gestor de servicios complementarios comprometen a la entidad a asegurar las condiciones de fiabilidad y seguridad estándares y que resuelva problemas de desequilibrios de generación y consumo en

tiempo real sin perturbaciones en la frecuencia (50Hz) y manteniendo la potencia de la red en todo instante.
Existen tres servicios complementarios.

Servicio Complementario Primario: responde a desequilibrios pequeños y es obligatorio para todos los generadores por lo que no existe un mercado para gestionarlo.

Servicio Complementario secundario: permite al operador del sistema disponer de una reserva de capacidad muy flexible para resolver desequilibrios significativos entre la generación y la demanda.

Servicio Complementario Terciario: tiene por objetivo que en caso de que se haga uso de esa banda secundaria pueda restituirse la reserva que se esta utilizando. Este ultimó se celebraría a ultima día hora del día de hoy. La reserva se activa de forma manual modificando la potencia de las centrales que hubieran acertado al menor precio en el caso de tener que generar energía o al mayor precio de recompra en el caso de tener que consumir energía.

En los mercados de reserva tienen un rol crucial las centrales gestionables. Si se requiere mayor potencia de generación, una central de ciclo combinado, por ejemplo, puede incrementar la inyección de potencia a la red rápidamente, gracias a las capacidades físicas de esta tecnología. Si fuera necesario mayor consumo para compensar un exceso de suministro de potencia a la red, el bombeo de una gran central hidroeléctrica puede equilibrar el excedente.
Después tendríamos el mercado **intradiario** que permite a los agentes que han participado en el mercado diario gestionar posiciones anteriores convirtiéndose en un mecanismo eficaz para solventar incidencias y modificaciones en las previsiones tanto de oferta como de demanda. Se realizarían seis sesiones que se distribuirían a lo largo del día de hoy y mañana. En cada una de estas se puede ir ajustando la oferta y demanda de todas las horas que pueda quedar por suministrar a lo largo del día.
Finalmente tenemos los desvíos que son para resolver desequilibrios entre la oferta y la demanda que puedan identificarse unas horas antes del despacho tras la celebración de cada mercado intradiario. Consisten pedir ofertas a los generadores en el sentido opuesto a los desvíos que están previstos en el sistema.

Precios
Dentro de este orden de mercados, en el mercado diario/spot cuando se define entre generadores y consumidores la energía que se producirá y consumirá, los generadores ofertan una energía eléctrica que pueden y están dispuestos a generar a un precio, que como mínimo, sea el que propone su oferta. Atendiendo a sus costes marginales y de oportunidad de generación de electricidad, cada generador, ofertará un precio diferente. El mecanismo de elaboración de la oferta se explica más adelante.

Las ofertas de compra las realizan entidades comercializadoras con el objetivo de abastecer a sus clientes. La demanda de electricidad es muy inelástica, es decir, es mayor el coste de no aprovisionar a los clientes con electricidad que pagar un alto precio por esta y asegurar que la generación cubrirá su demanda. Las ofertas se realizarán al precio máximo permitido, que es de 0,18 €/kWh. Las centrales de bombeo e industrias programables harán ofertas de compra a un precio inferior. Estos actores del mercado pueden afrontar el hecho de no comprar si el precio no es económicamente viable para

sus actividades. Así, la demanda se satisface hasta que se alcanza el precio por encima del cual no hay mas compradores dispuestos a pagar. Este es el llamado **precio de casación**, el cual podemos observar en la **Ilustración 5**. Es importante indicar que, si bien las unidades productoras ofertan a un precio menor que el precio de casación y las ofertas de compras se han realizado a un precio mayor, toda la energía que se ha vendido para esa hora se retribuye al mismo precio obtenido por el cruce de las curvas de oferta y demanda. Es lo que se denomina mercado marginalista. El precio marginal es el que otorga el valor de toda la energía vendida.

Ilustración 5.Ejemplo de casación de oferta y demanda en una hora concreta. [4]

¿De qué depende el precio al que los generadores subastan su energía?

El precio de oferta de la energía no refleja necesariamente los costes variables de generación, o al menos no únicamente. Estos costes engloban la operación y el mantenimiento de la central, además de combustibles y otros fungibles cuyos costes son proporcionales a la generación de electricidad. Esto significa que a los costes variables es necesario agregarles el coste de oportunidad que puede suponer generar electricidad en un momento y no en otro, especialmente en el caso de centrales hidroeléctricas y tecnologías que almacenen energía. Para una central hidroeléctrica los costes variables asociados a turbinar el agua que se encuentra almacenada en el embalse son ínfimos. Sin embargo, el coste de oportunidad de hacerlo en un momento y no en otro futuro a un precio superior tiene que repercutir el valor del agua almacenada. Por otra parte, en una época de lluvias intensas que hagan que el nivel del embalse se aproxime al desborde o incluso llegue a esa situación el valor del agua sería cero. Por lo tanto, realizará ofertas a precio muy bajo o incluso cero para asegurarse de que entra en la casación. En el caso de centrales térmicas de carbón o gas, si el generador puede revender dicho combustible a un tercero, entonces consumir dicho combustible tiene un coste de oportunidad que deberá sumar en su oferta al coste variable por la propia generación de la electricidad. Es decir que ese coste de oportunidad no es el precio al que se adquirió el combustible sino el precio al que puede revenderlo. En el caso de un parque eólico o de una central hidráulica de afluente se transforma un recurso natural sin costes variables en electricidad. Por lo tanto, si tienen una ocasión de generar una situación de viento favorable o de agua en el rio el no hacerlo no aumentara la posibilidad de obtener mayores beneficios en el futuro. Ya que ni ahora el combustible ni puede almacenarlo para otro momento. Por esa razón ofertaran a precio cero para asegurarse de poder entrar en la casación. El resto de los generadores de régimen especial por conllevar ventajas significativas para el país tienen el privilegio de tener asegurado que lo que vayan a producir se consuma por lo que podríamos

considerar que también ofertan a precio cero. Después dependiendo de cada tecnología y cada caso concreto, los generadores del régimen especial reciben una prima independientemente del precio de casación o una prima variable que se suma al precio que se obtiene a la casación. Y finalmente tenemos las centrales nucleares que también ofertan a precio cero. Pero en este caso la razón es diferente. Las centrales nucleares tienen poca capacidad de variar su nivel de producción en el tiempo y son centrales consideradas de base es decir que están todo el tiempo produciendo a su potencia nominal. Por tanto, las ofertas a precio cero buscan asegurar la casación para mantener el nivel de producción constante. Dejando que el precio que recibirán como retribución lo marquen el resto de las tecnologías que ofertan a otros precios mayores por otros motivos.

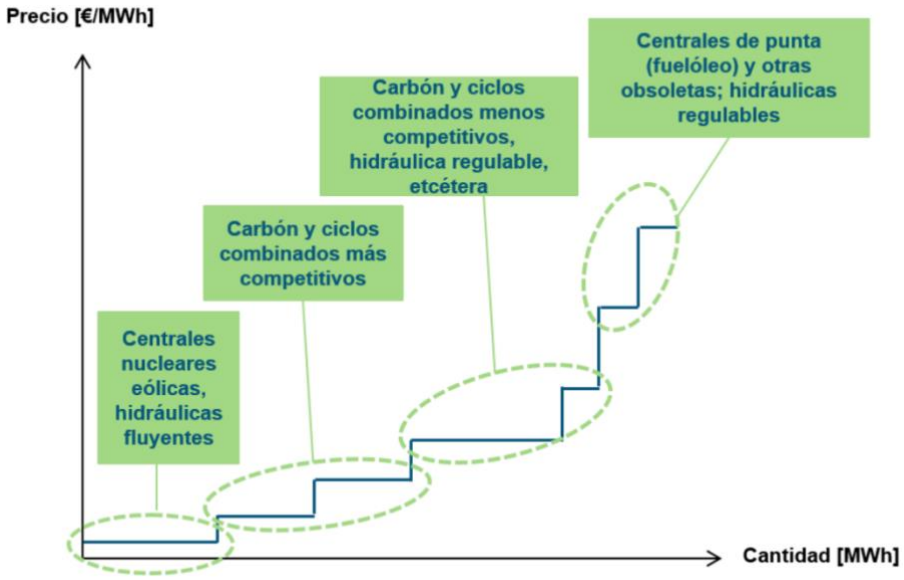


Ilustración 6. La curva de oferta de electricidad del mercado [5]

Por lo tanto, tal y como lo hemos visto del cruce de las curvas de oferta y demanda saldría al precio de casación inicial. Sin embargo, estas curvas iniciales hay que corregirlas acorde a las condiciones complejas. Estas condiciones son las que los generadores imponen cuando hacen sus ofertas sobre varias horas a la vez. El ejemplo mas típico es la oferta de ingresos mínimos por la cual se fija un nivel mínimo de ganancias en un día una vez que se casan las veinticuatro curvas de un día si un agente ha declarado una condición de ingresos mínimos que no se cumple, sus tramos de ofertas se retiran. Así es como se obtiene la curva modificada que es la que se emplea para fijar finalmente el precio de casación de la correspondiente hora y este es el precio que se pagara a todos los generadores. Normalmente este precio lo marca las centrales térmicas de carbón o gas en algunas puntas de consumo el precio lo marcan las generadoras mas caras como las centrales de fuel o las grande hidráulicas y algunas horas al ano cuando la demanda se puede cubrir mediante los generadores ofertan a precio cero, la electricidad se vende a grandes. Por lo tanto, el precio de la electricidad varia cada hora del día dependiendo de múltiples factores como la demanda para esa hora, el precio de los combustibles fósiles, las condiciones climatológicas como la cantidad de viento sol y lluvia que tengamos, las restricciones técnicas de la red etc.

¿De donde sale el precio fijo que pagamos por kilovatio/hora en nuestras tarifas?
Hasta el 1 de julio de 2009 existió la tarifa integral por la que el estado fijaba el precio de la electricidad para canal incluyendo tanto la componente regulada como la de libre mercado. Pero también como hemos visto después del año 1997 hemos tenido un periodo de adaptación para abandonar nuestras tarifas estables por la administración en la línea del marco legal estable y la componente de energía de la tarifa se debía establecer mediante el mecanismo de libre mercado. Por lo que desaparecieron las viejas tarifas y todos los consumidores debíamos pasar al mercado minorista contratando una comercializadora. Sin embargo, para aquellos consumidores que en esta fecha de 1 de julio 2009 no habían contratado una comercializadora quedan casi el 90% de los consumidores se diseño lo que se conoce como tarifa de ultimó recurso a la que pasaban automáticamente. De la misma forma, su comercializadora pasaba a ser la comercializadora de ultimó recurso que había creado su distribuidora para dar servicio en este nuevo mercado.

¿Y como se fija esa tarifa de ultimo recurso?
EL PVPC (precio voluntario para el pequeño consumidor) o antiguamente llamada TUR (tarifa de ultimo recurso), al igual que la antigua tarifa integral también tiene una componente regulada fija por el estado y otra componente de libre mercado que se fija de acuerdo con el precio alcanzado en la llamada subastas de CESUR. Estas subastas las organiza OMEL, el operador del mercado eléctrico, cada tres meses. Y en ellas las grandes compañías eléctricas, bancos y fundos de inversión empujan por la electricidad que se van a consumir en los próximos meses, forma que se comprometen a comprar la electricidad que se va a vender en la subasta diaria y apagar los generadores según el precio de casación diaria. Sin embargo, venderán esa electricidad a las comercializadoras a un precio fijo. Por lo tanto, el que se comprometa a vender la electricidad a menor precio será el que se lleve la subasta. De no asumir el suficiente riesgo pueden no resultar ganadoras de la subasta, pero si asumen demasiado estarán expuestos a perdidas económicas.
Una vez realizada la subasta CESUR, la tarifa de ultimo recurso se obtiene añadiendo la componente regulada y un margen de beneficios estipulado desde la administración a las comercializadoras de ultimo recurso.

SITUACIÓN DE GRANDES EMPRESAS ELÉCTRICAS EN IA

El objetivo de ese tema es de definir conceptos tal como Inteligencia Artificial (IA), Machine Learning (Aprendizaje Automático) y Deep Learning (Aprendizaje Profundo) para a continuación observar las aplicaciones de esas tecnologías en el sector energético en general y luego desarrollar ejemplos de implementaciones de esas tecnologías en las empresas energéticas de renovables enfocándose en Iberdrola, empresa por la cual se ha realizado la prueba de desarrollo que veremos posteriormente.

Inteligencia Artificial, Machine Learning y Deep Learning

En los últimos años, la Inteligencia Artificial (IA) ha ganado relevancia en una gran variedad de sectores. El uso de esas tecnologías sigue muy reciente y queda mucho por hacer, pero se profundizará posteriormente. Antes de todo, vamos a definir cada tecnología empezando por la Inteligencia Artificial. John McCarthy, un prominente

informático que recibió el premio Turing en 1971 ha introducido el termino *Inteligencia Artificial* o "cloud computing" al cual le asigna la siguiente definición:
"La Inteligencia Artificial es la ciencia y la ingeniería que consiste en crear máquinas inteligentes, especialmente programas informáticos inteligentes. Está relacionada con la tarea similar de utilizar ordenadores para comprender la inteligencia humana, pero la IA no tiene por qué limitarse a métodos biológicamente observables." [6]
Para que la definición sea completa, define también el termino *Inteligencia*:
"La inteligencia es la parte computacional de la capacidad de alcanzar objetivos en el mundo. Las personas, muchos animales y algunas máquinas presentan distintos tipos y grados de inteligencia." [6]
Como se puede observar el la **Ilustración 7**, Inteligencia Artificial incluye el Aprendizaje Automático (Machine Learning) y el Aprendizaje Profundo (Deep Learning).

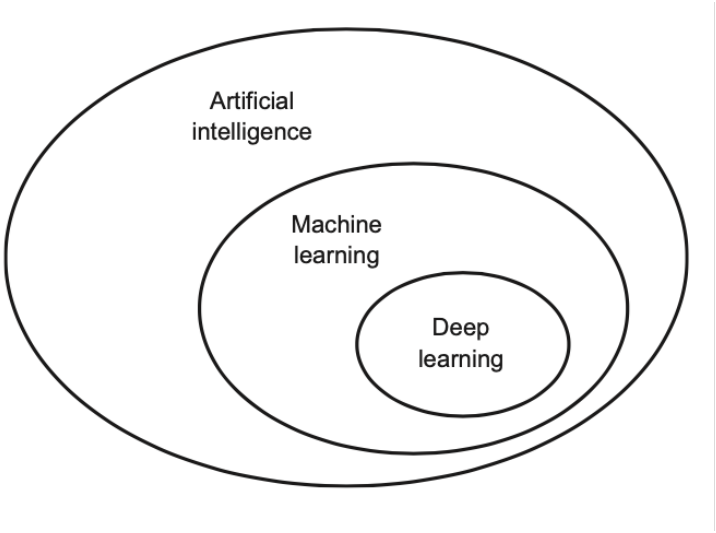


Ilustración 7. Inteligencia Artificial, Aprendizaje Automático, Aprendizaje Profundo [7]

Aprendizaje Automático
El aprendizaje automático surge de esta pregunta: ¿podría un ordenador ir más allá de "lo que sabemos ordenar que haga" y aprender por sí mismo cómo realizar una tarea determinada? ¿Podría un ordenador sorprendernos? En lugar de que los programadores elaboren reglas de procesamiento de datos a mano, ¿podría un ordenador aprender automáticamente estas reglas observando los datos?
Esta pregunta abre la puerta a un nuevo paradigma de programación. En la programación clásica, el paradigma de la IA simbólica, los humanos introducen reglas (un programa) y datos que deben procesarse de acuerdo con esas reglas, y de ahí salen las respuestas (véase la **Ilustración 8**). Con el aprendizaje automático, los humanos introducen datos y las respuestas que se esperan de los datos, y de ahí salen las reglas. Estas reglas pueden aplicarse a nuevos datos para producir respuestas originales.

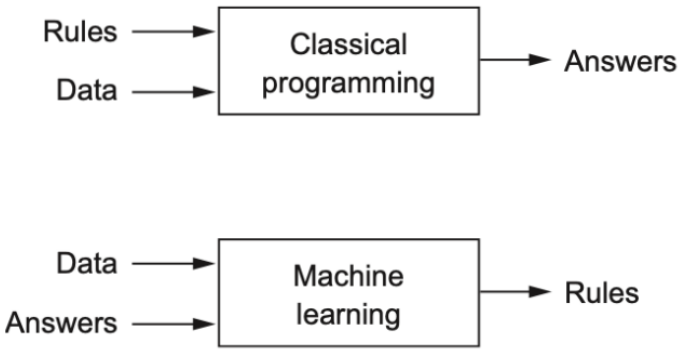


Ilustración 8. Aprendizaje automático: un nuevo paradigma de programación. [7]

Un sistema de aprendizaje automático se entrena en lugar de programarse explícitamente. Se le presentan muchos ejemplos relacionados con una tarea, y encuentra una estructura estadística en estos ejemplos que finalmente permite al sistema crear reglas para automatizar la tarea. Por ejemplo, si se quiere automatizar la tarea de etiquetar las fotos de vacaciones, se puede presentar a un sistema de aprendizaje automático muchos ejemplos de fotos ya etiquetadas por humanos, y el sistema aprendería reglas estadísticas para asociar fotos específicas a etiquetas específicas.
Aunque el aprendizaje automático no empezó a florecer hasta la década de 1990, se ha convertido rápidamente en el subcampo más popular y exitoso de la IA, una tendencia impulsada por la disponibilidad de hardware más rápido y conjuntos de datos más grandes. El aprendizaje automático está estrechamente relacionado con la estadística matemática, pero se diferencia de ésta en varios aspectos importantes. A diferencia de la estadística, el aprendizaje automático tiende a tratar con conjuntos de datos grandes y complejos (como un conjunto de datos de millones de imágenes, cada una de ellas compuesta por decenas de miles de píxeles) para los que el análisis estadístico clásico, como el análisis bayesiano, sería poco práctico. Como resultado, el aprendizaje automático, y especialmente el aprendizaje profundo, presenta comparativamente poca teoría matemática -quizás demasiado poca- y está orientado a la ingeniería. Es una disciplina práctica en la que las ideas se demuestran empíricamente con más frecuencia que teóricamente.

Aprendizaje Profundo
El aprendizaje profundo es un subcampo específico del aprendizaje automático: una nueva forma de aprender representaciones a partir de datos que hace un énfasis en el aprendizaje de capas sucesivas de representaciones cada vez más significativas. Lo profundo en el aprendizaje profundo no es una referencia a ningún tipo de comprensión más profunda lograda por el enfoque; más bien, representa esta idea de capas sucesivas de representaciones. El número de capas que contribuyen a un modelo de datos se denomina profundidad del modelo. Otros nombres apropiados para el campo podrían haber sido aprendizaje de representaciones en capas y aprendizaje de representaciones jerárquicas. El aprendizaje profundo moderno a menudo implica decenas o incluso cientos de capas sucesivas de representaciones, y todas ellas se aprenden automáticamente a partir de la exposición a los datos de entrenamiento. Mientras tanto, otros enfoques del aprendizaje automático tienden a centrarse en el aprendizaje de sólo una o dos capas de representaciones de los datos; de ahí que a veces se les llame aprendizaje superficial.
Como ejemplos de aplicaciones de inteligencia artificial existen entre varios:

- Reconocimiento de voz: Es una capacidad que utiliza el Natural Language Processing (NLP) para procesar el habla humana en un formato escrito. Hoy en día, muchos dispositivos móviles incorporan el reconocimiento del habla en sus sistemas para realizar búsquedas por voz (Siri) o proporcionar más accesibilidad en cuento a los mensajes de texto.

- Servicio de atención al cliente: Los "chatbots" online están sustituyendo a los agentes humanos a lo largo del recorrido del cliente. Responden a las preguntas más frecuentes sobre temas como el envío de pedidos, o proporcionan asesoramiento personalizado, vendiendo productos cruzados o sugiriendo tallas para los usuarios, cambiando la forma en que pensamos sobre la participación de los clientes en los sitios web y las plataformas de medios sociales. Algunos ejemplos son los bots de mensajería en los sitios de comercio electrónico con agentes virtuales, las aplicaciones de mensajería, como Slack y Facebook Messenger, y las tareas que suelen realizar los asistentes virtuales y de voz.

- "Computer Vision": Esta tecnología de IA permite a los ordenadores y sistemas derivar información significativa de imágenes digitales, vídeos y otras entradas visuales, y basándose en esas entradas, puede actuar. Esta capacidad de ofrecer recomendaciones la distingue de las tareas de reconocimiento de imágenes. Gracias a las redes neuronales convolucionales, la visión por ordenador tiene aplicaciones en el etiquetado de fotos en las redes sociales, en la obtención de imágenes radiológicas en la sanidad y en los coches auto conducidos en la industria del automóvil.

- Motores de recomendación: A partir de los datos de consumo anteriores, los algoritmos de IA pueden ayudar a descubrir tendencias de datos que pueden utilizarse para desarrollar estrategias de venta cruzada con mejor eficiencia. Esto se utiliza para hacer recomendaciones de complementos relevantes a los clientes durante el proceso de compra para los minoristas en línea. Es el mismo proceso cuando Netflix por ejemplo recomienda películas basadas en los perfiles de sus clientes.

- Trading Automatizado: Diseñadas para optimizar las carteras de valores, las plataformas de negociación de alta frecuencia impulsadas por la IA realizan miles o incluso millones de operaciones al día sin intervención humana.

IA en el sector de la Energía Eléctrica

La Inteligencia Artificial es cada vez más importante en la industria energética y tiene un gran potencial para el diseño futuro del sistema energético. En el contexto de la industria energética, la IA tiene muchas aplicaciones que permite tener ventajas competitivas, reducir costes o mejorar procesos. Como ejemplos de la utilización de la Inteligencia Artificial en el sector eléctrico tenemos:

Trading de electricidad: La electricidad tanto como el oro es un producto que se puede comprar, vender y intercambiar en mercados. Para hacer cualquier operación en ese mercado hay que entender el mercado y las fluctuaciones del precio de la mejor mañera

posible. Al entender mejor el mercado y acertar las predicciones de demanda utilizando datos, permite tener una ventaja competitiva muy importante en el mercado.
Consumo inteligente de energía: La energía hace un paso mas en la era digital. Según un informe sobre digitalización y economía publicado en 2017 por la IEA [8], en 2040 habrá 11 billones de dispositivos inteligentes o "Smart devices" en mas de 1 billón de hogares. Estos dispositivos conectados permiten ayudar a los consumidores a controlar y gestionar su consumo de electricidad. La combinación de contadores inteligentes con una tecnología inteligente más amplia podría ahorrar a los edificios, por sí solos, alrededor del 10% de su consumo total de energía. El coste medio de un contador inteligente se ha reducido en aproximadamente una cuarta parte en la última década, con casi 600 millones de estos dispositivos desplegados en todo el mundo.

Almacenamiento de Energía Inteligente: La IA puede mejorar la tecnología de almacenamiento facilitando la integración de tecnologías como micro redes de energía renovable, el almacenamiento en baterías a escala de servicios públicos o Central hidroeléctrica reversible. Como se anoto anteriormente en el apartado
Equilibrio entre generación y demanda, hasta ahora no se ha logrado almacenarla electricidad de forma práctica, fácil y barata. No obstante, la presión sobre los agentes de la energía para equilibrar la producción y la demanda esta creciendo mucho por las redes modernas creciendo rápidamente junto a la proliferación de fuentes de energía intermitentes como la eólica y la solar. A medida que la tecnología mejora y los costes se reducen, el almacenamiento inteligente de energía está desempeñando un papel más importante en los servicios auxiliares de la red, funciones que ayudan a los operadores de la red a equilibrar y apoyar la transmisión de energía de los generadores a los consumidores.
En tiempos en los que se produce un desajuste del equilibrio oferta/demanda, la IA puede permitir una asignación más eficiente y, de este modo, ahorrar energía que, de otro modo, se desperdiciaría para un uso posterior. La integración de varios sistemas de almacenamiento distintos no sólo maximiza la rentabilidad, sino que los sistemas inteligentes de almacenamiento de energía aumentan aún más la seguridad al mejorar el control de la frecuencia y el voltaje causados por la generación intermitente de energía.

Robots: En el sector energético, una aplicación de la IA seria la creación de robots autónomos para que inspeccionen líneas de alta tensión en tierra o patrullar el fondo marino en buscas de recursos valiosos, comunicando su ubicación para una futura extracción de lo que han encontrado. En esos casos la IA esta aplicada a sustituir a los humanos en tareas que serian peligras y de la misma forma no arriesgar mas vidas humanas.
El proyecto de robot que explora el fondo marino se esta desarrollando por la colaboración entre la empresa ExxonMobil (empresa petrolea de estados unidos) y el MIT. [9]

Resulta claro que, por los ejemplos visto anteriormente, la IA tiene una multitud de aplicaciones tanto en el sector energético en general como en el sector eléctrico. Además, en el futuro tendremos aun mas datos, mejores informes y tecnologías mas avanzadas con costes reducidos. Esa evolución permitirá nuevas aplicaciones de IA que no se pueden implementar hoy en día. En el mensaje de Bill Gates para los estudiantes graduándose de 2017 menciona que si empezara ahora a buscar oportunidades para

tener un impacto en el mundo, primero se enfocaría en el sector de IA y en segundo sobre el sector energético.

"*If I were starting out today and looking for the same kind of opportunity to make a big impact in the world, I would consider three fields. One is artificial intelligence. We have only begun to tap into all the ways it will make people's lives more productive and creative. The second is energy, because making it clean, affordable and reliable will be essential for fighting poverty and climate change.*" [10]

IA en una empresa de energías renovables: Iberdrola

El grupo Iberdrola es hoy un líder energético global, el primer productor eólico y una de las mayores compañías eléctricas por capitalización bursátil del mundo. Su sede esta en Bilbao (España) y tiene 28836 empleados. Se han adelantado dos décadas a la transición energética para combatir el cambio climático y ofrecer un modelo de negocio limpio, confiable e inteligente. Con respeto a las diferentes actividades destinadas al suministro de energía eléctrica visto retrospectivamente en **Estructura del sector**, Iberdrola es generadora, distribuidora y comercializadora. La capacidad instalada de la compañía supera ya los 55.000 MW y tiene instalaciones en Estados Unidos, México, Brasil, España, Reino Unido, Alemania, Francia, Portugal, Hungría, Rumania, Grecia, Chipre y Australia. En la Ilustración **9** podemos observar la capacidad por países en MW.

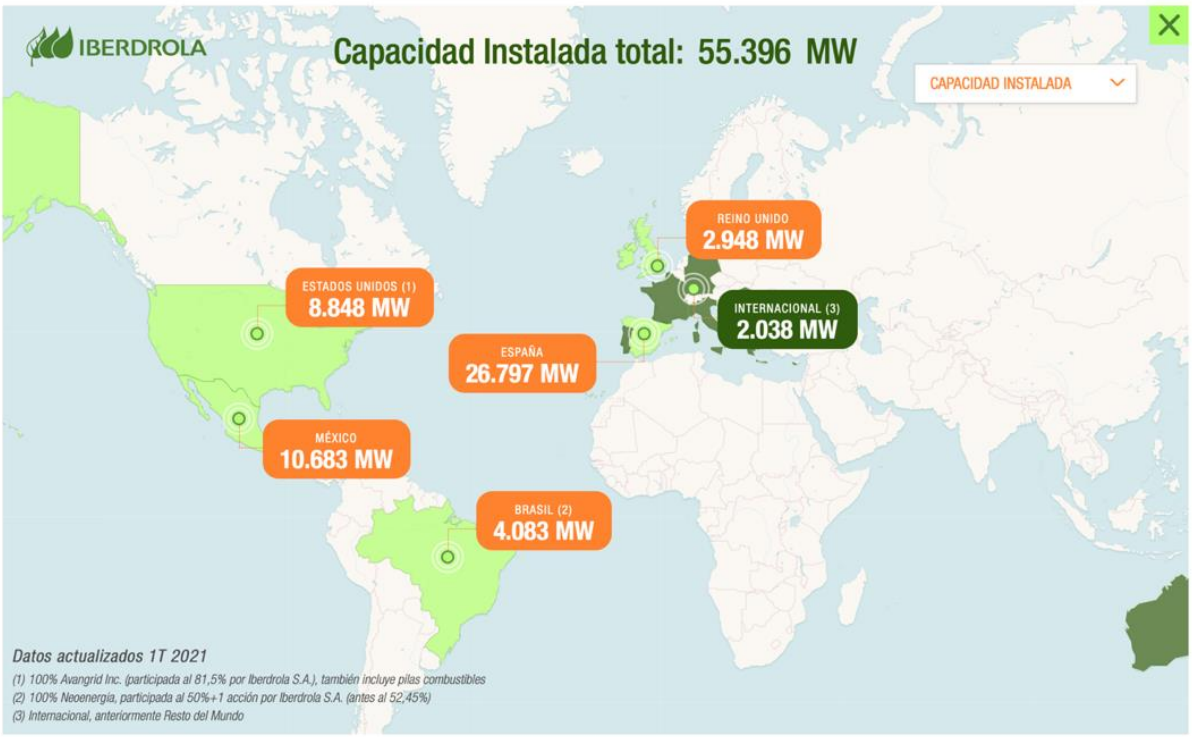


Ilustración 9. Capacidad Instalada de Iberdrola a cierre del Primer trimestre 2021 [11]

Este trabajo se ha realizado dentro del departamento de **Estrategia Digital**, cual se encarga de la parte de gobierno y arquitectura en Gestión de la Energía Global (GEM, por sus siglas en inglés). El equipo del que depende "Estrategia Digital" es "Digital" que engloba la parte de estrategia, desarrollo y ciberseguridad. Sus principales actividades son:

- Realizar, estudiar y mejorar la estrategia tecnológica del departamento de gestión de la energía.
- Encargarse de la arquitectura tecnológica del área, estudiando las diferentes opciones y participando en las decisiones.
- Carácter Global: el equipo y la posición son globales dentro de la compañía, estando dentro de sus responsabilidades el estudio y aplicación de las iniciativas en diferentes países.

En el departamento Gestión Global de la Energía (GEM), se encarga de gestionar los activos de generación y acudir al mercado para optimizarlos y garantizar unas mejores condiciones a los clientes. GEM es un equipo global dentro de Iberdrola, presente en diferentes países.

Como cualquier empresa que utiliza datos a gran escala Iberdrola usa la inteligencia artificial para reducir costes, mejorar procesos y encontrar nuevas oportunidades. En concordancia a la **Ilustración 10** se utiliza el ML para impulsar la innovación, reducir costes, mejorar la relación con el cliente, mejorar la segmentación de anuncios y mejorar la segmentación del publico objetivo. Aun que estos beneficios del ML en empresas son generales, Iberdrola saca valor de sus datos en muchas partes de la empresa. Además, hemos visto posteriormente en Error! Reference source not found. ejemplos muy concretos que se desarrollan también en Iberdrola. Este trabajo esta enfocado sobre todo en el uso del Machine Learning para realizar predicciones de la demanda de la electricidad con el objetivo de poder entender mejor el mercado y cubrirse de variaciones importantes en el precio/demanda. A continuación, se definirá mas en detalle los análisis predictivos, métodos estadísticos avanzados para la regresión y la clasificación, y programas de “Automated Machine Learning” o AutoML por sus siglas en inglés.

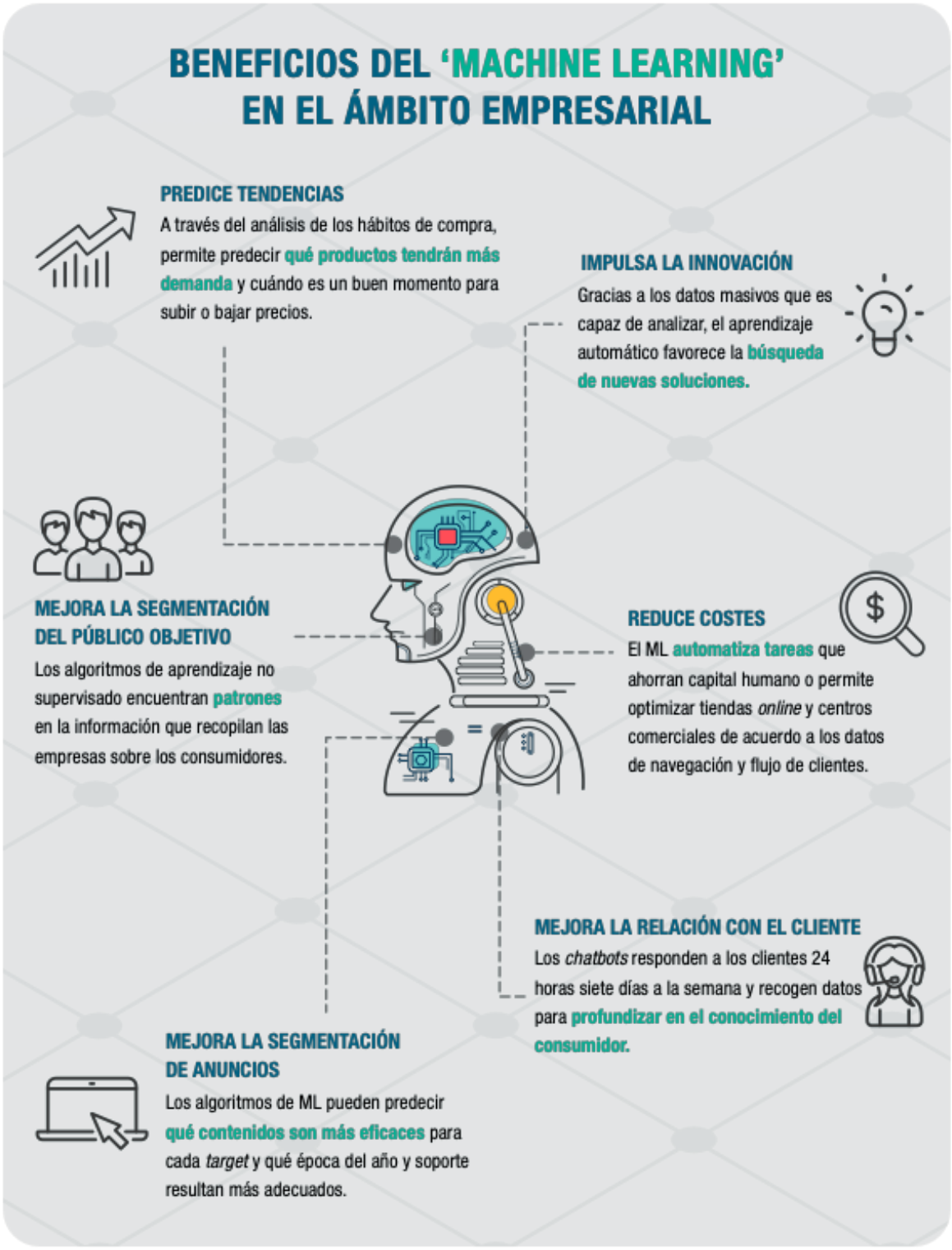


Ilustración 10. Beneficios del "Machine Learning" en el ámbito profesional. [12]

ANÁLISIS PREDICTIVO Y PROGRAMAS DE "AUTOMATED ML"

A continuación, se va a plantear la definición del análisis predictivo, las descripciones de los diferentes aprendizajes (Supervisado, no supervisado, semisupervisado y por refuerzo). Después aclararemos los problemas de Regresión y clasificación dentro del aprendizaje supervisado antes de explicar los métodos de aprendizajes mas utilizados.

Análisis predictivo

El análisis predictivo es una sección de la analítica avanzada que realiza predicciones sobre resultados futuros utilizando datos históricos combinados con modelos estadísticos, técnicas de extracción de datos y aprendizaje automático. Las empresas emplean el análisis predictivo para encontrar patrones en estos datos e identificar riesgos y oportunidades. Este asociado mucho con Big Data y ciencia del dato

Para sacar partido de los datos, los científicos de datos (o "data scientist" en ingles) utilizan algoritmos de aprendizaje profundo, aprendizaje automático, para encontrar patrones y hacer predicciones sobre eventos futuros. Entre ellos se encuentran la regresión lineal y no lineal, las redes neuronales, las máquinas de vectores de suporte y los árboles de decisión. Los aprendizajes obtenidos a través de la analítica predictiva pueden utilizarse posteriormente dentro de la analítica prescriptiva para impulsar acciones basadas en conocimientos predictivos. El aprendizaje automático o ML se fundamenta de cuatros grupos de algoritmos cuales vamos a explicar a continuación:

Aprendizaje Supervisado

El aprendizaje supervisado, también conocido como aprendizaje automático supervisado, es una subcategoría del aprendizaje automático y la inteligencia artificial. Se define por su uso de conjuntos de datos etiquetados para entrenar algoritmos que clasifiquen datos o predigan resultados con precisión. A medida que los datos de entrada se introducen en el modelo, éste ajusta sus ponderaciones hasta que el modelo se ha ajustado adecuadamente, lo que ocurre como parte del proceso de validación cruzada. El aprendizaje supervisado ayuda a las organizaciones a resolver una serie de problemas del mundo real a escala, como clasificar pacientes con una enfermedad cardiaca.

El aprendizaje supervisado utiliza un conjunto de entrenamiento para enseñar a los modelos a obtener el resultado deseado. Este conjunto de datos de entrenamiento incluye datos de entradas y datos de salidas correctas, que permiten al modelo aprender con el tiempo. El algoritmo mide su precisión a través de la función de pérdida, ajustándose hasta que el error se haya minimizado lo suficiente.

El aprendizaje supervisado puede dividirse en dos tipos de problemas cuando se extraen datos: la clasificación y la regresión cuyos veremos mas en profundidad en la parte Clasificación y Regresión.

Aprendizaje no supervisado

El aprendizaje no supervisado, también conocido como aprendizaje automático no supervisado, utiliza algoritmos de aprendizaje automático para analizar y agrupar conjuntos de datos no etiquetados. Estos algoritmos descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humana. Su capacidad para descubrir similitudes y diferencias en la información lo convierten en la solución ideal para el análisis exploratorio de datos, las estrategias de venta cruzada, la segmentación de clientes y el reconocimiento de imágenes. Los modelos de aprendizaje no

supervisado se utilizan para tres tareas principales: análisis de grupos o "clustering", asociación y reducción de la dimensionalidad.
Para ilustrar ese tipo de aprendizaje, una aplicación posible seria en el sector de la salud, diagnosticar los pacientes con rapidez y precisión mediante modelos de detección, clasificación y segmentación de imágenes.

Aprendizaje semisupervisado
El aprendizaje semisupervisado es una técnica de aprendizaje automático que etiqueta algunos de los datos de la base de datos de una IA, pero no todos. Con este punto de referencia, la técnica puede inferir o aprender lo que representan los datos no etiquetados con mucha más precisión que en el aprendizaje no supervisado (en el que no se etiquetan los datos), pero sin el tiempo y los costes necesarios para el aprendizaje supervisado en lo cual se etiquetan todos los datos.

Aprendizaje por refuerzo
El aprendizaje por refuerzo es un enfoque del aprendizaje automático que se inspira en la psicología conductista. Es similar a cómo un niño aprende a realizar una nueva tarea. El aprendizaje por refuerzo contrasta con otros enfoques de aprendizaje automático en que no se le dice explícitamente al algoritmo cómo realizar una tarea, sino que resuelve el problema por sí mismo.
A medida que un agente, que puede ser un coche auto conducido o un programa que juega al ajedrez, interactúa con su entorno, recibe un estado de recompensa en función de su rendimiento, como por ejemplo conducir hasta su destino de forma segura o ganar una partida. A la inversa, el agente recibe una penalización por actuar de forma incorrecta, como salirse de la carretera o ser jaqueado.
Con el tiempo, el agente toma decisiones para maximizar su recompensa y minimizar su penalización mediante programación dinámica. La ventaja de este enfoque de la inteligencia artificial es que permite que un programa de IA aprenda sin que un programador le diga cómo debe realizar la tarea.

Clasificación y Regresión

Dentro del aprendizaje supervisado visto anteriormente hemos visto que permite resolver dos tipos de problemas cuales son regresión y clasificación. Vamos a ver ahora la diferencia entre ellos antes de enfocarnos sobre problemas de regresión y mas adelante Time Series.

Clasificación: La clasificación utiliza un algoritmo para asignar con precisión los datos de la prueba a categorías específicas. Reconoce entidades específicas dentro del conjunto de datos e intenta sacar algunas conclusiones sobre cómo deben etiquetarse o definirse esas entidades. Los algoritmos de clasificación más comunes son los clasificadores lineales, las máquinas de vectores de soporte (SVM, por su sigla en ingles), los árboles de decisión, los vecinos más próximos o "Nearest Neighbour" y los bosques aleatorios o "Random Forest".

Regresión: La regresión se utiliza para comprender la relación entre las variables dependientes e independientes. Se suele utilizar para hacer proyecciones, como por ejemplo de los ingresos por ventas de una empresa determinada. La regresión lineal, la regresión logística y la regresión polinómica son algoritmos de regresión populares. Este trabajo consiste en resolver este tipo de problema mediante un programa de AutoML que definiremos mas adelante.

Algoritmos de aprendizaje supervisados

Aquí vamos a explicar brevemente métodos de aprendizaje mas utilizados. Se definirá cada método utilizando su nombre ingle porque resuelta mas fácil de entender el trabajo posteriormente, ya que todos los modelos que se van a usar vienen del programa de AutoML DataRobot, cual esta en ingles.

Neural Networks
Las redes neuronales, utilizadas principalmente para los algoritmos de aprendizaje profundo, procesan los datos de entrenamiento imitando la interconectividad del cerebro humano a través de capas de nodos. Cada nodo se compone de entradas, pesos, un sesgo (o umbral) y una salida. Si ese valor de salida supera un umbral determinado, se "dispara" o activa el nodo, pasando los datos a la siguiente capa de la red. Las redes neuronales aprenden esta función de mapeo a través del aprendizaje supervisado, ajustándose en base a la función de pérdida a través del proceso de descenso de gradiente. Cuando la función de coste es igual o cercana a cero, podemos confiar en la precisión del modelo para obtener la respuesta correcta.

Naive Bayes
Naive Bayes es un enfoque de clasificación que adopta el principio de la independencia condicional de las clases del Teorema de Bayes. Esto significa que la presencia de una característica no influye en la presencia de otra en la probabilidad de un resultado determinado, y cada predictor tiene un efecto igual en ese resultado. Hay tres tipos de clasificadores Naive Bayes: Naive Bayes multinomial, Naive Bayes Bernoulli y Naive Bayes gaussiano. Esta técnica se utiliza principalmente en la clasificación de textos, la identificación de spam y los sistemas de recomendación.

Regresión lineal
La regresión lineal se utiliza para identificar la relación entre una variable dependiente y una o más variables independientes, y suele aprovecharse para hacer predicciones sobre resultados futuros. Cuando sólo hay una variable independiente y una variable dependiente, se llama regresión lineal simple. Cuando el número de variables independientes aumenta, se denomina regresión lineal múltiple. Para cada tipo de regresión lineal, se busca trazar una línea de mejor ajuste, que se calcula mediante el método de los mínimos cuadrados. Sin embargo, a diferencia de otros modelos de regresión, esta línea es recta cuando se representa en un gráfico.

Regresión logística
Mientras que la regresión lineal se utiliza cuando las variables dependientes son continuas, la regresión logística se selecciona cuando la variable dependiente es categórica, lo que significa que tienen resultados binarios, como "verdadero" y "falso" o "sí" y "no". Aunque ambos modelos de regresión buscan comprender las relaciones entre las entradas de datos, la regresión logística se utiliza principalmente para resolver problemas de clasificación binaria, como la identificación de spam.

Máquina de vectores de apoyo (SVM)
Una máquina de vectores de soporte es un popular modelo de aprendizaje supervisado desarrollado por Vladimir Vapnik, que se utiliza tanto para la clasificación de datos como para la regresión. Sin embargo, normalmente se utiliza para problemas de clasificación, construyendo un hiperplano donde la distancia entre dos clases de puntos de datos es

máxima. Este hiperplano se conoce como la frontera de decisión, que separa las clases de puntos de datos (por ejemplo, naranjas frente a manzanas) a ambos lados del plano.

K-nearest neighbor

El algoritmo K-Nearest neighbor, también conocido como KNN, es un algoritmo no paramétrico que clasifica los puntos de datos en función de su proximidad y asociación con otros datos disponibles. Este algoritmo asume que se pueden encontrar puntos de datos similares cerca unos de otros. Por ello, busca calcular la distancia entre los puntos de datos, normalmente a través de la distancia euclidiana, y luego asigna una categoría basada en la categoría más frecuente o en la media.

Su facilidad de uso y su bajo tiempo de cálculo lo convierten en el algoritmo preferido por los científicos de datos, pero a medida que el conjunto de datos de prueba crece, el tiempo de procesamiento se alarga, lo que lo hace menos atractivo para las tareas de clasificación. El KNN se suele utilizar para motores de recomendación y reconocimiento de imágenes.

Decision Tree

Popular por su inteligibilidad y sencillez, el “Decision Tree” o árbol de decisión es uno de los algoritmos más fáciles de visualizar e interpretar, lo que resulta muy útil a la hora de presentar los resultados a un público no técnico, como suele ocurrir en la industria. Si consideramos simplemente un árbol en un estado similar al de un diagrama de flujo, de la raíz a las hojas, donde el camino a una hoja desde la raíz define las reglas de decisión sobre las características, entonces ya tenemos un buen nivel de intuición necesario para entender el aprendizaje del árbol de decisión.

Random Forest

El random forest es otro algoritmo flexible de aprendizaje automático supervisado que se utiliza tanto para la clasificación como para la regresión. El "forest" hace referencia a una colección de árboles de decisión no correlacionados, que luego se fusionan para reducir la varianza y crear predicciones de datos más precisas.

XGBoost

XGBoost es un algoritmo de aprendizaje automático basado en un árbol de decisión que utiliza un marco de refuerzo de gradiente. En los problemas de predicción con datos no estructurados (imágenes, texto, etc.), las redes neuronales artificiales tienden a superar a todos los demás algoritmos o marcos. Sin embargo, cuando se trata de datos estructurados/tabulares de pequeño a mediano tamaño, los algoritmos basados en árboles de decisión se consideran los mejores de su clase en este momento. En la siguiente Ilustración 11 se muestra la evolución de los algoritmos basados en árboles a lo largo de los años.

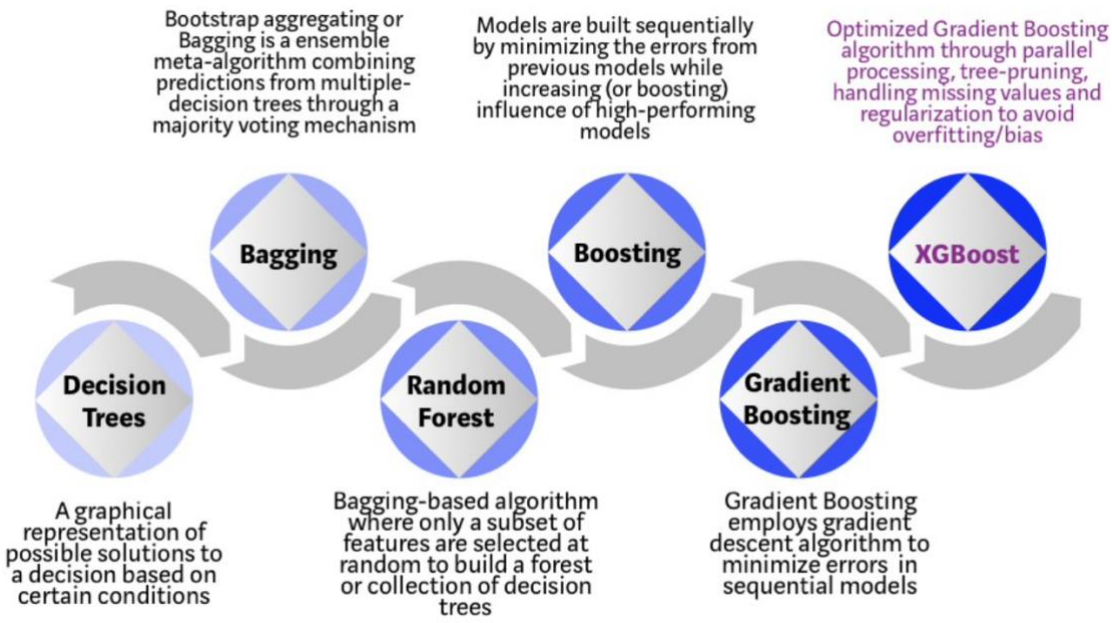


Ilustración 11. Evolución de los algoritmos basados en árboles a lo largo de los años. [13]

Ahora que se entiende mejor los conceptos claves del aprendizaje supervisado, vamos a estudiar programas de AutoML y precisamente DataRobot, ya que es el programa utilizado a lo largo de ese trabajo.

PROGRAMAS DE AUTOML Y DATAROBOT

En este capitulo se trata de profundizar el autoML, desarrollar el concepto de AutoML y estudiar como esta creciendo ese mercado. Dado que a continuación se entrena modelos solo mediante de herramienta de autoML es importante entender en que consiste y como se esta desarrollando ese sector. Además, veremos aplicaciones muy comunes de aplicaciones de autoML a casos de negocios reales.

Automated Machine Learning

El aprendizaje automático (AutoML) representa un cambio fundamental en la forma en que las organizaciones de todos los tamaños abordan el aprendizaje automático y la ciencia de los datos. La aplicación de los métodos tradicionales de aprendizaje automático a los problemas empresariales del mundo real requiere mucho tiempo, recursos y es un reto. Requiere expertos en varias disciplinas, incluidos los científicos de datos, que son algunos de los profesionales más solicitados en el mercado laboral en estos momentos.

El aprendizaje automático cambia esta situación, facilitando la creación y el uso de modelos de aprendizaje automático en el mundo real mediante la ejecución de procesos sistemáticos en datos brutos y la selección de modelos que extraen la información más relevante de los datos, lo que a menudo se denomina "la señal en el ruido". AutoML incorpora las mejores prácticas de aprendizaje automático de los mejores científicos de datos para hacer que la ciencia de los datos sea más accesible en toda la organización.

La construcción manual de un modelo de aprendizaje automático es un proceso de varios pasos que requiere conocimientos especifico, experiencia matemática y habilidades de ciencias de la computación, lo que es mucho pedir a una empresa, por no hablar de un científico de datos (siempre que pueda contratar y retener a uno). Y no sólo eso, sino que hay innumerables posibilidades de que se produzcan errores humanos y sesgos, lo que degrada la precisión del modelo y devalúa los conocimientos que se pueden obtener de él. El autoML permite a las organizaciones utilizar los conocimientos incorporados de los científicos de datos sin gastar tiempo y dinero para desarrollar las capacidades por sí mismos, mejorando al mismo tiempo el retorno de la inversión en iniciativas de ciencia de datos y reduciendo la cantidad de tiempo que se necesita para capturar el valor.

El AutoML hace posible que las empresas de todos los sectores (sanidad, los mercados financieros, la tecnología financiera, la banca, el sector público, el marketing, la venta al por menor, los deportes, la fabricación, etc.) aprovechen la tecnología de aprendizaje automático y de IA, una tecnología que antes solo estaba al alcance de las organizaciones con grandes recursos a su disposición. Al automatizar la mayor parte de las tareas de modelado necesarias para desarrollar e implementar modelos de aprendizaje automático, el aprendizaje automático permite a los usuarios empresariales implementar soluciones de aprendizaje automático con facilidad, permitiendo así que los científicos de datos de una organización se centren en problemas más complejos. [14]

Un mercado creciente y sus principales actores

El tamaño del mercado del autoML está creciendo rápidamente a medida que la tecnología se hace más popular. Un informe de 2020 elaborado por Research & Markets comparte que el mercado generado tiene unos ingresos de 300 millones de dólares en 2019 y se espera que aumente hasta los 14.500 millones de dólares en 2030. [15] Según el mismo informe, los principales impulsores de este crecimiento son :

- El aumento de la demanda de soluciones de detección de fraude más eficientes
- La creciente necesidad de recomendación personalizada de productos
- La creciente importancia del “predictive lead scoring”

El interés por el autoML también está aumentando rápidamente, y se espera que este aumento continúe durante al menos unos años más.

El importe total de la financiación es un buen indicador del éxito, ya que los inversores prefieren poner su dinero en empresas de éxito. Por lo tanto, el proveedor mejor financiado puede ser considerado como un proveedor de autoML exitoso que puede obtener el mayor beneficio a largo plazo.

Según Crunchbase, la solución de autoML mejor financiada es actualmente DataRobot, con una financiación total de 431 millones de dólares. Después de DataRobot, H2O.ai ocupa el segundo lugar, y Dataiku es el tercer proveedor con 151,1 millones de dólares y 147 millones de dólares, respectivamente. En la siguiente tabla, puede encontrar los otros proveedores de autoML que cuentan con una financiación superior al millón de dólares, incluyendo su año de fundación y el tamaño de la empresa. [16]
Otros proveedores de autoML que cuentan con una financiación superior al millón de dólares. [16]

Tabla 1. Otros proveedores de autoML que cuentan con una financiación superior al millón de dólares [16]

COMPANY	TOTAL FUNDING AMOUNT	FOUNDED YEAR	NUMBER OF EMPLOYEES
DataRobot	$430.6M	2012	1001-5000
H2O.ai	$151.1M	2012	11-50
Dataiku	$146.8M	2013	251-500
dotData	$43M	2018	51-100
Compellon (Acquired by Hoist Finance)	$11.6M	2010	11-50
Coldlight Solutions (Acquired by PTC)	$11M	2007	1-10
PurePredictive	$10.2M	2011	11-50
Ople	$10M	2017	11-50
Predikto (Acquired by United Technologies)	$7.6M	2013	11-50
VEDA Data Solutions	$7.2M	2015	1-10
Snark AI	$1.7M	2018	1-10
MyDataModels	$1M	2018	11-50
Tazi.ai	$1M	2015	11-50
DMWay	$1M	2013	1-10

DataRobot

DataRobot es una empresa de IA creada en 2012. Tiene como objetivo de permitir a las organizaciones aprovechar el poder transformador de la IA a través de su plataforma de IA y ayudar a los clientes a convertir rápidamente los datos en valor. Su plataforma de IA permite a los clientes preparar sus datos, crear y validar modelos de aprendizaje automático -incluyendo modelos de series temporales - y desplegar y supervisar esos modelos en una única solución.

Aprendizaje supervisado en DataRobot
De los grupos de algoritmos que hemos visto anteriormente en el apartado Automated Machine Learning, DataRobot propone el uso de Aprendizaje supervisado, semisupervisado y no supervisado. Además, propone el aprendizaje operacional (Machine Learning Operations o MLOps). La tecnología y las prácticas de Machine Learning Operations (MLOps) proporcionan un medio escalable y gobernado para desplegar y gestionar modelos de aprendizaje automático en entornos de producción.

El aprendizaje automático supervisado es uno de los motores más potentes que permiten a los sistemas de IA tomar decisiones empresariales con mayor rapidez y precisión que los humanos. Las empresas de todos los sectores lo utilizan para resolver problemas como:

- Reducir la pérdida de clientes
- Determinar el valor de vida del cliente
- Personalizar las recomendaciones de productos
- Asignación de recursos humanos
- Previsión de ventas
- Previsión de la oferta y la demanda
- Detectar el fraude
- Predicción del mantenimiento de los equipos

La variada librería de algoritmos de aprendizaje automático de DataRobot y su tecnología de planes de modelos (model Blueprint) incorporan algoritmos de aprendizaje automático supervisado como “bagging, boosting, deep learning, frequency-severity methods, generalized additive models, generalized linear models, kernel-based methods, random forests” y muchos otros (cuando utilizaremos DataRobot mas tarde, tendremos la información en ingles, por lo cual no han sido traducidos). Además, el equipo de científicos de datos experimentados de DataRobot investiga, desarrolla y prueba constantemente nuevos algoritmos de código abierto para incorporar los modelos de aprendizaje automático supervisado más avanzados. [17]

Base de datos de series temporales
En ese trabajo se quiere predecir la evolución de la demanda a dos meses para hacer posiciones de coberturas o tener una interpretación adelantada del mercado eléctrico. Para realizar previsión o predicción, trabajaremos con bases de datos de series temporales o “Time Series”. Una base de datos de series temporales (TSDB por su siglo en ingles) es una base de datos optimizada para datos “time-stamp” o datos de series temporales. Los datos de series temporales son simplemente mediciones o eventos que son rastreados, monitoreados, muestreados y agregados en el tiempo. Puede tratarse de métricas del servidor, monitorización del rendimiento de la aplicación, datos de red, datos de sensores, eventos, clics, operaciones en un mercado y muchos otros tipos de datos analíticos.

Una base de datos de series temporales se construye específicamente para manejar las métricas y los eventos o las mediciones que tienen una marca de tiempo. Una TSDB está optimizada para medir los cambios en el tiempo. Las propiedades que hacen que los datos de las series temporales sean muy diferentes a otras cargas de trabajo de datos son la gestión del ciclo de vida de los datos, el resumen y los escaneos de gran rango de muchos registros.

Como se ha afirmado antes DataRobot esta constantemente investigando, desarrollando y probando los últimos algoritmos de código abierto para incorporar los modelos de aprendizaje automático supervisado más avanzados. Por lo cual a continuación explicamos unos de los modelos que mas aparecen en el trabajo.

Early stopping of Gradient Boosting: El "early stopping support" en Gradient Boosting nos permite encontrar el menor número de iteraciones que sea suficiente para construir un modelo que generalice bien a los datos no vistos.

Ridge Regressor: La regresión Ridge es una versión regularizada de la regresión lineal. La regresión Ridge permite a los algoritmos de aprendizaje automático no solo ajustarse a los datos, sino también mantener los pesos del modelo lo más pequeños posible.

ARIMA: Uno de los métodos más utilizados en la previsión de series temporales se conoce como modelo ARIMA, que significa “Auto Regressive Integrated Moving Average”. ARIMA es un modelo que puede ajustarse a los datos de las series temporales para predecir puntos futuros de la serie.
Podemos dividir el término ARIMA en tres términos, AR, I, MA:
AR(p) es el modelo auto regresivo, el parámetro p es un número entero que confirma cuántas series rezagadas se van a utilizar para predecir períodos futuros.

I(d) es la parte de diferenciación, el parámetro d indica el número de órdenes de diferenciación que se van a utilizar para hacer estacionarias las series.
MA(q) significa modelo de media móvil o "moving average", la q es el número de términos de error de previsión retardados en la ecuación de predicción. SARIMA es ARIMA estacional y se utiliza con series temporales con estacionalidad.

Auto ARIMA: La ventaja de utilizar Auto ARIMA sobre el modelo ARIMA es que después del paso de preprocesamiento de datos se puede saltar pasos y ajustar directamente el modelo. Utiliza los valores AIC (Akaike Information Criterion) y BIC (Bayesian Information Criterion) generados al probar diferentes combinaciones de valores p, q y d para ajustar el modelo.

ESTRATEGIA PARA CONZEPTUALIZACION

En ese capítulo, se trata de explicar los pasos seguidos y las métricas utilizadas para llegar al mejor modelo posible. Una vez definido los objetivos y la estrategia seguida, veremos más en detalle la adquisición de datos, la modelización de series temporales con DataRobot, las métricas de optimizaciones que se usarán y la visualización de los datos.

OBJETIVO DEL TRABAJO

En un contexto general, una buena previsión de la demanda eléctrica permite a las empresas que se dedican a venderla comprar nada más que la electricidad que se necesita. Tener los mejores modelos de predicciones de la demanda eléctrica representa una ventaja competitiva muy importante. Este trabajo se enfoca en conceptualizar modelos de IA utilizando solamente datos públicos como indicadores económicos del país o de industrias específicas.

Actualmente GEM ya cuenta con modelos de predicción con unos resultados muy buenos. Estos modelos están en producción y se utilizan en la operativa diaria. Los modelos desarrollados durante el TFM podrán nutrir este ecosistema de modelos, bien como modelos finales o intermedios, así como abrir la posibilidad de una nueva vía de desarrollo más autónoma gracias al autoML.

Se recuerdan a continuación los objetivos del trabajo:

1. Entender si un modelo realizado con autoML, sin conocimientos específicos del mercado eléctrico y con datos públicos puede resultar de utilidad
2. Conseguir modelos precisos con información pública y herramientas de autoML enfocados al mercado eléctrico con un MAPE inferior a 3%.
3. Entender el funcionamiento del departamento de Gestión de la energía.
4. Estudiar la puesta en producción de estos modelos, especialmente a través de herramientas de BI.

ESTRATEGIA DEL TRABAJO

Antes de decidir que se iba a predecir la evolución de la demanda (mensualmente), se han planteado varios proyectos de predicción interesantes comprobando la disponibilidad de los datos que se necesitaba según el proyecto. A continuación, tenemos una lista de ideas de proyectos:

- Generación renovable por tecnología/combustible (MWh) en España
- Precios horarios del mercado diario en España
- Cómo va a cambiar el mercado de la electricidad en el futuro (flexibilidad, generación, demanda) (1~2 años)

- Precio de capacidad de energía
- Estimación de La Elasticidad del Precio de La Demanda de Electricidad En España

Es bastante complicado construir bases de datos con datos horarios y diarios dado que no existen muchas fuentes que publiquen informaciones relevantes en el horizonte de tiempo requerido.
Después de estudiar los diferentes proyectos y los datos que se pueden conseguir, ha quedado claro que se va a predecir la evolución de la demanda. Sirve para ajustar la demanda de los meses que vienen y se pueden probar varios inputs de datos macroeconómicos (por ejemplo, probar sectores económicos diferentes, la actividad laboral o datos demográficos).

A continuación, se observa la lista de datos que se quieren usar en bases de datos como input para entrenar modelos:

1. Demanda eléctrica en España (Output): REE
2. IRE industria/servicios
3. Importaciones y exportaciones por sectores de actividad: Data Comex
4. Temperaturas de ciudades de España: AEMET
5. Tasa de Actividad:
6. Producciones (equipo, energía, industrial, bienes):
7. Días laborales para cada mes
8. Sábados y Domingos para cada mes
9. días festivos para cada mes
10. Incidencia acumulada Covid 19
11. Número de contagios coronavirus en España: cnecovid
12. Número de muertes por covid 19: cnecovid
13. Número de ventas de coches eléctricos: EVvolumes (suscripción)
14. Ibex 35: Yahoo Finance
15. Cotización de las empresas de energías: Yahoo Finance
16. Ruido en varios barrios de ciudades
17. Precipitaciones en esas ciudades: Instituto de Energía Solar
18. Sugerencias, quejas y agradecimientos (año en curso)

Teniendo en cuenta la variable output que se quiera predecir y las variables input, queda por hacer la adquisición de esos datos, construir datasets después de haber realizado un análisis exploratorio de los datos, entrenar los modelos, capturar sus valores y diseñar un dashboard que permita facilitar la utilización de las predicciones dentro de

la empresa. Se debe conseguir como producto final un Dashboard que contiene suficiente información además de las predicciones para mejorar la planificación y operación del departamento de Trading de Iberdrola.

ADQUISICION DE DATOS PUBLICOS

Existen cuatro métodos de adquisición de datos:

- recopilación de nuevos datos;
- conversión/transformación de datos existentes;
- intercambio de datos;
- y compra de datos.

Esto incluye la recopilación automatizada (por ejemplo, de datos derivados de sensores), el registro manual de observaciones empíricas y la obtención de datos existentes de otras fuentes.
En el apartado anterior hemos visto los datos que se van a buscar y juntar para crear bases de datos y entrenar modelos de Machine Learning mediante DataRobot. Debido a lo que ese trabajo se realiza con datos públicos, se define ese término de la siguiente manera:
Los datos públicos son informaciones que pueden ser utilizadas, reutilizadas y redistribuidas libremente por cualquier persona sin que existan restricciones legales locales, nacionales o internacionales de acceso o uso. En otras palabras, cualquier persona puede acceder a esos datos y utilizarlos tomando en cuenta siempre la política de datos de la entidad que detiene los mismos.
En la empresa, los datos pueden clasificarse como públicos si la información está disponible para todos los empleados y todas las personas o entidades externas a la corporación. Ejemplos de datos públicos en la empresa son los comunicados de prensa, las descripciones de puestos de trabajo y los materiales de marketing destinados al público en general.

Definir si la fuente de datos es buena

Antes de descargar los datos escogidos se comprueba las fuentes de datos de los datos encontrados. Para reconocer una fuente de datos como “buena” éstos se evalúan siguiendo los tres pasos siguientes:

1. Comprobar el dominio: Mirar las tres letras que aparecen al final del nombre de dominio del sitio, como "edu" (educativo), "gov" (gubernamental), "org" (sin ánimo de lucro) y "com" (comercial). Por lo general, los sitios web .edu y .gov son creíbles, pero hay que ser cauteloso con los sitios que utilizan estos sufijos en un intento de fraude. Los sitios web comerciales, como los de organizaciones de noticias de renombre, también pueden ser buenas fuentes, pero investigue un poco para buscar signos de fiabilidad.

2. Examinar con detenimiento la fuente: Comprobar si hay autores en el artículo o estudio. Si es el caso observar si citan o enlazan a fuentes autorizadas. Además, comprobar si se menciona credenciales y la fecha de publicación. En algunos casos, puede no importar si la fuente es antigua o no se ha actualizado

recientemente, pero en campos de estudio en los que la información puede cambiar rápidamente, los datos pueden estar obsoletos.

3. Buscar información adicional que respalde lo que se ha encontrado. A medida que se encuentra información, intentar verificar su autenticidad y legitimidad utilizando otros sitios fiables. Si se encuentra otro sitio creíble que contradice la fuente original, es posible que haya que seguir investigando.

Técnicas utilizadas para adquirir datos

Una vez que se ha comprobado que la fuente y los datos son buenos, hay que descargarlos para consolidarlos en un mismo dataset antes de poder entrenar cualquier modelo. En el mejor escenario, encontramos un fichero ".csv" que contiene todos los datos requeridos y solamente hace falta descargarlo. Sin embargo, muchas fuentes de datos no son tan fácilmente accesibles. Puede ocurrir que no se puedan descargar los datos porque haya un límite que impida descargar más de 3 años o solo que los datos están en otro formato. Por lo cual se desarrollan a continuación las diferentes técnicas de adquisición de datos utilizados en ese trabajo.

API

Una interfaz de programación de aplicaciones, o API por sus siglas en inglés ("Application Programming Interface"), permite a las empresas poner los datos y la funcionalidad de sus aplicaciones a disposición de desarrolladores externos de terceros, socios comerciales y departamentos internos de sus empresas. Esto permite que los servicios y productos se comuniquen entre sí y aprovechen los datos y la funcionalidad de cada uno a través de una interfaz documentada. Los desarrolladores no necesitan saber cómo se implementa una API; simplemente utilizan la interfaz para comunicarse con otros productos y servicios. El uso de las API se ha disparado en la última década, hasta el punto de que muchas de las aplicaciones web más populares hoy en día no serían posibles sin las API.

Una API es un conjunto de reglas definidas que explican cómo se comunican los ordenadores o las aplicaciones entre sí. Las interfaces de programación de aplicaciones se sitúan entre una aplicación y el servidor web, actuando como una capa intermediaria que procesa la transferencia de datos entre sistemas.

Así es como funciona una API:

1. Una aplicación cliente inicia una llamada a la API para obtener información, lo que se conoce como solicitud. Esta solicitud se procesa desde una aplicación hasta el servidor web a través del Identificador Uniforme de Recursos (URI por sus siglas en inglés) de la API e incluye un verbo de solicitud, cabeceras y, a veces, un cuerpo de solicitud.
2. Tras recibir una solicitud válida, la API realiza una llamada al programa externo o al servidor web.
3. El servidor envía una respuesta a la API con la información solicitada.
4. La API transfiere los datos a la aplicación solicitante inicial.

En el trabajo utilizaremos varias API de varias fuentes de datos para adquirir los datos mencionados anteriormente. Permite ahorrar tiempo considerando que se puede automatizar tanto la descarga de datos como la limpieza y las modificaciones que se quieren aplicar cada vez.

Web Scrapping

"Web scrapping" o raspado de web, es un término que designa varios métodos utilizados para recopilar información en Internet. Por lo general, esto se hace con un software que simula la navegación humana por la web para recoger fragmentos específicos de información de diferentes sitios web. El web scraping también se denomina extracción de datos de la web, "screen scraping" o "web harvesting". El Web scrapping es esencialmente una forma de minería de datos. Elementos como los informes meteorológicos, los detalles de las subastas, los precios del mercado o cualquier otra lista de datos recopilados pueden ser buscados en los esfuerzos de Web scrapping.

La práctica del Web scrapping ha suscitado mucha controversia porque las condiciones de uso de algunos sitios web no permiten ciertos tipos de extracción de datos. A pesar de los desafíos legales, el Web scrapping promete convertirse en una forma popular de recopilar información a medida que este tipo de recursos de datos agregados adquieren mayor capacidad.

A lo largo del trabajo se utiliza Web scrapping para conseguir ciertos datos utilizando el lenguaje de programación Python y las librerías requests y selenium.

La librería requests es el estándar para hacer peticiones HTTP en Python. Abstrae las complejidades de hacer peticiones detrás de una bonita y sencilla API para que se pueda centrar en interactuar con los servicios y consumir datos en la aplicación.

Selenium es un proyecto que engloba una serie de herramientas y bibliotecas que permiten y apoyan la automatización de los navegadores web. Proporciona extensiones para emular la interacción del usuario con los navegadores, un servidor de distribución para escalar la asignación de navegadores y la infraestructura para las implementaciones de la especificación W3C WebDriver que permite escribir código intercambiable para los principales navegadores web. En ese trabajo se utiliza Selenium con Chrome Driver.

Limpieza y preprocesamiento de los datos con Pandas

El siguiente punto trata de limpiar y preprocesar los datos para poder juntarlos y exportar como ".csv" el dataset que se va a entrenar. Para ese procesamiento se usa una librería de Python llamada Pandas, que se utiliza para un análisis de datos más rápido, limpieza de datos y preprocesamiento de datos. Pandas está construida sobre la biblioteca numérica de Python, llamada numpy.

Los pasos típicos de preprocesamiento de datos para el aprendizaje automático son:

1. Cargar datos (por ejemplo, cargar un archivo csv de Internet)
2. exploración de datos (por ejemplo, estadísticas de resumen, visualización de datos, etc.)

3. limpiar los datos (por ejemplo, manejar los datos que faltan)
4. transformación de datos (por ejemplo, ingeniería de características, escalado, reformateo como matriz Numpy o Spark RDD (Resilient Distributed Dataset))

Después de ese último paso, tenemos que consolidar los diferentes datasets para que los valores de cada columna coincidan con las otras según la fecha y el año. Hay que cuidar en el tipo de las columnas al momento de realizar "joins" para no tener problemas.

Dado que tenemos un dataset con todos los datos que queremos usar para nuestro modelo (tanto las variables input como la variable output), puede empezar la predicción de modelos en DataRobot. Asegurarse de que para entrenar un modelo de serie temporal el dataset incluya en su conjunto de datos una columna "time-stamped", comprobar que cada fila representa una unidad de tiempo y utilizar la misma unidad de tiempo de forma coherente en todo el conjunto de datos.

MODELIZACIÓN DE SERIES TEMPORALES CON DATAROBOT

En ese punto vamos a desarrollar varios términos técnicos de ML aplicados en DataRobot. Se trata de explicar diferentes pasos que vamos a seguir al momento de hacer el desarrollo de prueba de concepto. Además, trata de entender más en profundidad cómo DataRobot está configurado por defecto y cuáles son sus configuraciones.

Partición de datos

Después de identificar un conjunto de datos para utilizar con su modelo, DataRobot divide los datos en subconjuntos, utilizando algunos datos para el entrenamiento del modelo y otros datos para verificar que el modelo puede generalizar a nuevos datos. Esto se llama partición de datos. Las tres particiones son

Partición de entrenamiento: La subsección de un conjunto de datos a partir de la cual el algoritmo de aprendizaje automático aprende las relaciones entre las características y la variable objetivo

Partición de validación: La subsección de un conjunto de datos utilizada para identificar las relaciones entre los resultados conocidos para la variable objetivo y las otras características del conjunto de datos

Partición de retención: La subsección de un conjunto de datos que proporciona una estimación final del rendimiento del modelo de aprendizaje automático después de haber sido entrenado y validado. La partición de retención se denomina a veces datos de prueba ("testing data").

La partición de datos le permite desarrollar modelos más precisos que son relevantes para los datos que recogerá en el futuro y permite también establecer un compromiso entre sesgo y varianza del modelo. Al entrenar los datos, validarlos y probarlos, se obtiene una idea de la precisión de los resultados del modelo. Por defecto, DataRobot se entrena con el 64% de los datos, se valida con el 16% de los datos y se prueba con el 20% de los datos.

Tipos de series temporales

Existen tres tipos de series temporales, las cuales son: regular, semi-regular o irregular.

Serie temporal regular
Ese tipo de serie tiene intervalos de tiempo fijos entre filas. Unos ejemplos de series temporales regular pueden ser datos de ventas diarias en los que no hay intervalos, datos de un negocio que está abierto 7 días a la semana, datos de sensores que informan cada minuto o por último datos de una planta de fabricación que funciona 24 horas al día, 7 días a la semana

Serie temporal semirregular
La serie temporal semirregular está construida con intervalos de tiempo fijos entre la mayoría de las filas, con algunos espacios regulares. Para ilustrar ese tipo de serie, unos ejemplos serían datos diarios del mercado de valores en los que se excluyen los fines de semana; datos de lunes a viernes, de 09:30 a 16:00 o un negocio que sólo abre de lunes a viernes de 08:00 a 17:00.

Serie temporal irregular
En la serie irregular, el último tipo de serie, la distancia temporal entre datos consecutivos no sigue un patrón fijo. Se utiliza en series temporales para operaciones con bonos, huelgas de luz o accidentes de tráfico.

El tipo de previsión que puede realizar depende del tipo de conjunto de datos. DataRobot maneja los tres tipos de datos de series temporales. Con las series temporales regulares y semirregulares, puede realizar previsiones basadas en el tiempo y en las filas. Con los datos irregulares, sólo puede realizar previsiones basadas en las filas, ya que no hay suficiente información basada en el tiempo.
Con la previsión basada en el tiempo, puede realizar una previsión de hasta 1.000 pasos hacia adelante (suponiendo que tenga suficientes datos de entrenamiento).

El presente trabajo utiliza únicamente serie temporal regular dado que la evolución de la demanda eléctrica está disponible mensualmente y se ha visto anteriormente que las fuentes de datos públicas más interesantes comparten datos mensuales.

Out of time validation vs Time series modelling

Aunque se utilizará solamente modelado de series temporales es importante precisar que en función del tiempo DataRobot ofrece dos tipos de modelado:

Out of time validation
La validación fuera de tiempo o “out of time validation” (OTV) es similar a un modelo típico de clasificación o regresión, pero conserva los datos en orden temporal en lugar de barajarlos. Esto permite entrenar con datos más antiguos y validar con datos más recientes. La OTV sólo utiliza las características presentes en el conjunto de datos y no realiza ninguna ingeniería de características adicional.

Modelización de series temporales

La modelización de series temporales o “Time series modeling”, como la OTV, mantiene los datos en orden temporal, pero también realiza ingeniería de características de series temporales. Esta ingeniería de característica construye características de retardo (“lag features”) que se basan en los rasgos temporales originales, pero con retardo añadido. Las características retardadas pueden aumentar el poder de predicción del modelo basado en el tiempo. La ingeniería de características permite pronosticar múltiples pasos temporales en el futuro. Por ejemplo, con las series temporales, se puede prever la venta de entradas entre 14 y 30 días.

La elección de utilizar la OTV o la modelización de series temporales depende a menudo del problema que se está resolviendo. Como verá en ejercicios posteriores, a veces tiene sentido probar ambos enfoques para ver cuál proporciona la mayor señal y ayuda a resolver su problema de la mejor manera. Con DataRobot, es fácil probar rápidamente ambos enfoques.

Feature Derivation Window and Forecast Distance

Al crear un proyecto de series temporales, dos de los parámetros que se pueden actualizar son Ventana de derivación de características (FDW por su siglo en ingles) y la distancia de previsión (FD por su siglo en ingles). La Ventana de derivación de características es la ventana que genera características de retardo y las estadísticas rodantes. Por otra parte, Distancia de previsión representa la distancia hacia el futuro que se desea pronosticar. La distancia de previsión depende del caso de uso. DataRobot permite pronosticar hasta 1.000 intervalos de tiempo en el futuro si se tienen suficientes datos de entrenamiento.

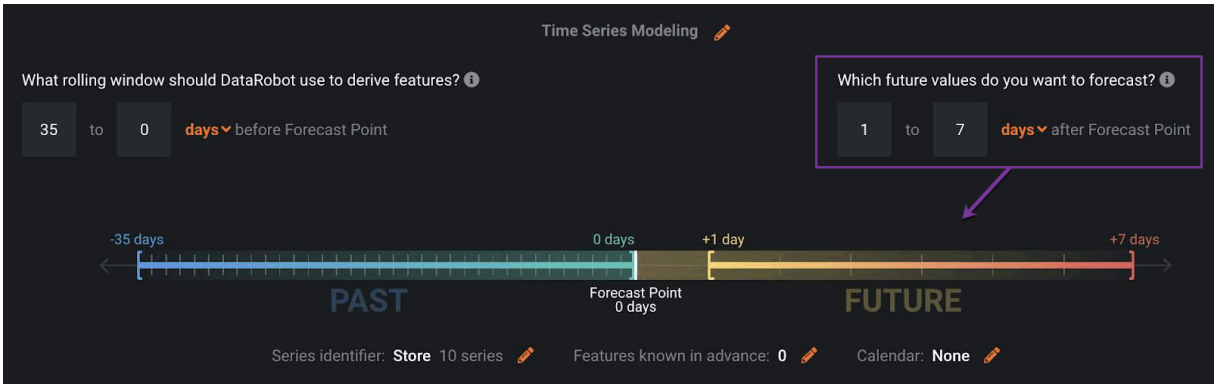


Ilustración 12. "Feature Derivation Window" y "Forecast Distance" en DataRobot. [18]

“Partitionning” y “Backtesting”

En el caso de las series temporales, no se pueden tomar muestras aleatorias de los datos en particiones, como se hace en el aprendizaje automático tradicional. Hacerlo sería ignorar la estructura temporal inherente a los datos.

Por lo tanto, el enfoque correcto de particionamiento con series temporales es el “backtesting”. Con el “backtesting”, se entrena con datos históricos y se valida con datos recientes. Con la partición, se puede ajustar los periodos de validación y el número de pruebas retrospectivas para que se adapten mejor a las necesidades, teniendo en cuenta los datos y el caso de uso.

"Missing Values" y reentrenamiento del modelo

La aplicación de automatización de series temporales de DataRobot maneja los "missing values" o valores ausentes de varias maneras diferentes. Si el valor que falta se encuentra en el objetivo que se quiere predecir, la automatización de series temporales elimina ese registro incluso si las covariables tienen datos válidos. En los otros casos se utiliza la mediana de la columna de los valores característicos y se crean características de retardo para los valores que faltan.

Con la modelización de series temporales, el reentrenamiento de un modelo para su despliegue puede no utilizar el 100% del conjunto de datos cargados. La aplicación de Series Temporales Automatizadas se reentrena en las filas más recientes utilizando el mismo periodo de entrenamiento (cantidad de datos) utilizado por otros modelos en la Tabla de Líderes ("leaderboard").

MÉTRICAS DE OPTIMIZACIÓN DEL MODELO

La definición de las métricas de optimización del modelo es un paso fundamental para comprender la precisión de un modelo y poder compararlo con otros. Después se define lo que se considera un "buen modelo" para un modelo que predice la evolución de la demanda eléctrica con datos públicos.

Métricas de optimización del modelo

Las métricas de optimización del modelo ayudan a comprender la precisión con la que el modelo predice el valor objetivo de los puntos de datos que no se han visto antes. Diferentes problemas requieren diferentes tipos de métricas. Las métricas comunes para los proyectos de clasificación son LogLoss y AUC (área bajo la curva ROC), mientras que una métrica común para la regresión es RMSE (error cuadrático medio). DataRobot recomienda automáticamente una métrica de optimización basada en su conjunto de datos. En ese trabajo se utiliza sobre todo el MAPE (error medio de porcentaje absoluto) y el RMSE.

Definir si un modelo de aprendizaje automático es "bueno"

El problema de modelización predictiva es único para cada caso. Esto incluye los datos específicos que tiene, las herramientas que se están utilizando y el objetivo que se quiere conseguir. Aunque existen ya modelos que predicen la evolución de la demanda de la electricidad, tener el mejor modelo posible es muy importante considerando que represente una ventaja competitiva dentro de una misma industria. Por lo cual no se pueden saber las métricas de optimizaciones de los modelos existentes y se requiere definir para sus propios modelos de aprendizaje automático las métricas que se consideran "buenas". Se pueden tener ideas de cómo de "bueno" es un modelo basándose en el conocimiento de la industria, pero en nuestro caso haciendo un modelo solamente con datos públicos no se sabe si los resultados conocidos se pueden alcanzar. Lo mejor que se puede hacer es comparar el rendimiento de los modelos de aprendizaje automático en sus datos específicos con otros modelos también

entrenados en los mismos datos y volver a entrenar los mismos modelos con otras variables de entrada comprobando como las distintas variables impactan el modelo.

Para tener un objetivo más concreto, se puede calcular el coeficiente de variación de la variable de salida del modelo es decir la variable que se quiere predecir. El coeficiente de variación de la demanda de la electricidad, por la cual tenemos datos desde 2007 hasta el último mes es de 6,20% como se puede observar en la Ilustración 13 a continuación.

```
Entrée [97]: coef_variation = DATASET_DEMANDA['value'].std()/DATASET_DEMANDA['value'].mean()*100
             print(coef_variation,"%")

             6.195072995778969 %
```

Ilustración 13. Coeficiente de variación de demanda eléctrica (mensual)

El coeficiente de variación permite hacernos una idea de métricas de optimización del modelo como el RMSE o el MAPE que se puede considerar bueno. Dado que el coeficiente de variación para la evolución de la demanda eléctrica es de 6,2% predecirla con un modelo con un error menor puede ser interesante. El MAPE que se quiere conseguir en ese trabajo es acercar los 2% de error medio de porcentaje absoluto.

USO EFICAZ DE LA INFORMACIÓN – VISUALIZACIÓN

Un “dashboard” es una herramienta de gestión de la información que rastrea, analiza y muestra visualmente los indicadores clave de rendimiento (KPI por su siglo en ingles), las métricas y los puntos de datos clave para supervisar la salud de una empresa, un departamento o un proceso específico. Son personalizables para satisfacer las necesidades específicas de un departamento y una empresa. Entre bastidores, un dashboard se conecta a sus archivos, anexos, servicios y API, pero en la superficie muestra todos estos datos en forma de tablas, gráficos de líneas, gráficos de barras y medidores. Un dashboard es la forma más eficaz de realizar un seguimiento de múltiples fuentes de datos, ya que proporciona una ubicación central para que las empresas supervisen y analicen el rendimiento. El seguimiento en tiempo real reduce las horas de análisis y las largas líneas de comunicación que antes suponían un reto para las empresas.

En cuento a herramientas de inteligencia de negocio (BI), Power BI y Tableau se consideran como las "mejores". A continuación, vemos cual son sus características respectivas.
Power BI es la herramienta de Microsoft que está estrechamente relacionada con Excel. Esta en el mercado desde 2013. Power BI no es una sola aplicación o software, sino que es un conjunto de servicios y aplicaciones basado en la nube. Sus componentes principales, cuales son Power Query, Power Pivot, Power View, Power Map y Power Q&A, permiten cubrir las necesidades de inteligencia de negocio. Una de las áreas en las que Power BI brilla más en comparación con sus compañeros es su integración con Office 365. Según las cifras, Office 365 es el servicio en la nube más utilizado por el número de usuarios. Por ello, los usuarios de Office 365 se sienten intuitivamente atraídos por Power BI para la inteligencia y el análisis empresarial. Además, Microsoft está impulsando a sus usuarios a utilizar Power BI para visualizar sus datos. También tiene sus propias desventajas. Por un lado, la funcionalidad de Power BI es un poco inferior a la de sistemas de BI más antiguos como Tableau. Microsoft llama a la combinación de

diferentes herramientas y aplicaciones de Power BI "Microsoft Power Platform". Incluye Power BI, Power Apps, Power Automate, Power Virtual Agents, etc. Desafortunadamente, Power BI se queda un poco corto cuando se trata de una implementación local. Para disfrutar de toda la potencia de Power BI “on-premise” (el uso de instalación en vez de tener las bases de datos en la nube), es necesario instalar Power BI Report Service y el SQL Server.

En cuento a Tableau, es una plataforma de análisis que existe desde 2003, 10 años antes que Power BI. Debido a su ventaja de tiempo, es comprensible que Tableau sea más potente y refinada que Power BI. La diferencia entre las dos herramientas solo puede ser anotado por usuarios muy avanzados o cuando se usa datasets muy grandes. Tableau le cubre de principio a fin: desde la colaboración, el análisis, el descubrimiento de contenidos, la preparación y el acceso a los datos, hasta la implementación. En comparación con Power BI, Tableau tiene un despliegue más flexible. Puede instalar la versión de escritorio sin tener que instalar el servidor SQL. Por desgracia, las características de Tableau tienen un precio muy alto. Lo que se puede hacer con Power BI costará siete veces más con Tableau. Además, usar Tableau de verdad significa construir su propio almacén de datos. Su implementación más el coste incremental de las licencias de Tableau requiere inversión importante. Ademas, la conexión a más aplicaciones de terceros le costará más. En cuanto a los precios, Tableau es más caro que Power BI por un amplio margen. Para la realización de dashboards en ese trabajo se utilizará el programa Tableau. Es una plataforma de análisis visual cuya función es de resolver problemas, permitiendo a las personas y a las organizaciones sacar el máximo partido de sus datos.

Por cuanto ese trabajo se basa en el desarrollo de un modelo de predicción mediante DataRobot, el dashboard debe permitir facilitar la toma de decisiones para los departamentos de trading de Iberdrola que opera en el mercado eléctrico”. A continuación, citamos informaciones que pueden ser relevantes a un trader para planificar o operar en el mercado:

- Datos históricos de la evolución de la demanda (grafico de línea)
- Predicciones de los dos próximos meses (información)
- Acción por realizar (comprar/vender) según la predicción realizada (información)
- Datos históricos de una industria del IRE (grafico de línea)
- Predicción de esa industria (información)
- Coeficiente de error entre el valor real de la evolución de la demanda y el previsto, para los últimos 12 meses (grafico de línea)
- Media del coeficiente de error (información)

El Dashboard debe permitir al trader de plantearse diferentes componentes del mercado eléctrico, recomendarle la operación adecuada(comprar/vender) con la cantidad de electricidad que se va a necesitar y dar informaciones para que tenga en cuenta las limitaciones del modelo con sus coeficientes de errores pasados al momento de ejecutar la operación.

DESAROLLO DE PRUEBA DE CONCEPTO

METODOLOGÍA

El objetivo principal de la prueba de concepto consiste en cumplir los objetivos planteados anteriormente. Éstos incluyen entrenar un modelo de aprendizaje automático supervisado que predice la demanda mensual de electricidad a dos meses futuros, con un MAPE por debajo de 3%, con datos públicos y mediante DataRobot.
En relación a la problemática expuesta, vamos a seguir el proceso entero del desarrollo de un modelo en Iberdrola desde la adquisición de los datos hasta el dashboard, cuyo objetivo es aportar informaciones relevantes para mejorar la toma de decisiones. Incluye, entre varias tareas, analizar las métricas de optimización obtenidas o realizar predicciones para abril y mayo para comprobar predicciones realizadas con datos reales. Una vez que se ha visto el proceso entero para obtener valor de un modelo, se comparan todos los modelos realizados con la evolución de las distintas estrategias seguidas para encontrar el mejor modelo. Finalmente se trata de destacar las lecciones aprendidas durante ese desarrollo de prueba de concepto tanto sobre la predicción de la evolución de la demanda como sobre entrenar modelos predictivos con datos públicos o con herramientas de autoML.

ADQUISICION DE DATOS PUBLICOS

Se ha considerado el uso de seis datos de entrada para entrenar el modelo. Vamos a ver cómo se ha obtenido cada dato diferente y a hacer un análisis exploratorio para estudiar las relaciones entre las variables. Para ese modelo en concreto se quieren usar variables de entrada que permiten analizar si existen entre ellas patrones que se repiten en el tiempo. Los datos que vamos a adquirir para construir el DataFrame final y entrenar el modelo son los siguientes:

- Evolución de la demanda de la electricidad en España
- Temperaturas mínimas y máximas mensuales medias
- Covid
- Número de días por meses
- Número de sábados y domingos
- Número de días festivos

Adquisición de los datos y creación del dataset

En primer lugar, se descarga la variable de salida del modelo. Los datos se pueden encontrar en la página web de la REE y se pueden descargar a través de la API de REE [19]. En la siguiente imagen se aprecia la respuesta de una “request” para descargar la evolución de la demanda con un rango de tiempo superior a 24 meses. Además, no se pueden descargar datos anteriores al año 2007. Para poder tener todos los datos de la evolución de la demanda eléctrica en un mismo documento se ha realizado un código disponible en la parte de Anexos llamada *Descarga de la evolución de la demanda eléctrica*.

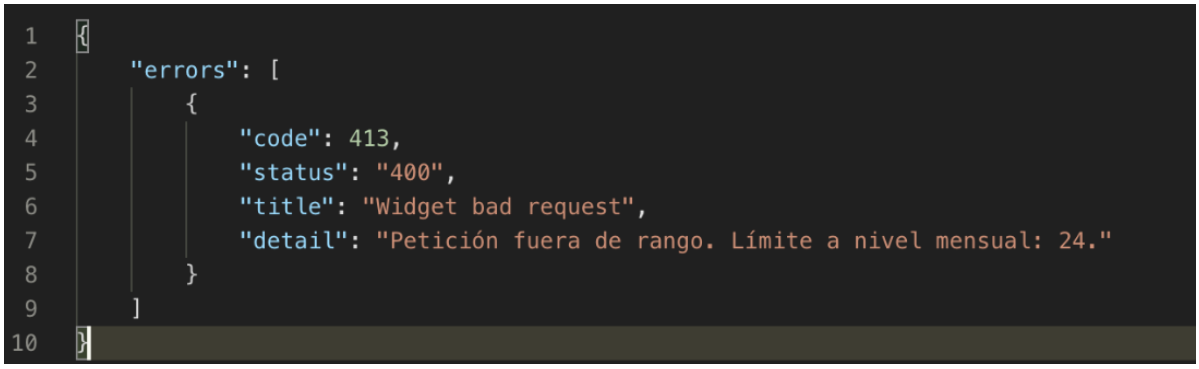


Ilustración 14. Limite de datos que se pueden consultar a nivel mensual en la REE.

Ese código primeramente crea una lista de rangos de fechas con 24 meses de separación entre cada fecha (Ilustración 15). Después se realiza un bucle "for" haciendo peticiones a la API de la REE para tener los datos entre cada dos fechas. Para conectarse a la API de la REE hay que precisar los parámetros principales: cuales son el idioma, la categoría y el widget que se quiere consultar (en ese caso ponemos "evolución" que se corresponde a la evolución de la demanda eléctrica mensual).

```
Out[2]: [datetime.date(2010, 1, 1),
         datetime.date(2011, 12, 31),
         datetime.date(2012, 1, 1),
         datetime.date(2013, 12, 31),
         datetime.date(2014, 1, 1),
         datetime.date(2015, 12, 31),
         datetime.date(2016, 1, 1),
         datetime.date(2017, 12, 31),
         datetime.date(2018, 1, 1),
         datetime.date(2019, 12, 31),
         datetime.date(2020, 1, 1),
         datetime.date(2021, 12, 31),
         datetime.date(2021, 1, 1),
         datetime.date(2021, 5, 31)]
```

Ilustración 15. Lista de fechas para realizar varias peticiones mediante un bucle "for".

Luego se añaden las fechas por las cuales se quieren consultar los datos utilizando la lista creada previamente y se pide el valor "time_trunc" al mes. Cada vez que se descargan los datos de 2 años, le aplicamos modificaciones mediante la librería Pandas para aislar las columnas "Date" y "value". Se cambia el tipo de la columna "Date" a "time-stamped" para que coincida con otros dataset más tarde y la columna "value" corresponde a los valores mensuales de la evolución de la demanda.
Una vez las modificaciones hechas cada descarga de dos años se añade en una lista denominada "datos". Cuando tenemos todos los datos en una lista, la concatenamos y la pasamos a DataFrame antes de quitar todos los valores duplicados que podría tener la tabla.
Finalmente, después de comprobar que no haya duplicados y que tenemos los datos buenos, descargamos el DataFrame como fichero ".csv", cual contiene todos los datos de la demanda eléctrica desde 2007 hasta el mes anterior. No se ha descargado el mes actual porque no está completo y tiene un impacto negativo sobre las predicciones.

En segundo se descarga las temperaturas medias de las mínimas y máximas. Para realizarlo se usa la API de la AEMET [20] cual necesita los siguientes parámetros: una

clave API o "API key" que permite identificarnos cuando descargamos datos, una fecha de inicio, una fecha final y un idema que corresponde a la estación que se quiere consultar. La clave API para acceder a los datos se obtiene a través de la pagina web y se recibe por correo. Tal como la demanda eléctrica, descargamos los datos desde 2007 hasta el día de hoy y para el idema se utiliza las temperaturas de la estación "Madrid Aeropuerto". El código se puede encontrar en la parte *Descarga de las temperaturas medias mensuales de las mínimas y máximas* del Anexo. Después de descargar los datos se crea una nueva columna "Date" para que sea similar en formato y en tipo a la columna "Date" del DataFrame anterior. Así podremos juntar las tablas sin tener que hacer modificaciones.

- Covid
- Número de días por meses
- Número de sábados y domingos
- Número de días festivos

En tercero vamos a crear los DataFrames de las variables de entrada restantes dado que se pueden crear directamente con Pandas y librerías especificas según cada una. Para todos los DataFrames que vamos a crear a continuación se crea la misma columna "Date" que en los DataFrames anteriores para poder juntarlos una vez que hemos creado cada uno. El código se encuentra en la parte *Creación del dataset mediante Python* del Anexo.

Número de días por meses: Creamos una nueva columna titulada "num_days_per_month" con las mismas fechas que en la columna "Date". Para tener los números de días por meses, aplicamos la función de la librería datetime "days_in_month" para toda la columna.

Número de días festivos: Se utiliza la librería workalendar en Python para hacer un DataFrame de las fechas de cada día festivo. Para tener todos los años desde 2007 se ha utilizado un bucle "for". Con esos datos se crea un DataFrame y se realiza un "groupby" (agrupar por) para hacer un conteo de cuentas fiestas ocurren en el mismo y año.

Número de sábados y domingos: Para hacer un conteo de los sábados y domingos de cada mes, desde la columna fecha se ha creado dos columnas "year" y "month" y las hemos convertido en listas. Con la función disponible en la parte *Creación del dataset mediante Python* del Anexo, se hace un conteo respectivamente para sábados y domingos en listas. Después sumamos ambos resultados en una misma lista y añadimos los resultados en el DataFrame.

Datos de COVID-19: La pandemia de coronavirus (COVID-19) ha tenido un impacto importante en la evolución de la demanda eléctrica en España sobre todo durante el confinamiento como se puede observar en la Ilustración 16. Por tal motivo se quiere modelizar su impacto utilizando ceros cuando todavía no existía y unos desde los primeros casos hasta hoy en día.

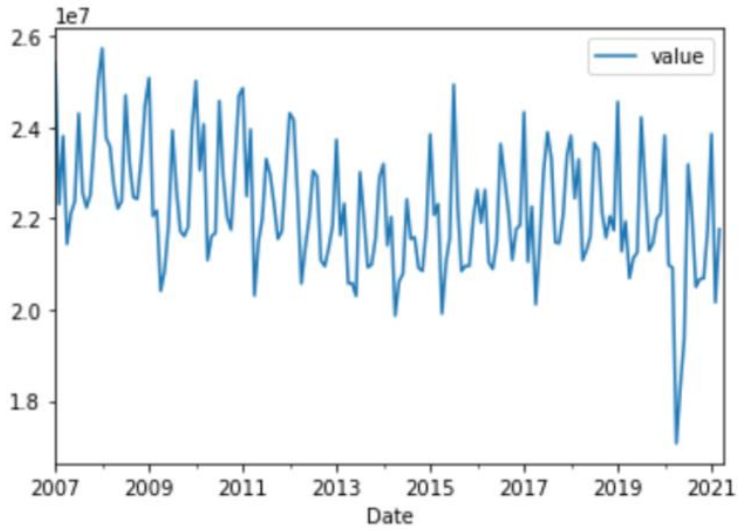


Ilustración 16. Evolución de la demanda eléctrica (GW) en España desde 2007.

Finalmente, consolidamos los DataFrames de cada variable (el código se encuentra en la parte *Creación del dataset mediante Python* del Anexo). Se realiza con la función de Pandas "merge", haciendo un "left join" en la columna "Date" de cada uno de los DataFrames. En la siguiente tabla se puede observar que los datos históricos que vamos a usar en el entrenamiento del modelo. Las variables de entrada como el número de días por mes o el número de sábados y domingos por mes son valores que se conocen en el momento de la predicción. Es importante distinguir estos dos tipos de datos al momento de configurar DataRobot para entrenar el modelo dado que para hacer predicciones tenemos que crear una tabla con ciertos datos conocidos al futuro.

Tabla 2. Últimas líneas del DataFrame después de haber juntado los datos.

	Date	value	num_days_per_month	saturday_sundays_count	tm_min	tm_max	festive_days_count	covid
166	2020-11-01	2.068827e+07	30	9	5.5	15.2	1.0	1
167	2020-12-01	2.176296e+07	31	8	2.3	10.7	3.0	1
168	2021-01-01	2.386055e+07	31	10	-1.7	8.1	2.0	1
169	2021-02-01	2.016023e+07	28	8	4.0	14.6	0.0	1
170	2021-03-01	2.176541e+07	31	8	3.0	16.9	0.0	1

Análisis Exploratorio del dataset

A continuación se muestran un pequeño resumen de los datos que tenemos en el Dataset y que vamos a usar para predecir la evolución de la demanda eléctrica, usando la funciones describe() y dtypes.

Tabla 2. Descripción de los datos con la función describe().

	value	num_days_per_month	saturday_sundays_count	tm_min	tm_max	festive_days_count	covid
count	1.710000e+02	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000
mean	2.221971e+07	30.432749	8.690058	8.432164	21.661988	0.830409	0.076023
std	1.376527e+06	0.825963	0.791771	6.250672	8.283493	0.901247	0.265814
min	1.706091e+07	28.000000	8.000000	-2.600000	8.100000	0.000000	0.000000
25%	2.133179e+07	30.000000	8.000000	2.950000	13.850000	0.000000	0.000000
50%	2.205439e+07	31.000000	8.000000	7.400000	20.400000	1.000000	0.000000
75%	2.312654e+07	31.000000	9.000000	14.000000	28.950000	1.000000	0.000000
max	2.574239e+07	31.000000	10.000000	20.700000	37.100000	3.000000	1.000000

```
Date                    datetime64[ns]
value                          float64
num_days_per_month               int64
saturday_sundays_count           int64
tm_min                         float64
tm_max                         float64
festive_days_count             float64
covid                            int64
dtype: object
```

Ilustración 17. Descripción de los datos con la función dtype.

Un punto importante en el que se ha basado para desarrollar el Dataset que se ha utilizado, como se va a explicar en el siguiente apartado, es que hay una correlación fuerte entre las variables tm_min, tm_mes y tm_max (**Error! Reference source not found.**). Tiene sentido que estén muy correlacionadas aun que tienen valores diferentes ya que las temperaturas van variando según las temporadas y van aumentando y disminuyendo de mañera similar. Se observa también una relación negativa moderada entre las variables de temperaturas y el conteo de los días festivos.

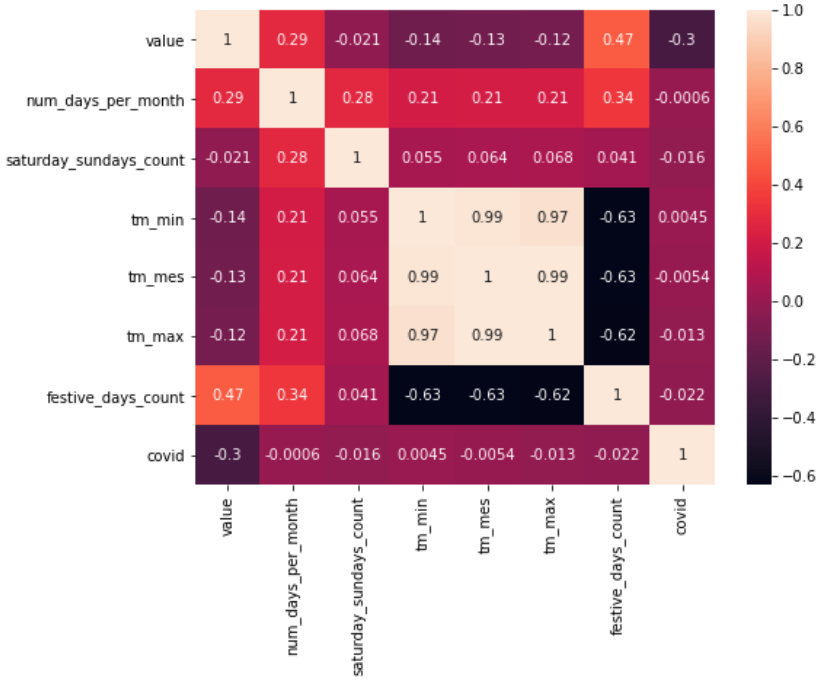


Ilustración 18. Matriz de correlación del dataset

Además, se ha decidido que de las tres variables tm_min, tm_mes y tm_max guardamos solamente tm_min y tm_max porque tener tres variables muy correlacionadas no añade valor al modelo.

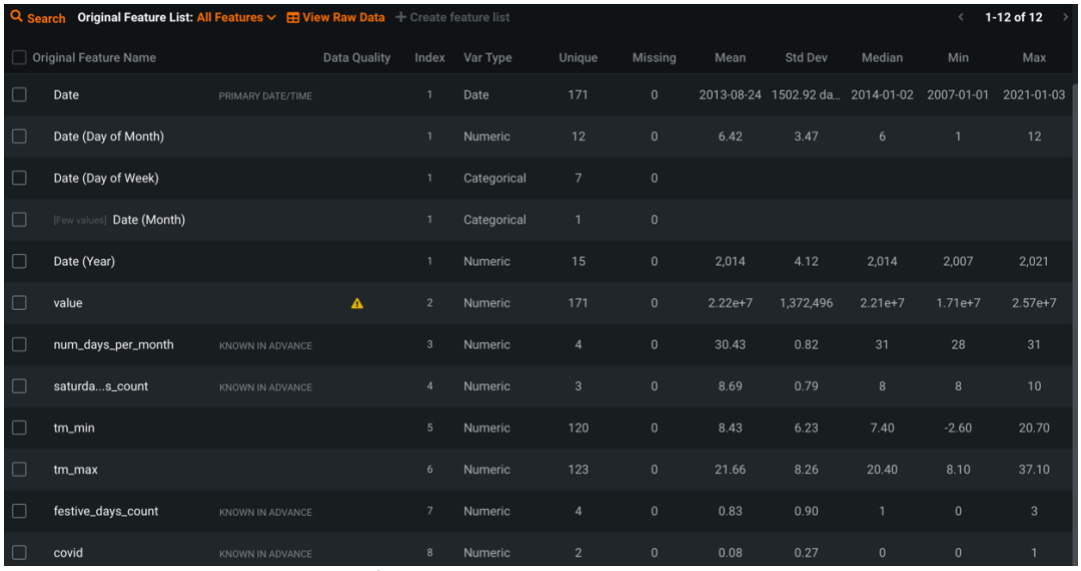
Resumiendo, se ha construido el Dataset con los datos enunciados anteriormente utilizando varias técnicas de adquisición de datos según la accesibilidad de cada uno. Dado que tenemos los datos juntos, en los próximos capítulos vamos a desarrollar como configurar DataRobot para entrenar modelos en DataRobot, analizar métricas de optimización del mejor modelo y predecir la demanda de la electricidad de los próximos meses.

MODELOS DE SERIES TEMPORALES

En este capítulo se utilizará DataRobot para entrenar modelos que predicen la demanda eléctrica mensual de los dos próximos meses. En primer lugar, se configura DataRobot para que entrene el modelo de la mejor mañera. Con ello se tratará de observar cual es el mejor modelo que ha entrenado y comprobar que sus predicciones ajustan la realidad.

Para el conjunto definido en el capítulo anterior, se ha creado una partición de los datos que los divide en 64 % para entrenamiento, 16 % para la validación y 20 % para test. Así se obtiene una idea de la precisión de los resultados del modelo. Más adelante se observará las métricas de optimizaciones y las predicciones del mejor modelo.

Primeramente, se crea un nuevo proyecto en DataRobot y se descarga el dataset creado previamente. DataRobot realiza un análisis exploratorio de los datos automáticamente como se puede observar en la Ilustración 19. Se dispone de un conjunto de datos de 172 registros y 12 variables (DataRobot cuenta la fecha y crea variables con solo los años, los meses y los días). Como se ha visto anteriormente DataRobot detecta automáticamente valores atípicos (“outliers”) o valores perdidos. Para nuestro dataset se puede observar con el triángulo amarillo que DataRobot ha detectado “outliers” en la variable “value”. Como se ha visto en la Ilustración 16 tiene sentido dado que la demanda eléctrica ha sido muy impactada por el confinamiento y ha bajado hasta 17.0609 GW en abril 2020. Parece muy complicado si no imposible encontrar una variable que puede adelantar este tipo de evento y una bajada tan importante ya que nunca había bajado tanto. Vamos a entrenar el modelo con esos valores atípicos, pero se podría realizar una derivación de esa variable ...



Original Feature Name	Data Quality	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
Date	PRIMARY DATE/TIME	1	Date	171	0	2013-08-24	1502.92 da...	2014-01-02	2007-01-01	2021-01-03
Date (Day of Month)		1	Numeric	12	0	6.42	3.47	6	1	12
Date (Day of Week)		1	Categorical	7	0					
[Few values] Date (Month)		1	Categorical	1	0					
Date (Year)		1	Numeric	15	0	2,014	4.12	2,014	2,007	2,021
value	⚠	2	Numeric	171	0	2.22e+7	1,372,496	2.21e+7	1.71e+7	2.57e+7
num_days_per_month	KNOWN IN ADVANCE	3	Numeric	4	0	30.43	0.82	31	28	31
saturda...s_count	KNOWN IN ADVANCE	4	Numeric	3	0	8.69	0.79	8	8	10
tm_min		5	Numeric	120	0	8.43	6.23	7.40	-2.60	20.70
tm_max		6	Numeric	123	0	21.66	8.26	20.40	8.10	37.10
festive_days_count	KNOWN IN ADVANCE	7	Numeric	4	0	0.83	0.90	1	0	3
covid	KNOWN IN ADVANCE	8	Numeric	2	0	0.08	0.27	0	0	1

Ilustración 19. Analisis exploratorio de DataRobot

Después de comprobar que los datos se han importado correctamente definimos la variable de salida: “value”, y la variable que representa el tiempo (“Date”) considerando que queremos utilizar serie temporal regular se elige “Automated time series forecasting with backtesting” (Ilustración 20). Además, se indica a DataRobot que se quiere predecir una única serie visto que no se quiere predecir más variables que la evolución de la demanda eléctrica (Ilustración 21).

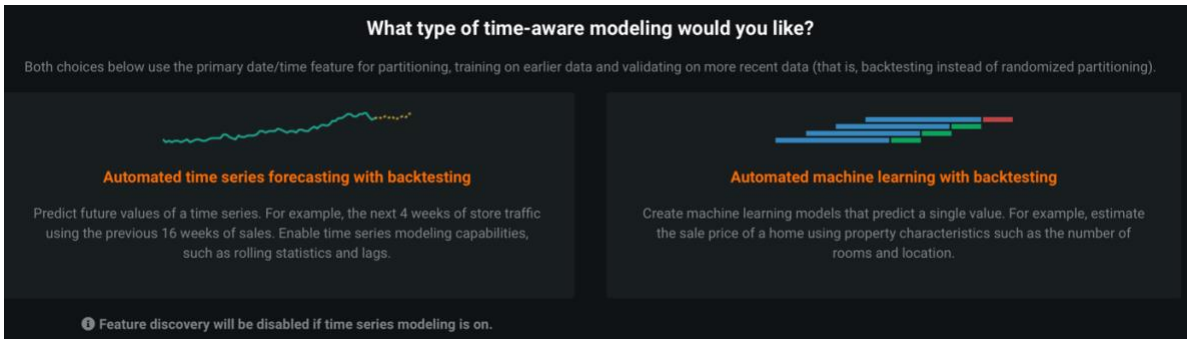


Ilustración 20. Opciones de DataRobot al momento de entrenar el dataset

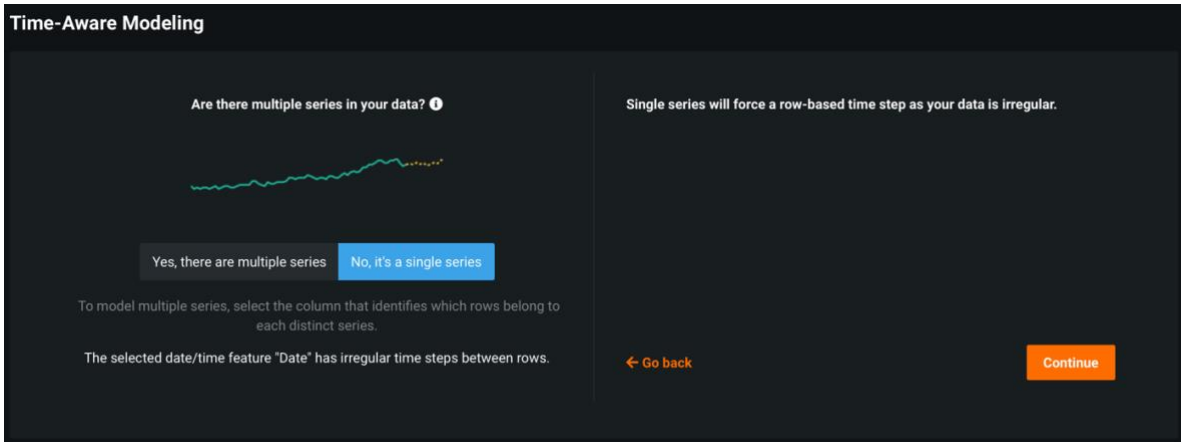


Ilustración 21. Tipos de series en DataRobot

La figura siguiente (Ilustración 22) permite optimizar la ventana de variables derivadas y la ventana de previsión. Se ha definido la ventana de variables derivados amplia para utilizar muchos datos históricos y una ventana de previsión a dos meses para intentar obtener pocas predicciones pero que aciertan con un error muy baja. En el capítulo de predicciones del modelo vamos a comprobar varias ventanas de previsión para observar como las predicciones van cambiando.

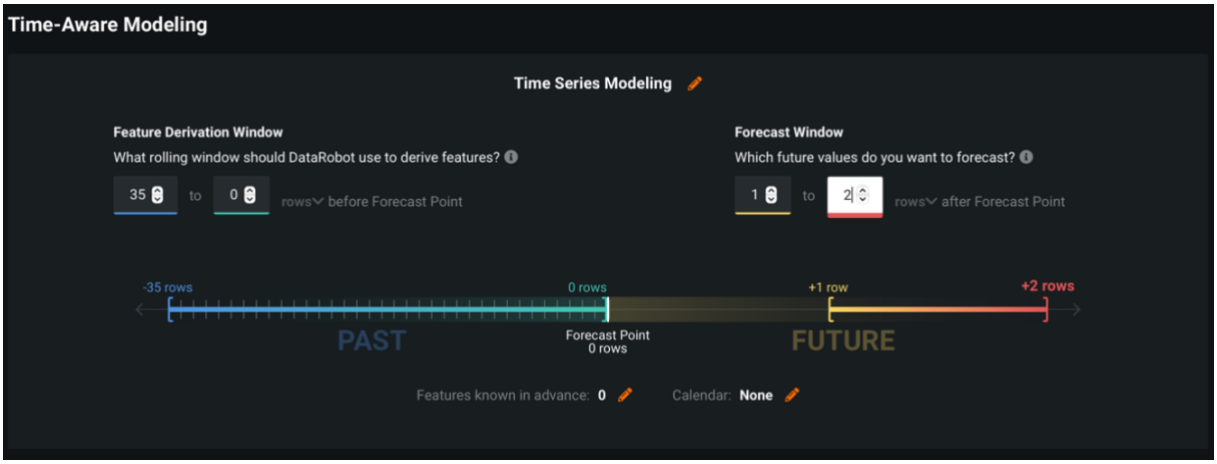


Ilustración 22." Feature Derivation Window" y "Forecast Window"

Por último, indicamos a DataRobot cuales son las variables que conocemos en el futuro como se puede observar en la Ilustración 23. En el momento de crear el dataset para hacer predicciones tendremos que rellenar las columnas de esas variables dado que el modelo lo utiliza para predecir. Podemos conocer el número de días del mes siguiente, el número de sábados y domingos o el número de días festivos.

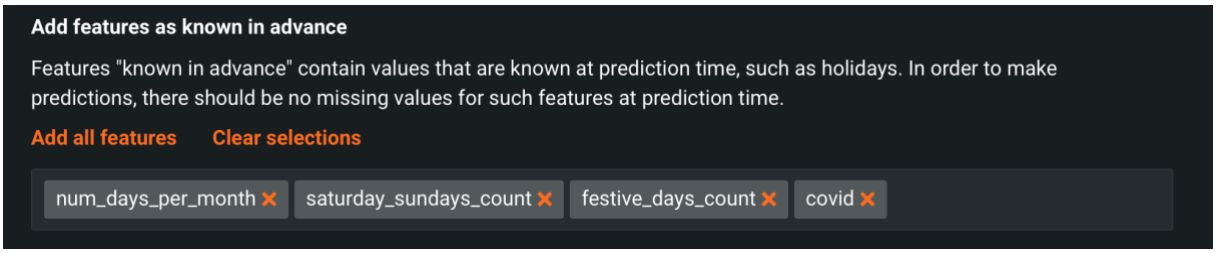


Ilustración 23. Añadir variables conocidas de antemano

Después de haber configurado DataRobot para entrenar modelos según la variable "value" y según el dataset que se ha creado, se puede iniciar el entrenamiento de varios modelos mediante el botón "Start" con la métrica de optimización RMSE elegida automáticamente por DataRobot (Ilustración 24). En el próximo capitulo se entrena varios modelos y se observa las métricas de optimizaciones.

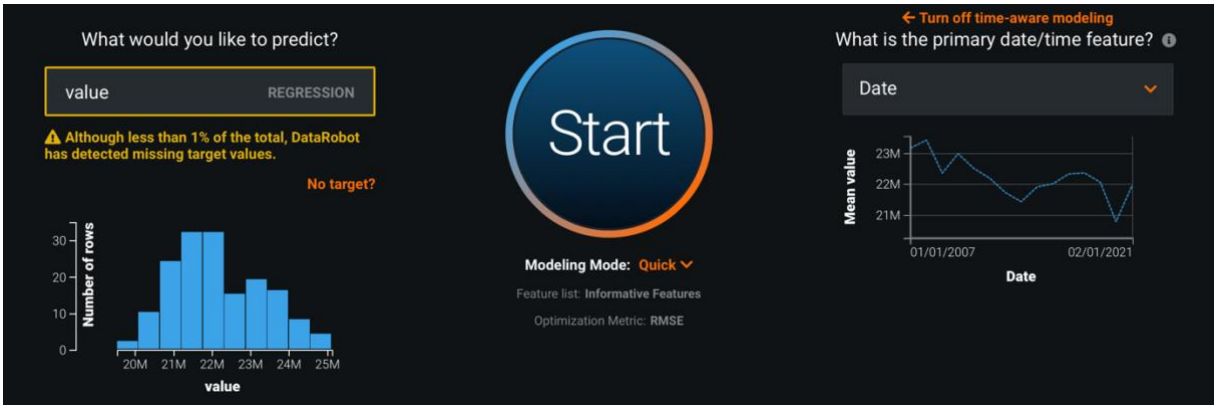


Ilustración 24. Captura de pantalla del botón de inicio para entrenar modelos en DataRobot

METRICAS DE OPTIMIZACION DEL MODELO

En esta parte del documento se analiza los modelos creados por DataRobot mediante las diferentes métricas de optimización. En la tabla siguiente se aprecia las métricas de optimización de los modelos que tienen RMSE y MAPE más bajos antes de probar como de bueno predicen. Se observa que los dos modelos que tienen las mejores métricas de optimización son el "Non-seasonal AUTOARIMA with Fourier terms" y el "AVG Blender". El "fourier terms" sirve para modelizar una serie que devuelve una matriz que contiene términos de una serie de Fourier, hasta el orden K, adecuada para su uso en ARIMA o AUTOARIMA. Una serie de Fourier es una serie infinita que converge puntualmente a una función periódica y continua a trozos (o por partes). En cuento al "AVG Blender", es un modelo que combina dos modelos para mejorar la precisión. En ese caso el AVG Blender crea un nuevo modelo a partir de los modelos que tienen las mejores métricas de optimización. En otras palabras, crea un nuevo modelo a partir del "Non-seasonal AUTOARIMA with Fourier terms" y del "eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3) dado que tienen los RMSE menores.

Tabla 3. Resultados de los modelos entrenados por DataRobot con las métricas de optimización RMSE y MAPE

Modelo	RMSE	MAPE
Non-seasonal AUTOARIMA with Fourier terms	7.6634e+5	2.7539
AVG Blender	6.5162e+5	2.4217
eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)	7.8103e+5	2.7985
eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)	8.1479e+5	2.8571
Ridge Regressor using Linearly Decaying Weights with Forecast Distance Modeling	9.4018e+5	3.7137

En la captura de pantalla a continuación (Ilustración 25) están representados los modelos entrenados por DataRobot, clasificados según los RMSE de cada uno. Se puede elegir la métrica de optimización de los modelos aun que DataRobot siempre recomienda utilizar una basado en las variables del dataset subido anteriormente. Es importante decir que con el dataset que se ha utilizado para entrenar modelos, cual contiene 172 registros, entrenar modelos muy complejos como un "eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)" no es muy relevante.

Aunque tiene un RMSE muy bajo, porque el dataset no tiene muchos datos, los modelos van a sufrir sobreajuste. En nuestro caso es más interesante utilizar variaciones y regresiones sobre los datos históricos para encontrar patrones en la evolución de la demanda eléctrica. Los tipos de modelos que permiten resolver este tipo de problema con la mejor precisión posible son modelos Auto Regresivos (AR) o AUTOARIMA. En el próximo apartado veremos las diferencias al momento de predecir entre estos dos tipos de modelos.

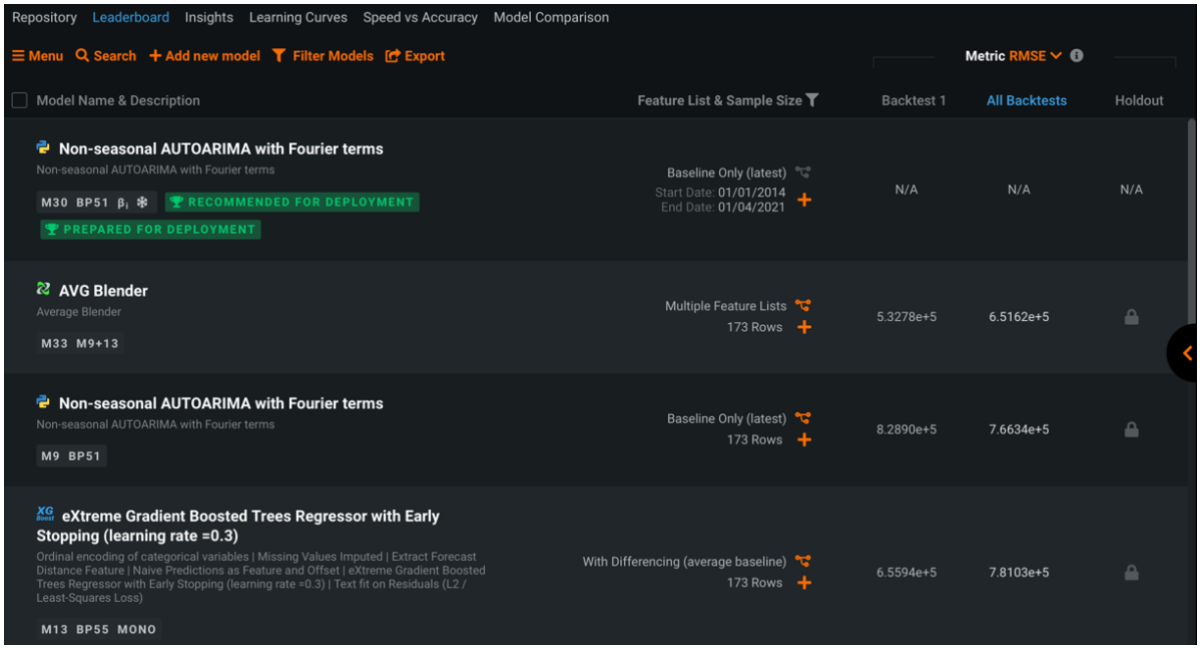


Ilustración 25. Modelos y descripciones en DataRobot

DataRobot nos permite entender mejor los modelos que se han entrenado mediante esquemas. La complejidad de los esquemas va variando según los modelos que se usan. En la Ilustración 26 se observa el esquema del modelo "Non-seasonal AUTOARIMA with Fourier terms" que vamos a usar a continuación para realizar predicciones. En comparación, el esquema del modelo “eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)” es mucho más complejo como lo demuestra la Ilustración 27.

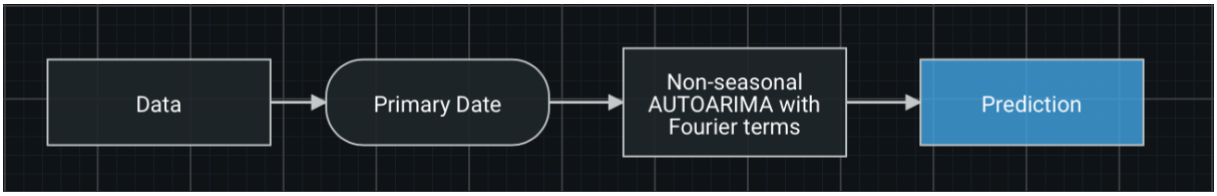


Ilustración 26. « Blueprint » o esquema del modelo "Non-seasonal AUTOARIMA with Fourier terms"

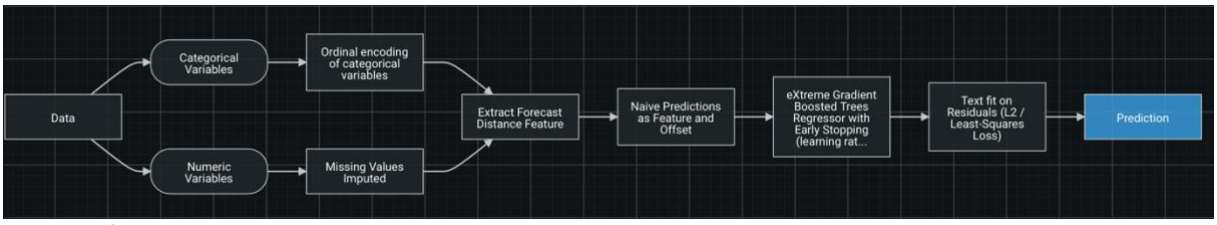


Ilustración 27. «Blueprint » o esquema del modelo "eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)"

También se puede usar un “Lift Chart” o Grafico de mejora respecto al modelo predictivo (Ilustración 28) para medir el cambio en términos de puntuación de la mejora respecto al modelo predictivo. Permite comparar modelos entre ellos u observar en qué puntos las predicciones del modelo se vuelven menos útiles. Además, puede determinar el punto en el que las predicciones del modelo se vuelven menos útiles. Un gráfico de beneficios es un tipo de gráfico relacionado que contiene la misma información que un gráfico de mejora respecto al modelo predictivo, pero que también muestra el aumento proyectado en los beneficios asociados al uso de cada modelo.

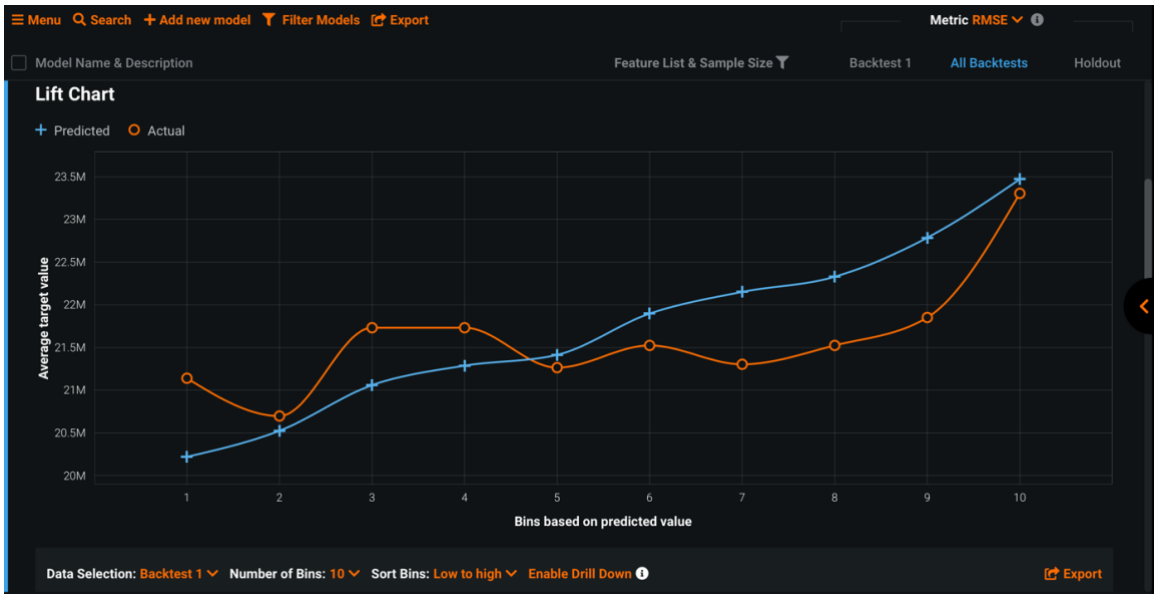


Ilustración 28. "Lift Chart" o Gráfico de mejora respecto al modelo predictivo en DataRobot

Para modelos basados en árboles de decisión como XGBoost, se puede observar la importancia de las variables según el modelo (Ilustración 29). En ese caso no nos interesa tanto dado que como se ha dicho anteriormente, con nuestro dataset utilizar un modelo tan complejo solo puede resultar en un modelo sobre ajustado. En el capítulo siguiente se realizarán predicciones con ambos modelos "Non-seasonal AUTOARIMA with Fourier terms" y “eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)” para demostrarlo.

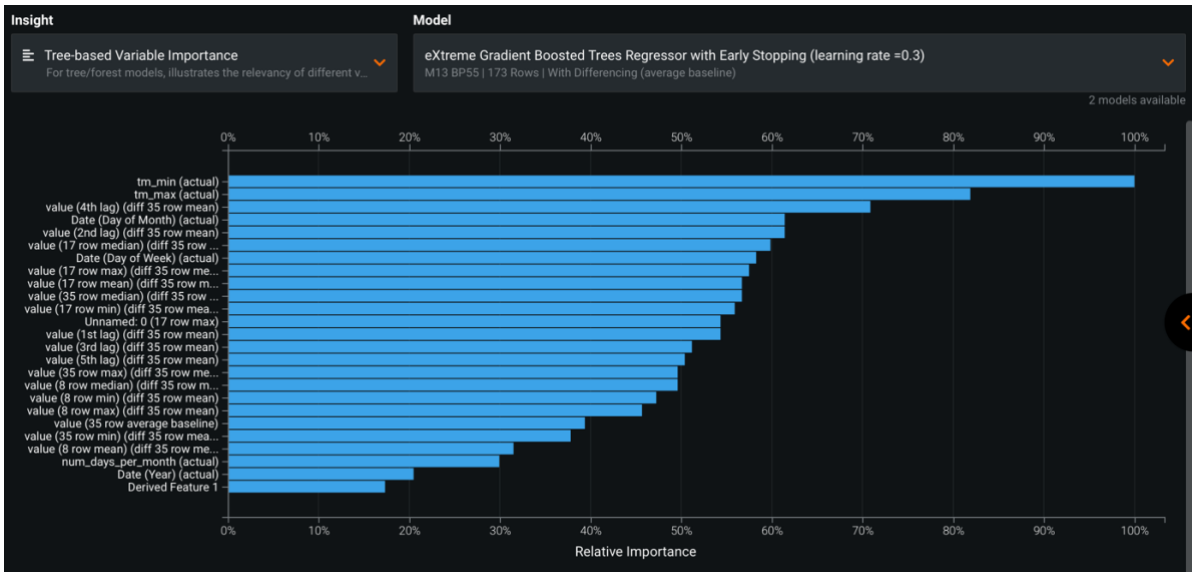


Ilustración 29. Importancia de las variables para los modelos de árboles de decisiones en DataRobot

A continuación, se realizarán predicciones de distintos modelos para poder comparar con los datos reales de abril y mayo los coeficientes de error de cada uno. Se trata de observar cual de los modelos entrenados predice mejor la evolución de la demanda eléctrica.

RESULTADOS: PREDICCIONES DEL MODELO

En este punto se realizarán predicciones de distintos modelos entrenados por DataRobot para comparar las métricas de optimización y observar algunas diferencias entre esas métricas y como de bien ajustan las predicciones. A continuación, se utilizan dos de los modelos vistos anteriormente: el modelo AUTOARIMA y XGBOOST. Se recuerda las métricas de optimización de dichos modelos en la Tabla 4. El dataset construido previamente utiliza los datos hasta marzo 2021. Para poder comprobar como de bien las predicciones aproximan los valores reales de la evolución de la demanda eléctrica, se utilizan los meses de abril y mayo para predecir y comparar las predicciones con los valores reales disponibles en la REE.

Tabla 4. Comparación de las métricas de optimización entre los modelos AUTOARIMA y XGBOOST

Modelo	RMSE	MAPE
Non-seasonal AUTOARIMA with Fourier terms	7.6634e+5	2.7539%
eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)	7.8103e+5	2.7985%

El modelo *Non-seasonal AUTOARIMA with Fourier terms* con un MAPE de 2.7539% es el que mejor acierta la realidad de todos los modelos que se han probados a lo largo del trabajo (Tabla 5). Tiene un coeficiente de error medio de 1.132% por lo cual se ha cumplido uno de los objetivos del trabajo, el cual es: entrenar un modelo que predice la evolución de la demanda eléctrica por debajo de tres porcientos de error.

Tabla 5. Predicciones de la demanda eléctrica del modelo AUTOARIMA para abril y mayo

Tiempo	Valor Real	Previsión	Coeficiente de Error
01/04/2021	19872225.296	19554797.874	1.623%
01/05/2021	20337879.18	20469048.164	0.641%

A pesar de que el modelo *eXtreme Gradient Boosted Trees Regressor with Early Stopping (learning rate =0.3)* tiene un MAPE muy similar al modelo AUTOARIMA el coeficiente de error es mucho mas grande: 5.6245%. Se puede explicar esa diferencia tan importante dado que un modelo XGBoosted es un modelo muy complejo que necesita muchos registros. En ese caso el dataset entrenado tiene 172 registros lo cual es muy poco para este tipo de modelo. Es la razón por la cual después de haber probado muchos modelos y sus predicciones, los modelos que mejor predicen la demanda eléctrica son modelos auto-regresivos, ARIMA o Auto-ARIMA cuyos modelos están explicados en el apartado *DataRobot*.

Tabla 6. Predicciones de la demanda eléctrica del modelo XGBOOST para abril y mayo

Tiempo	Valor Real	Previsión	Coeficiente de Error
01/04/2021	19872225.296	21598647.039	7.993%
01/05/2021	20337879.18	21022415.874	3.256%

Avanzando en nuestro razonamiento se trata de determinar si el modelo *Non-seasonal AUTOARIMA with Fourier terms* tiene valor. Considerando que tiene un MAPE de 2.7539%, el cual es más de la mitad del coeficiente de variación de la evolución de la demanda eléctrica (6,20%) y un coeficiente de error es de 1.132%, parece que ese modelo permite de acertar muy bien la predicción de la demanda eléctrica. Puesto que las predicciones son mensuales el modelo no permite operar directamente en el mercado eléctrico dado que el consumo es por hora. Lo que permite es planificar como un trader puede operar en el mercado eléctrico porque cuanto más adelanto tienen las predicciones, mejor va a poder negociar.
Anteriormente se han observado las predicciones de un modelo que tiene mucho potencial con un coeficiente de error muy bajo. En el próximo punto se buscará realizar un Dashboard mediante la herramienta Tableau para facilitar la interpretación de los datos y ayudar al equipo de Trading de Iberdrola que se encarga de operar en el mercado eléctrico, por ejemplo.

VISUALIZACION DE DATOS

En ese punto se quiere utilizar las métricas y los puntos de datos clave del modelo anterior y así permitir al de departamento de Trading de Iberdrola primeramente planificar operaciones en el mercado eléctrico y en un segundo tiempo operar. El Dashboard está conectado a varios ficheros de formato ".csv" que se actualizan a medida que los datos llegan. A continuación, desarrollamos el ejemplo de un Dashboard que podría ser transmitido al departamento de Trading de Iberdrola.

Se ha decido utilizar Tableau por varias razones. Dado que no se utiliza una base de datos muy grande y no se utilizan opciones muy avanzadas, no hay realmente una diferencia entre Tableau y Power BI. Dicho eso se elige Tableau para ilustrar el ejemplo porque tiene un despliegue más flexible y se puede instalar la versión de escritorio sin tener que instalar el servidor SQL. Aunqueel precio sea más alto, en este trabajo no lo tomamos en cuenta dado que será solo para ilustrar un ejemplo.

En la Ilustración 30 se puede observar un ejemplo de un Dashboard realizado para facilitar la planificación del departamento de Trading de Iberdrola. Este compuesto de varios indicadores que deben permitir al Trader de entender como de bien ha aproximado el modelo los valores de demanda de los meses pasados tanto como unos indicadores para la toma de decisión. Además de dar indicadores que permiten entender tanto la evolución de la demanda eléctrica como informaciones sobre el modelo, el Dashboard da indicaciones de lo que el modelo recomienda a un momento t para operar en el mercado. Citamos a continuación informaciones que aparecen en el Dashboard y cómo aparecen:

- Coeficiente de error entre el valor real de la evolución de la demanda y el previsto, para los últimos 12 meses (grafico de línea)

- Media del coeficiente de error (información)
- Datos históricos de la evolución de la demanda (grafico de línea)
- Predicciones de los dos próximos meses (información)
- Acción por realizar (comprar/vender) según la predicción realizada (información)
- Información del modelo (información)

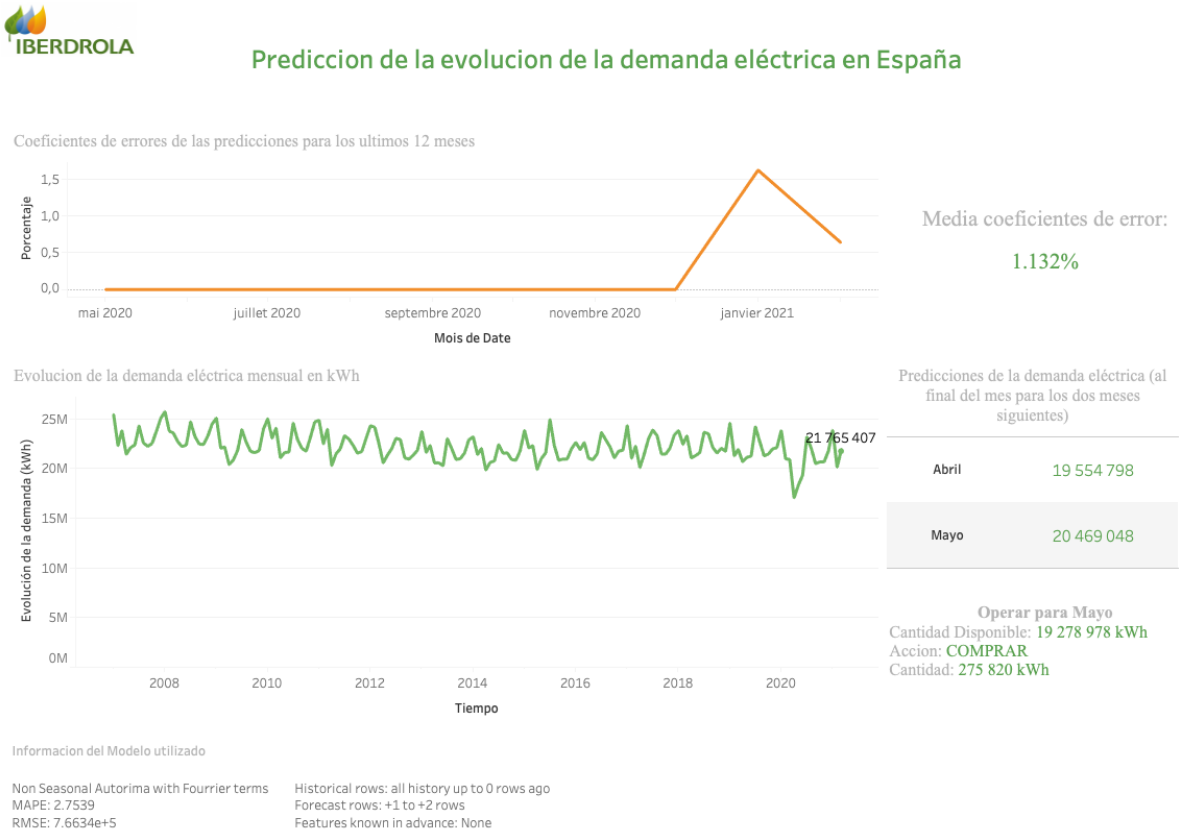


Ilustración 30. Dashboard, predicción de la evolución de la demanda eléctrica en España

El coeficiente de error entre el valor real de la evolución de la demanda y el previsto para los últimos 12 meses tanto como la media de los coeficientes de error permite, al momento de planificar/operar en el mercado, tomar en cuenta como ha sido acertando el modelo los últimos meses. Permite a la persona que toma decisiones relativizar sobre la predicción y así mejorar su decisión final. Los datos históricos de la evolución de la demanda eléctrica permiten tener referencias visuales de los meses de otros años. Puede resultar útil al Trader de comprobar como estaba la demanda eléctrica en los meses de los años anteriores y permite añadir otro indicador para mejorar la toma de decisión final. Las predicciones y la acción por realizar basado en las predicciones son los resultados directos del modelo y la información principal del Dashboard. Permite al Trader o de planificar o de operar en el caso de que fuera una predicción por hora o diaria. Finalmente se puede observar a bajo del Dashboard la información del Modelo para dar aún más información sobre cómo han sido realizadas las predicciones.

En el próximo capitulo vamos a repasar todos los modelos que se han entrenado a lo largo del trabajo con sus fuentes de datos. El modelo enseñado previamente es el que mejor ha funcionado, pero para llegar a un modelo “bueno” con datos públicos requiere entrenarlo sobre muchos datasets diferentes.

OTROS MODELOS QUE SE HAN PROBADO

Antes de llegar al modelo visto anteriormente, se ha entrenado un número importante de modelos con datasets muy diferentes y variables de entradas distintas. En es punto se tratará de ver la estrategia seguida para llegar a un modelo bueno y las diferentes variables de entrada usado a lo largo del trabajo.

Al momento de crear datasets se ha juntado indicadores económicos con importaciones de productos según industrias específicas para ver qué industria predecía mejor la demanda eléctrica. Otra estrategia ha sido utilizar el Índice Red Eléctrica (IRE) para predecir la evolución de la demanda eléctrica. Se trata de un indicador cuyo objetivo es facilitar una información adelantada de la evolución del consumo eléctrico del conjunto de empresas que tienen un consumo eléctrico medio/alto, así como su desglose por sectores de actividad (actividades industriales y de servicios). Se han realizado predicciones de ese indicador para utilizarlo como input en el modelo. Finalmente se han intentado predecir sectores específicos del IRE dado que se conoce el porcentaje de cada sector y que si el modelo aproxima bien se pueden usar las predicciones como input para predecir mejor la demanda eléctrica. El mejor resultado ha sido el modelo visto en el capítulo anterior dado que tiene un MAPE bajo y un coeficiente de error muy bajo.

Para adquirir datos de consumo/importación se ha utilizado la página web DataComex. Dado que esa página no deja descargar más de tres años de datos seguidos, se ha realizado un código en Python para automatizar la descarga de esos datos. Además, se han utilizado muchas páginas del gobierno para tener datos de indicadores económicos sobre España.

Finalmente adquirir datos, crear un dataset, entrenar un modelo y volver a empezar para intentar conseguir un modelo mejor es un proceso muy largo. Empieza con lo que se quiere predecir, después pensar en todos los datos que podrían tener una correlación con esa variable que se quiere predecir y cuando no ocurren más ideas buscar fuentes válidas para descargar esos datos. Se realiza siempre análisis exploratorios de los datos para asegurarse de que lo datos sean limpios, de que no haya outliers o missing values. Solamente después de esas etapas se entrena el modelo y se utilizan métricas de optimizaciones para compararlo con otros modelos. El último paso consiste en preguntarse qué se puede cambiar, añadir o quitar del dataset anterior para mejorar la predicción. Esta fase requiere conocimiento de cómo funcionan los diferentes tipos de modelos para poder reconocer según lo que se quiere predecir cuál de los modelos tiene más sentido entrenar.

LECCIONES APRENDIDAS DE DESARROLLO

En este punto se abordan y presentan las lecciones aprendidas del desarrollo realizado en este Trabajo Fin de Máster. A lo largo del trabajo se ha hablado de diferentes temas como adquisición de datos, modelos de aprendizaje supervisado o creación de datasets. A lo largo del trabajo se han extraído valiosas lecciones que se detallan a continuación.

Al momento de adquirir datos hay que ser creativo para descargar rangos de tiempos muy amplios. Muchas de las fuentes públicas no permiten a un usuario descargar datos por un periodo muy amplio y hay que utilizar técnicas vistas en el apartado Técnicas utilizadas para adquirir datos. Según lo que se quiere descargar la dificultad de lo que hay que hacer para adquirir el dato varía. La parte que consiste en adquirir los datos, limpiarlos y aplicar cambios para poder juntarlos es lo que más tiempo toma al momento de hacer modelos de ML.

Predecir la demanda eléctrica con datos públicos tiene ciertos límites. Para utilizar predicciones de la demanda eléctrica y operar en el mercado eléctrico se requiere predicciones horarias o diarios. Encontrar datos públicos en ese rango de tiempo es muy complicado dado que la mayor parte del tiempo se publican demasiado tarde para usarlos en un modelo o son datos privados. Por esas razones resulta difícil realizar esas predicciones horarios o diarios. Sin embargo, predecir la evolución de la demanda eléctrica a nivel mensual tiene mucho valor. Se pueden realizar modelos muy buenos con datos públicos dado que hay muchos más datos disponibles en ese rango de tiempo, pero las aplicaciones de esos modelos estarán más enfocadas a planificar operaciones que a operar en el mercado es diario.

Cuando un dataset no tiene muchos registros, ciertos modelos complejos como el XGBoost no pueden aprender de manera suficiente. Aunque las métricas de esos modelos puntúen alto, las predicciones no serán buenas y es necesario tener esto en cuenta. Requieren muchos registros para ser interesantes y en la situación abordada a lo largo de este trabajo no se encuentran accesibles, de forma pública, los datos de la evolución de la demanda eléctrica antes de 2007. Los 171 registros que se han utilizado para entrenar los modelos vistos anteriormente representan muy pocos datos. Dado que la relación entre los valores macroeconómicos y el consumo de electricidad es diferente antes y después de 2007, entrenar modelos con datos anteriores a ese periodo no resulta interesante. Otro factor es la crisis financiera de 2008 en la cual se observa una reducción de la producción industrial y la actividad del sector de servicios ha incrementado, excepto por Alemania que aumenta aproximadamente un uno por ciento su consumo de electricidad cada año.

En estos casos, trabajando con series temporales y un número de datos inferior a 600 registros, es una mejor opción el utilizar modelos como ARIMA o Auto-ARIMA, ya que éstos utilizan medias móviles o modelos auto-regresivos.

No se ha encontrado una correlación entre número de variables y la precisión de los modelos, por lo que los esfuerzos se han centrado en extraer datos que se complementan para la predicción de la variable de salida.

Aunque una herramienta de Auto-ML como DataRobot entrena diferentes tipos de modelos, es muy importante tener una idea lo que hace cada uno de ellos. Ya que, pese a que un modelo puede tener métricas muy buenas, sus predicciones no serán necesariamente igual de buenas. El rol del científico de datos es tener claras las características del dataset y de los modelos para entender los resultados y poder elegir el modelo que mejor resultados ofrecerá en un entorno real.

La implementación de un dashboard tiene mucho valor dado que permite integrar en la operativa de negocio el valor generado con los modelos. Esta integración mejora la toma

de decisiones y acorta tiempos de determinados procesos. Al momento de diseñar estos dashboards, además de dar una predicción o cualquier información clave, es importante pensar los diferentes indicadores que pueden complementar esa información para que la toma de decisión sea más completa y mejor.

CONCLUSIONES

El objetivo del trabajo es entender si un modelo realizado con autoML, sin conocimientos específicos del mercado eléctrico y con datos públicos, puede resultar de utilidad.

En ese trabajo, dado que se utilizan únicamente datos públicos resulta complicado predecir rangos de tiempos tan cortos. La información que se podría usar para realizar modelos por hora o día, o no está disponible al momento de entrenar el modelo, o no es accesible de forma pública. Sin embargo, predecir la evolución de la demanda eléctrica mensual tiene mucho valor, permite al departamento de Trading de Iberdrola, por ejemplo, planificar como va a operar en el mercado eléctrico según las predicciones de la demanda de los meses siguientes.

En ese trabajo se ha conseguido un modelo que adelanta la evolución de la demanda eléctrica mensual, con un MAPE de 2,75% y un coeficiente de error medio de 1,132%. Aunque no se puede usar ese modelo para operar directamente en los mercados diario o intradiario, tiene mucho valor utilizarlo para planificar operativas o inversiones y entender el mercado eléctrico. Cumpliéndose así el primer objetivo del trabajo y llegando a la conclusión de que puede resultar muy útil un modelo realizado con autoML, sin conocimientos específicos del mercado eléctrico y con datos públicos. Además, dado que el modelo tiene un MAPE por debajo de 3%, cumple el segundo objetivo del trabajo.

El análisis de las distintas herramientas de visualización de datos, con sus ventajas y desventajas ha permitido conocer cuál es mejor según en qué situación. En este trabajo se ha utilizado Tableau porque tiene un despliegue más flexible y se puede instalar la versión de escritorio sin tener que instalar el servidor SQL. La Ilustración 30 representa un ejemplo de cómo los datos podrían ser organizados para facilitar la planificación y la operación en el mercado eléctrico del equipo de Trading de Iberdrola. En la parte VISUALIZACION DE DATOS se ha estudiado en profundidad la productivización del modelo que se ha entrenado para facilitar los datos a otros departamentos de la empresa que lo pueden necesitar. Con esas partes se cumple el último objetivo de ese trabajo, que consistía en estudiar la puesta en producción de estos modelos, dando prioridad a las herramientas BI.

A pesar de que se han cumplido todos los objetivos del trabajo, queda la pregunta de si las herramientas de AutoML pueden remplazar los equipos de IA en las empresas, o entender mejor cuáles son sus límites. Dependerá realmente del tamaño de la empresa y del presupuesto que está dispuesto por poner en IA. Lo que sí permite concluir el resultado de este trabajo es que el avance de la tecnología y los algoritmos de predicción puede hacer que el conocimiento experto sea una barrera de entrada cada vez menor en ciertas situaciones, especialmente si es posible comprar bases de datos con información más precisa y actualizada que únicamente información pública como la utilizada en este trabajo.

Como se ha mencionado en la parte Un mercado creciendo y sus principales actores el mercado del AutoML está creciendo rápidamente medida que la tecnología se hace más popular. El informe de Research & Markets [15] indica que el mercado generado por este

tipo de herramientas fue de 300 millones de dólares en 2019 y se espera que aumente hasta los 14.500 millones de dólares en 2030. El autoML permite entrenar muchos modelos en poco tiempo, desplegarlos cuando se ha entrenado un modelo lo suficientemente bueno y no requiere mucho conocimiento de ML o de programación para usarlo.

Puede resultar muy útil a las empresas pequeñas o medianas que no quieren o no pueden invertir mucho dinero en IA pero que necesitan predicciones como, por ejemplo, reducir la pérdida de clientes o hacer previsiones de ventas. Dicho eso, hoy en día y para empresas como Iberdrola, el autoML tiene limitaciones de cara a desarrollar casos de IA más complejos. Tener los mejores modelos y desarrollar un departamento de IA en una empresa resulta una ventaja competitiva muy importante, ya sea para generar modelos por vías más tradicionales o para utilizar con mayor conocimiento estas herramientas de autoML.

LINEAS FUTURAS DE INVESTIGACIÓN

Para futuros estudios similares a éste se considera interesante la posibilidad de entrenar modelos utilizando datos privados de Iberdrola o cualquier empresa que disponga de este tipo de información de mercado. Se podría analizar de manera más concreta la importancia que tiene el acceso a información especializada, no pública. Dentro de esto, sería interesante observar las diferentes entre información no pública pero accesible para su compra e información diferencial que posea una empresa y no esté disponible para otros actores.

Se considera también implementar las predicciones de ese modelo como input para realizar otros modelos para entender mejor el mercado eléctrico. Se podría realizar un trabajo que planifica operativas o inversiones utilizando conocimiento del mercado eléctrico y nuevos modelos de predicciones. Se podría utilizar el modelo realizado en este trabajo juntado a conocimientos financieros para operar con contratos de compraventa de energía (o PPA por su siglo en ingles). Este tipo de contrato requiere determinar una fecha para la entrega de la electricidad, así que se usara modelos predictivos para determinar tanto la cantidad de electricidad intercambiada como el mejor momento para realizar la transacción y hacer beneficios. Hay muchas formas de PPA en uso hoy en día y que varían de acuerdo con las necesidades del comprador, el vendedor, y la financiación de las entidades de contrapartida.

BIBLIOGRAFÍA

[1] M. para la T. E. y el R. Demográfico, “Ministerio para la Transición Ecológica y el Reto Demográfico
,” 2021. [Online]. Available: https://energia.gob.es/electricidad/Paginas/sectorElectrico.aspx

[2] R. E. de España, “El suministro de la Electricidad.” Domèneche-learningmultimedia,S.A., 2009 [Online]. Available: www.ree.es

[3] C. J. Gallego and M. Victoria, “Entiende el mercado eléctrico. El observatorio crítico de la energía.,” 2012 [Online]. Available: http://observatoriocriticodelaenergia.org/files_download/Entiende_el_mercado_electrico.pdf

[4] eurowon, “Funcionamiento del mercado eléctrico en España,” 2010 [Online]. Available: https://www.eurowon.com/2010/06/funcionamiento-del-mercado-electrico-en.html

[5] E. y Sociedad, “Manual de la Energía ” [Online]. Available: http://www.energiaysociedad.es/manenergia/6-1-formacion-de-precios-en-el-mercado-mayorista-diario-de-electricidad/

[6] M. John, “WHAT IS ARTIFICIAL INTELLIGENCE?,” 2014 [Online]. Available: https://homes.di.unimi.it/borghese/Teaching/AdvancedIntelligentSystems/Old/IntelligentSystems_2008_2009/Old/IntelligentSystems_2005_2006/Documents/Symbolic/04_McCarthy_whatisai.pdf

[7] C. François, *Deep Learning with Python*. United States of America: Manning Publications Co., 2018.

[8] I. E. Agency, “Digitalisation and Energy,” Nov. 2017 [Online]. Available: https://www.iea.org/reports/digitalisation-and-energy

[9] energyfactor, “A collaboration in curiosity: ExxonMobil and MIT explore the oceans,” 2016 [Online]. Available: https://energyfactor.exxonmobil.com/energy-innovation/collaborations/mit-collaboration/

[10] G. Bill, “Bill Gates has a message for every college grad who wants to change the world,” 2017 [Online]. Available: https://www.mic.com/articles/176935/bill-gates-has-a-message-for-every-college-grad-who-wants-to-change-the-world

[11] Iberdrola, “Iberdrola presenta sus principales datos operativos a cierre del Primer trimestre 2021 ,” 2021 [Online]. Available: https://www.iberdrola.com/conocenos/cifras/principales-datos-operativos

[12] Iberdrola, “Descubre los principales beneficios del ‘Machine Learning,’” 2021 [Online]. Available: https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico

[13] M. Vishal and A. S. Venkat, “XGBoost Algorithm: Long May She Reign!,” 2019 [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[14] DataRobot, “Automated Machine Learning,” 2021 [Online]. Available: https://www.datarobot.com/wiki/automated-machine-learning/

[15] ResearchAndMarkets, “ Automated Machine Learning Market Research Report: By Offering, Deployment Type, Enterprise Size, Application, Industry - Industry Size, Share, Development and Demand Forecast to 2030,” 2020 [Online]. Available: https://www.businesswire.com/news/home/20200408005248/en/Insights-Worldwide-Automated-Machine-Learning-Market-2020

[16] D. Cem, “AutoML Tech / Products of 2021 Compared: in-Depth Guide,” 2021 [Online]. Available: https://research.aimultiple.com/automl-comparison/

[17] DataRobot, “DataRobot Wiki,” 2021 [Online]. Available: https://www.datarobot.com/wiki/supervised-machine-learning/

[18] DataRobot, “Time Series Modeling,” 2021. [Online]. Available: https://university.datarobot.com/path/time-series-foundation-quest/time-series-modeling/598117/scorm/b2yhy8edtmzx

[19] R. E. de España, “REData API.” [Online]. Available: https://www.ree.es/es/apidatos

[20] A. E. de Meteorología, “AEMET OpenData,” 2021. [Online]. Available: http://www.aemet.es/es/datos_abiertos/AEMET_OpenData

ANEXOS

Descarga de la evolución de la demanda eléctrica.

In [ ]:

```
import requests
import pandas as pd
import json
from pandas import ExcelWriter
import numpy as np
import matplotlib
import csv
from datetime import datetime
from datetime import date
#import datetime
from datetime import timedelta
import time
from IPython.display import display
from IPython.display import Image
import os
```

In [ ]:

```
#Latest One For Date Range
list_day = []
#for i in range(2007,2021,2):
for i in range(2010,2021,2):
    start_year = date(i,1,1)
    list_day.append(start_year)
    end_year = date(i+1,12,31)
    list_day.append(end_year)
    #print(start_year)
    #print(end_year)

list_day.append(date(2021,1,1))
list_day.append(date(2021,5,31))
list_day
```

In [ ]:

```
#Check that we have the right dates
for i in range(len(list_day)-1,0,-2):
```

```
    print(list_day[i-1])
    print(list_day[i])
```

In [ ]:

```
datos = []
for i in range(len(list_day)-1,0,-2):
    #url para utilizar la API de REE
    url='https://apidatos.ree.es'

    # Parametros principales de la API (lang, category & widget)
    lang = 'es'
    category = 'demanda'
    widget = 'evolucion'

    # url a añadir para acceder a los datos segun la category y el widget
    url_ext = f'/{lang}/datos/{category}/{widget}?query'

    # Parametros para definir el rango de tiempo que se va a descargar + nacional o por region precisas
    #start_date = f'start_date=2021-03-12T00:00'
    start_date = f'start_date={list_day[i-1]}'
    start_date
    #end_date = 'end_date=2021-03-15T00:00'
    end_date = f'end_date={list_day[i]}'
    end_date
    time_trunc = 'time_trunc=month'
    #geo_limit = 'geo_limit='
    #geo_ids = 'geo_ids=8'

    # url que añade los ultimos parametros
    url_params = f'&{start_date}&{end_date}&{time_trunc}'
    #url_params = f'&{start_date}&{end_date}&{time_trunc}&{geo_ids}'

    response = requests.get(url+url_ext+url_params)
    data = json.loads(response.content.decode(response.encoding))
    print(data)
    # Se descarga la parte del json que contiene la los valores de la demanda
    #df2 = pd.DataFrame.from_dict(data['included'][0]['attributes']['values'])
    df2 = pd.DataFrame.from_dict(data['included'][0]['attributes']['values'])

    #df2.groupby([time.dt.day, times.dt.hour, times.dt.minute]).value.sum()
```

```
df2['datetime'] = pd.to_datetime(df2['datetime'], format='%Y-%m-%dT%H:%M:%S')
# Se crea una columna para la fecha y otra para el hora
df2['Date'], df2['Hour']= df2['datetime'].apply(lambda x:x.date()), df2['datetime'].apply(lambda x:x.time())

# Poner ahora en string, quitar despues de doce y agrupar por fecha y hora
# agrupar por date y hora. Luego hacer pivot hora

#df2.info()
#df2.shape

# Quitamos las columnas que no nos interesan
df_clean = df2.drop(['datetime'],axis=1)
df_clean = df_clean.drop(['percentage'], axis=1)
df_clean = df_clean.drop(['Hour'], axis=1)
#df_clean = df_clean[df_clean['Date']>date(2013,12,31)]

# Poner ahora en string, quitar despues de doce y agrupar por fecha y hora
#df_clean.Hour = df_clean.Hour.astype(str)
#df_clean['Hour'] = df_clean['Hour'].str[:2]
#df_clean.Hour = df_clean.Hour.astype(int)
#df_clean_grouped = df_clean.groupby(['Date','Hour'])['value'].sum()
df_clean_grouped = df_clean.groupby(['Date'])['value'].sum()
df_clean_grouped.head(30)

datos.append(df_clean_grouped)
time.sleep(3)
```

In [ ]:

```
df_concat = pd.concat(datos).to_frame()
df_concat_reset_index = df_concat.reset_index()
df_concat_reset_index = df_concat_reset_index.drop_duplicates(subset=['Date'], keep='last')
df_concat_reset_index['Date'].duplicated().any()
```

In [ ]:

```
df_concat_reset_index.tail()
```

In [ ]:

```
df_demanda_final = df_concat_reset_index.sort_values(by='Date')
df_demanda_final = df_demanda_final.reset_index(drop=True)
```

In [ ]:

```
#df_demanda_final[(df_demanda_final['value'] < 2.000000e+07)]
```

In [ ]:

```
df_demanda_final.tail()
```

In [ ]:

```
df_demanda_final.to_csv('/Users/thomasbustos/Desktop/Iberdrola/datos_last_version/datos/0_DATASETS/demand
a_final_with_april_and_may.csv',sep=';')
```

Descarga de las temperaturas medias mensuales de las mínimas y máximas

In [ ]:

```
import requests
import pandas as pd
from pandas import ExcelWriter
import json
import numpy as np
import matplotlib
import csv
import datetime
from datetime import datetime
from datetime import timedelta
from datetime import date
import time
from IPython.display import display
from IPython.display import Image
import os
```

In [ ]:

```
# Importar datos de temperaturas desde la AEMET
filename='apikey.txt'
f = open(filename, 'r')
api_key = f.read()
f.close()
#api_key
```

In [ ]:

```
idema = '3129'
```

In [ ]:

```
datos_temp_idema = []
#for i in range(len(idema)):
querystring = {'api_key':api_key}
anioIniStr = '2006-1'
#fechaFinStr = '2021-03-17T23:59:59UTC'
anioFinStr = '2021-6'
url2 = "https://opendata.aemet.es/opendata/api/valores/climatologicos/mensualesanuales/datos/"
url_ext2 = f'anioini/{anioIniStr}/aniofin/{anioFinStr}/estacion/3129'

response2 = requests.get(url2+url_ext2,params=querystring)
#print(response2.json())

acceso_datos = response2.json()
acceso_datos
#print('Descarga para ' + idema + ':' + acceso_datos['descripcion'])
resultados2 = requests.get(acceso_datos['datos'])
#print(resultados2)
datos_aemet_temp=pd.DataFrame(resultados2.json())
```

Notes
tm_max : Temperatura media mensual/anual de las máximas (grados celsius)
tm_min : Temperatura media mensual/anual de las mínimas (grados celsius)

In [ ]:

```
df_filtered = datos_aemet_temp.loc[datos_aemet_temp['indicativo'] == '3129']
df_filtered = df_filtered[['fecha','tm_min','tm_mes','tm_max']]
df_filtered[['year','month']] = df_filtered.fecha.str.split("-",expand=True,)
df_filtered['day'] = 1
df_filtered[['year','month','day']] = df_filtered[['year','month','day']].astype(str).astype(int)
#isdigit()
df_filtered = df_filtered[~df_filtered.fecha.str.contains('-13')]
df_filtered['Date'] = [date(year=x[1].year, month=x[1].month, day=x[1].day) for x in df_filtered.iterrows()]
df_filtered = df_filtered.drop('fecha',axis=1)
df_filtered = df_filtered.drop('year',axis=1)
df_filtered = df_filtered.drop('month',axis=1)
df_filtered = df_filtered.drop('day',axis=1)
df_filtered['Date'] = pd.to_datetime(df_filtered['Date'])
df_filtered = df_filtered[['Date','tm_min','tm_mes','tm_max']]
```

In [ ]:

```
df_filtered.tail(12)
```

In [ ]:

```
df_filtered.describe()
```

In [ ]:

```
df_filtered.dtypes
```

In [ ]:

```
df_filtered.to_csv('/Users/thomasbustos/Desktop/Iberdrola/datos_last_version/datos/temperaturas_mad_media_min_med_max_to_april_and_may.csv',sep=';')
```

Creación del dataset mediante Python

In [ ]:

```
import pandas as pd
import os
import datetime
from datetime import datetime
from datetime import date
import calendar
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
from plotnine import ggplot, aes, geom_line
from workalendar.europe import Spain
```

In [ ]:

```
os.chdir('/Users/thomasbustos/Desktop/Iberdrola/datos_last_version/datos')
```

Datos Demanda

In [ ]:

```
DATASET_DEMANDA = pd.read_csv('0_DATASETS/DATASET_DEMANDA_CORRECTED_2007_2021.csv',sep=';')
DATASET_DEMANDA = DATASET_DEMANDA.drop(DATASET_DEMANDA.columns[0],axis=1)
DATASET_DEMANDA['value'] = DATASET_DEMANDA['value'].drop(DATASET_DEMANDA['value'].index[171])
#predict_dates = ['2021-05-01','2021-06-01']
```

```
#DATASET_DEMANDA = DATASET_DEMANDA.append(pd.DataFrame( predict_dates, columns=[ 'Date']),ignore_index = True)

DATASET_DEMANDA['Date'] = pd.to_datetime(DATASET_DEMANDA['Date'])

DATASET_DEMANDA.tail()
```

In [ ]:

```
DATASET_DEMANDA.plot(x ='Date', y='value', kind = 'line')
```

# of days per month

In [ ]:

```
number_days_per_month = pd.DataFrame()

number_days_per_month['Date'] = DATASET_DEMANDA['Date']

number_days_per_month["num_days_per_month"] = pd.to_datetime(DATASET_DEMANDA['Date'])

number_days_per_month["num_days_per_month"] = pd.to_datetime(number_days_per_month["num_days_per_month"])

number_days_per_month["num_days_per_month"] = number_days_per_month["num_days_per_month"].dt.days_in_month

number_days_per_month.tail()
```

# of festive days

Utilisamos la libreria workalendar para hacer un conteo de los dias festivos de cada mes segun el año.

In [ ]:

```
def Extract(lst):
    return [item[0] for item in lst]
```

In [ ]:

```
#festive_days = pd.DataFrame()

#festive_days['Date'] = DATASET_DEMANDA['Date']

#festive_days["num_festive_days_per_month"] = pd.to_datetime(festive_days['Date'])
```

In [ ]:

```
cal = Spain()

list_festive_days = []

for i in range(2007,2022,1):

    festive_days_of_the_year = cal.holidays(i)

    list_festive_days_for_the_year = Extract(festive_days_of_the_year)

    dataframe = pd.DataFrame(list_festive_days_for_the_year, columns = ['date_festive_day'])
```

```
    list_festive_days.append(dataframe)


festive_days = pd.concat(list_festive_days)
festive_days['date_festive_day'].duplicated().any() # Comprobamos que no haya duplicados
festive_days['date_festive_day'] = festive_days['date_festive_day'].astype(str)
festive_days[['year','month','day']] = festive_days.date_festive_day.str.split("-",expand=True,)
festive_days['day'] = 1
festive_days[['year','month','day']] = festive_days[['year','month','day']].astype(str).astype(int)
festive_days['Date'] = [date(year=x[1].year, month=x[1].month, day=x[1].day) for x in festive_days.iterrows()]
festive_days = festive_days.drop('date_festive_day',axis=1)
festive_days = festive_days.drop('year',axis=1)
festive_days = festive_days.drop('month',axis=1)
festive_days = festive_days.drop('day',axis=1)
festive_days['Date'] = pd.to_datetime(festive_days['Date'])
festive_days['festive_day'] = 1


festive_days = pd.DataFrame(festive_days['festive_day'].groupby(festive_days['Date'].dt.to_period('M')).sum())
festive_days.rename(columns={'festive_day': 'festive_days_count'}, inplace=True)
festive_days = festive_days.reset_index()
festive_days['Date'] = festive_days['Date'].astype(str) + '-01'
festive_days['Date'] = pd.to_datetime(festive_days['Date'])


festive_days.head()
#festive_days.dtypes
#xyz = df_concat['dias_festivos'].tolist()
```

In [ ]:

```
#groupby_df = df_concat['date_festive_day,year'].groupby([df_concat.date_festive_day.dt.year, df_concat.date_festi
ve_day.dt.month]).agg('count')
#groupby_df
#dias_festivos_españa = pd.DataFrame(groupby_df)
#dias_festivos_españa.rename(columns={ dias_festivos_españa.columns[0]: "count" }, inplace = True)
#dias_festivos_españa.reset_index(inplace = True)
#dias_festivos_españa.reset_index(level=0, drop=True).reset_index()
#dias_festivos_españa


#df_concat['dias_festivos'].groupby()
#x = groupby_df.reset_index(name='count')



#dias_festivos_españa['day'] = 1
```

```
#dias_festivos_españa['month'] = dias_festivos_españa["dias_festivos"]
#dias_festivos_españa.head(20)


#lista_dias_festivos_españa = dias_festivos_españa.dias_festivos.tolist()
#dias_festivos_españa = dias_festivos_españa.reset_index()
```

In [ ]:

```
#festive_days = pd.DataFrame()
#festive_days['Date'] = DATASET_DEMANDA['Date']
#festive_days['dias_festivos'] = 0
#festive_days["dias_festivos"] = lista_dias_festivos_españa


#festive_days.tail()
```

# of saturdays and sundays

The dataset goes from 2007-01-01 to 2021-04-01
To improve this variable we could add the festive days but I haven't found a way to do so easily for each year.
Let's use for the moment saturdays and sundays

In [ ]:

```
sat_sund_counts = pd.DataFrame()
sat_sund_counts['Date'] = DATASET_DEMANDA['Date']
sat_sund_counts.Date.dt.month
sat_sund_counts['Year'] = sat_sund_counts.Date.dt.year
sat_sund_counts['Month'] = sat_sund_counts.Date.dt.month
#sat_sund_counts
year_list = sat_sund_counts['Year'].values.tolist()
month_list = sat_sund_counts['Month'].values.tolist()
```

In [ ]:

```
num_saturdays =[]
num_sundays = []
for i in zip(year_list, month_list):
    year = i[0]
    month = i[1]
    #Saturday
    day_to_count = calendar.SATURDAY
    matrix = calendar.monthcalendar(year,month)
```

```
    num_days_saturday = sum(1 for x in matrix if x[day_to_count] != 0)
    num_saturdays.append(num_days_saturday)
    #Sunday
    day_to_count = calendar.SUNDAY
    matrix = calendar.monthcalendar(year,month)
    num_days_sunday = sum(1 for x in matrix if x[day_to_count] != 0)
    num_sundays.append(num_days_sunday)

num_saturday_sunday_each_month = c = [x+y for x,y in zip(num_saturdays, num_sundays)]
#num_saturday_sunday_each_month
```

In [ ]:

```
sat_sund_counts['saturday_sundays_count'] = num_saturday_sunday_each_month
sat_sund_counts = sat_sund_counts.drop('Year',axis=1)
sat_sund_counts = sat_sund_counts.drop('Month',axis=1)
sat_sund_counts.tail()
```

Temperatura media min & max

In [ ]:

```
temperaturas = pd.read_csv('temperaturas_mad_media_min_med_max.csv',sep=';')
temperaturas = temperaturas.drop('Unnamed: 0',axis=1)
temperaturas = temperaturas.loc[(temperaturas['Date'] < '2021-04-01')]
predict_dates = ['2021-04-01']
temperaturas = temperaturas.append(pd.DataFrame(predict_dates, columns=[ 'Date']),ignore_index = True)
temperaturas['Date'] = pd.to_datetime(temperaturas['Date'])
temperaturas.tail()
```

Covid

In [ ]:

```
covid = pd.DataFrame()
covid['Date'] = DATASET_DEMANDA['Date']
# marzo 2020 hasta el plan de deconfinamiento:
#Fase 1: inicial (11 de mayo)
#Fase 2: intermedia
#Fase 3: avanzada
#21 de junio, con el final de la última prórroga del estado de alarma
covid['covid']=0
covid['covid'].loc[(covid['Date'] >= '2020-03-01')] = 1
#covid.tail()
```

Merging DATA !

In [ ]:

```
spmerge1 = pd.merge(DATASET_DEMANDA,number_days_per_month,how='left',on ='Date')
spmerge2 = pd.merge(spmerge1,sat_sund_counts,how='left',on ='Date')
spmerge3 = pd.merge(spmerge2,temperaturas,how='left',on ='Date')
spmerge4 = pd.merge(spmerge3,festive_days,how='left',on ='Date')
spmerge5 = pd.merge(spmerge4,covid,how='left',on ='Date')
```

In [ ]:

```
spmerge5['festive_days_count'] = spmerge5['festive_days_count'].fillna(0) # Se cambia los NaN por 0 dias festivos
spmerge5 = spmerge5.drop(spmerge5.index[[171]]) #Se entrena el modelo con datos hasta marzo para predecir los meses siguientes y poder comprobar si el modelo acierta la realidad
spmerge5['Date'] = pd.to_datetime(spmerge5['Date'])
spmerge5.tail()
```

In [ ]:

```
corr_df = spmerge5.corr(method='pearson')
#plt.matshow(corr_df)
plt.figure(figsize=(8, 6))
sns.heatmap(corr_df, annot=True)
plt.show()
```

We can observe on the correlation matrix that the three temperature are very correlated. We'll remove the tm_mes column to decrease this correlation. We'll be trying different models combine those columns to see which combination allows us to predict better the electricity demand.

In [ ]:

```
spmerge5 = spmerge5.drop('tm_mes',axis=1)
```

In [ ]:

```
spmerge5.tail()
```

In [ ]:

```
# Dataset
spmerge5.to_csv('/Users/thomasbustos/Desktop/Iberdrola/datos_last_version/datos/0_DATASETS/modelos_eugenio/dataset_elecricity_demand_LAST.csv',sep=";")
```

In [ ]:

```
ls 0_DATASETS/modelos_eugenio/
```

In [ ]:

```
spmerge5 = pd.read_csv('0_DATASETS/modelos_eugenio/dataset_elecricity_demand_LAST.csv',sep=';')
spmerge5 = spmerge5.drop(spmerge5.columns[0],axis=1)
#spmerge4 = spmerge4.loc[(spmerge4['Date'] < '01/04/2021')]
```

In [ ]:

```
spmerge5.tail()
```

Exploratory Analisis

In [ ]:

```
spmerge5['Date'] = pd.to_datetime(spmerge5['Date'])
spmerge5.plot(x ='Date', y='value', kind = 'line')
```

In [ ]:

```
spmerge5.describe()
```

In [ ]:

```
spmerge5.dtypes
```

In [ ]:

```
corr_df = spmerge5.corr(method='pearson')
#plt.matshow(corr_df)
plt.figure(figsize=(8, 6))
sns.heatmap(corr_df, annot=True)
plt.show()
```