ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA

PROYECTO DE OPTIMIZACION DE SERVICIOS Y VENTAS MEDIANTE MACHINE LEARNING

Autor: Ander Gurtubay Regúlez

Director: Carlos Luis Blanco Jabares

Madrid

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

PROYECTO DE OPTIMIZACION DE SERVICIOS Y VENTAS MEDIANTE MACHINE LEARNING

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

curso académico 2018/19 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

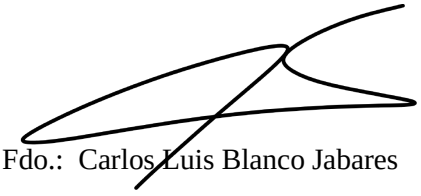
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Ander Gurtubay Regúlez Fecha: 18/ 06/ 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Carlos Luis Blanco Jabares Fecha: 22/ 06/2021

Vº Bº del Coordinador de Proyectos

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

Índice de la memoria

Índice de figuras

Índice de gráficas

Índice de tablas

Índice de ecuaciones

Agradecimientos

En primer lugar, me gustaría comenzar por agradecer a PwC por la enorme oportunidad que ha supuesto formar parte de este proyecto estos seis meses y por la confianza que han depositado al dejarme participar de forma tan activa en este proyecto. Además, agradécele a toda la gente que ha formado parte del proyecto y que me ha aguantado y me ha ayudado a sentirme como en casa cada día, pero sobre todo a mi tutor de este TFM, Carlos Blanco Jabares, y, en especial, a Damián Álvarez Piqueras que ha sido el encargado de guiarme y el que ha tenido que aguantar toda mis "pifias" y errores con mucha paciencia.

A Rocío, porque su forma de querer y su personalidad son únicas hasta un punto que no se imagina, por aguantarme, por apoyarme, por quererme como lo hace, porque, aunque no lo crea, es la base en la que me apoyo en mi día a día.

A mi familia, a mis padres por su apoyo y amor incondicional, quienes me ha hecho convertirme en lo que soy, son grandes responsable de que esté aquí y que haya logrado todo lo que me he propuesto. A mi hermano, por hacerme ser mejor persona y esforzarme por ser un ejemplo. Gracias y perdón por las veces que no exprese lo que sentía.

A mis amigos, David, Nano, Calleja, Adiran y Alonso, por ser la familia que yo elegí, que han estado presentes durante toda mi vida, creciendo juntos no solo en tamaño, sino como personas.

A la universidad y a todos los profesores que me han formado durante este ultimo año para convertirme en una mejor posible y, sobre todo, un mejor profesional, y que han hecho posible esto.

En general, a todo el mundo que ha formado parte de este año de recuerdos felices e imborrables.

Gracias a todos.

1. Introducción

En el presente trabajo de fin de máster se va a realizar el informe sobre todo el proceso de ejecución del proyecto que he realizado durante mis prácticas de formación en PwC. Dicho proyecto pretende poner en práctica los conocimientos y recursos que se han impartido durante este curso en la Universidad Pontificia de Comillas, a la vez que se adquiere experiencia profesional al participar en un proyecto real para una empresa, afrontando los desafíos que esto supone.

Dicho proyecto se trata de un proyecto interno al que me incorpore el pasado 18 de enero y que surge con la intención por parte de la empresa de optimizar los datos internos sobre los diferentes cliente, los proyectos que realizan con cada uno de ellos, el sector al que pertenece el cliente y el proyecto y los ingresos que estos generan, entre otros, para poder anticipar el comportamiento del mercado y de los clientes, pudiendo así realizar un enfoque más eficiente de estos, optimizando los recursos de la empresa y maximizando los beneficios y asegurando la continuidad de dichos clientes.

1.1 Motivación

Este proyecto surge como un proyecto interno de la compañía, dicha compañía es una multinacional que ofrece servicios profesionales a terceros y como tal, surge la necesidad de comprender a cada uno de sus clientes, el mercado en el que se engloban estos y que características similares puede encontrar en cada uno de sus clientes en esos mercados, para poder así, elaborar una estrategia eficaz y única para cada uno de esos clientes, basada en una serie de modelos que permita optimizar la rentabilidad y asegurar la continuidad de los clientes.

Este proyecto es uno de los pocos proyectos que ha realizado la empresa y que no pretende facilitar otro de los proyectos en algún departamento, sino que busca optimizar la forma en la que PwC afronta la estrategia comercial y los estudios que se realizan de clientes y proyectos. Como tal, no existe un gran número de proyectos que puedan servir de antecedente a este, aun así, las herramientas empleadas, así como los modelos de predicción son bien conocidas en el departamento y se han utilizado en muchos otros proyectos externos.

En resumen, este proyecto pretende apoyarse en datos internos de la compañía, así como de proveedores externos para desarrollar una serie de modelos para poder evaluar y clusterizar los clientes actuales y objetivo ofreciendo a la compañía y a cada una de sus secciones y departamentos una visión más amplia y precisa del mercado, con la que poder ser capaces de disponer de una información más precisa y valiosa a la hora de afrontar las diferentes propuestas, permitiendo diseñar un mix de proyecto y el orden de ejecución en base a dichos modelos.

1.2 Objetivos

El objetivo principal de este trabajo es diseñar una forma eficaz de anticipar el comportamiento del mercado sector de objeto de estudio, para poder así crear la mejor estrategia comercial que permita maximizar la rentabilidad y asegurar la continuidad de los clientes,

mediante el estudio de dichos clientes que permita anticipar en que proyectos pueden estar interesados y en qué orden se deben ejecutar dichos proyectos para maximizar las probabilidades de éxito. Para lograr este objetivo global se han determinado los siguientes objetivos principales:

- Crear una base de datos con las variables necesarias para cada uno de los objetivos que se van a exponer a continuación, depurando los datos internos de la compañía para ser capaces de obtener el máximo rendimiento en los modelos que se van a diseñar a partir de estos.
- Diseñar una forma eficiente de agrupar a nuestros clientes en base a sus características y a las variables internas de la compañía, para poder establecer las diferentes conexiones entre ellos y entre los diferentes mercados.
- Crear una forma eficaz de interpretar dichas conexiones para ser capaces de establecer el mix de proyecto óptimo para cliente, siendo capaces de optimizar los proyectos ya propuestos y de proponer nuevos proyectos a cada uno de nuestros clientes.
- Modelar una forma eficaz de analizar el rendimiento de la empresa con el paso del tiempo, siendo capaces de monitorizar el desempeño de la compañía semana a semana y pudiendo así observar el impacto que generan los diferentes sectores y proyectos en los beneficios de la compañía.
- Evaluar el rendimiento de los diferentes algoritmos, para ver si son capaces de cumplir los objetivos individuales propuestos y si en conjunto, cumplen el objetivo global marcado para el proyecto.

2. Metodología de Trabajo

En primer lugar, se realizará un preprocesado de los datos disponibles, para poder disponer estos de tal manera que sea posible utilizarlos como base para los modelos analíticos que desarrollaremos para alcanzar los diferentes objetivos del proyecto. Se deberán crear diferentes tablas de datos para cada uno de los modelos que deseamos realizar, siendo la selección adecuada de variables el mayor y el depurado de los datos los mayores retos que suponen esta primera fase que puede resultar critica para el devenir del proyecto.

Una vez hayamos obtenido nuestros datos históricos, tanto de los diferentes clientes como de la compañía pasaremos a desarrollar varios modelos de machine learning que permitan cumplir los objetivos ya mencionados.

El primero de estos modelos será un clustering de los diferentes clientes de la empresa, un modelo de aprendizaje no supervisado, que nos permitirá obtener las componentes principales de nuestra base de datos, criticas en el análisis de mercado y que pueden dar una nueva visión en la forma en la que se relacionan los clientes. Además, deberemos determinar el número óptimo de clúster de los que dispondrá nuestro modelo, para poder así agrupar a nuestros clientes y llevará a cabo el análisis de los resultados estudiando las diferencias entre los diferentes clústeres.

Una vez hayamos comprendido de manera eficaz las relaciones y características principales de los clientes, podremos diseñar un modelos de recomendado de proyectos, que se base en la nueva visión de cliente que nos ofrece el clustering previo, para establecer conexiones entre proyectos y clientes que permita recomendar nuevos proyectos de forma personal basándose en el histórico de proyectos, en los proyectos de clientes con características similares y en el grupo o mercado que se engloba dicho cliente al que queremos recomendar nuevos proyectos.

Por último, desarrollaremos una serie temporal de previsión de ventas de la compañía y por sector, de esta forma podremos encontrar las tendencias y las estacionalidades de dichas ventas y ser capaces de predecir cual debería ser el valor de las ventas a futuro para poder así, evaluar el rendimiento de la compañía y de los sectores, observando el impacto de este proyecto en general en las ventas y monitorizando semanalmente el desempeño de la empresa y su proyección a futuro.

La combinación de técnicas y modelos mencionados permitirá a PwC lograr los objetivos ya mencionados, siendo capaz de optimizar sus recursos y maximizar beneficios mediante el aprovechamiento de los datos internos y el estudio de sus clientes y las relaciones que existen entre estos.

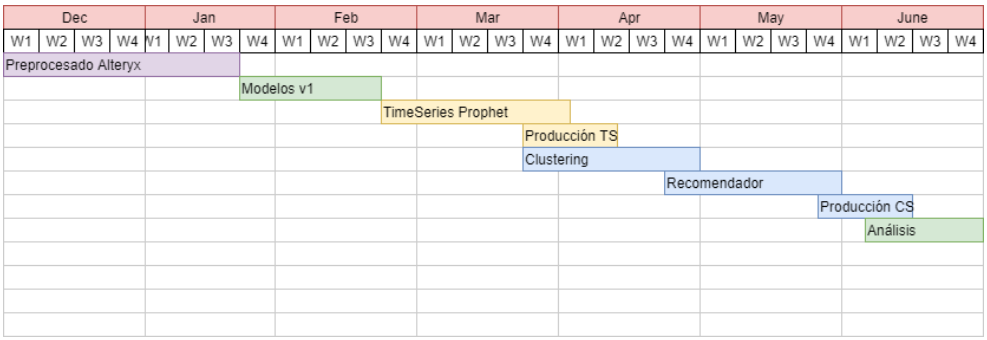


Figura 2.1: Cronograma del proyecto

Como observamos en la Figura 2.1, la duración estimada del proyecto será de unos 6 meses, dividiéndose este en tres partes claramente diferenciadas: una primera parte de preprocesado de datos (morado), en la que se realizará la extracción de la base de datos ya comentada, preparando dichos datos para cada modelo, eliminando aquellos datos que no sean útiles y solucionando los datos que puedan surgir con dichos datos y con la estructura de la base de datos. Un primer modelo de series temporales para la predicción de las ventas de la compañía y de cada uno de los sectores de esta (amarillo) y su posterior puesta en producción. Por último, un segundo modelo de clustering y recomendación de proyectos a los diferentes clientes de la compañía (azul) y su posterior puesta en producción.

De esta forma la primera etapa de extracción no debería de alargarse más de dos meses, siendo la duración estimada de 7 semanas. Una vez realizada dicha extracción, se realizará un proceso de prueba en el que se intentará desarrollar una primera versión de los dos modelos, aunque estos no estén optimizados ni sus resultados sean óptimos, se intentará alcanzar un producto mínimo funcional.

Posteriormente, se comenzará con la serie temporal, al ser el modelo que parece haber despertado un mayor interés entre los diferentes interesados del proyecto, siendo la duración estimada de su optimización y puesta en producción de 7 semanas, casi dos meses.

Por último, se comenzará con el clustering de clientes y el recomendador, el último modelo y fase del proyecto siendo la duración estimada de esta fase de unos dos meses y medio, unas 10 semanas.

Además, al finalizar el proyecto, se reservan un par de semanas para el análisis de los resultados obtenidos, así como para comprobar que ambos algoritmos funcionan correctamente y que tanto las visualizaciones como el código funcionan y se actualizan de forma automática sin problema.

3. Recursos a Emplear

Al tratarse de un proyecto que abarca diferentes modelos y que tiene unos objetivos con metas tan dispares, será necesario emplear una gran cantidad de herramientas y de librerías para poder terminar el proyecto con éxito.

En primer lugar, emplearemos Alteryx, una herramienta desarrollada por la compañía americana homóloga, que pretende facilitar el proceso de la extracción y la limpieza de datos a los científicos de datos de todo el mundo. Con este programa, realizaremos una consulta SQL contra la base de datos local de la compañía que se actualiza cada semana. Mediante Alteryx, podremos no solo llevar a cabo esta extracción de los datos, sino llevar a cabo la limpieza y el preprocesador de datos necesario para crear las tablas con los campos y datos necesarios para cada uno de los modelos. Alteryx es una herramienta de pago, que dispone de una versión de prueba de un mes, aunque este se antoja insuficiente si tenemos en cuenta el volumen del proyecto y su proyección a futuro.

En cuanto a los diferentes modelos, se desarrollarán usando Python como leguaje para programar estos modelos, y más en concreto Jupyter como IDE, para su desarrollo y optimización. Al tratarse Jupyter de una herramienta de código abierto, su acceso no supondrá ningún problema a través de Anaconda, una distribución de Python y R muy extendido entre la comunidad de los científicos de datos, ya que incluye muchos paquetes disponibles en los diferentes sistemas operativos y hace las tareas de gestor de paquetes de ambos lenguajes, lo que nos permitirá, además de obtener Jupyter, ser capaces de instalar el resto de las librerías necesarias para nuestro proyecto. Aun así, en el caso de no ser capaces de obtener alguna versión o librería a través de la consola e instalación de Anaconda, también es posible utilizar pip, un administrador de paquetes gratuito para Python que nos permitirá acceder a las librerías que no sea posible acceder con conda.

Estas librerías que necesitaremos para el desarrollo de modelos son muy variadas y algunas de ellas serán comunes en todos los proyectos como son las librerías de gestión de datos y tablas como pandas o numpy, o bien las librerías de visualización de resultados como seaborn o matplotlib. Por su parte, otras librerías serán específicas de cada proyecto, como son fbprophet,

una herramienta desarrollada por Facebook para facilitar la creación de series temporales a usuarios no tan experimentados, o la librería sklearn, una librería que pone a la disposición de cualquier usuario una amplia variedad de modelos de clasificación, regresión y clustering. Aun así, todos estos recursos son totalmente gratuitos y es posible descargarlos, junto a Jupyter, desde Anaconda con gran facilidad.

Por último, para ser capaces de presentar y exponer los avances de nuestro trabajo utilizaremos Power BI, una herramienta desarrollada por Microsoft que permite crear dashboards con diferentes visualizaciones sobre los datos obtenidos por los modelos a los que se pueden aplicar filtros y modificaciones variables en tiempo real que permiten expone los conceptos de una forma más efectiva. En este caso, Power BI también es una herramienta de pago, por lo que será necesario la obtención de licencia una vez superada la prueba gratuita inicial.

Una vez obtenidas todas estas herramientas mencionadas estaremos listos para comenzar con nuestro proyecto, disponiendo de los recursos necesarios para afrontar cada uno de los objetivos.

4. Estado del Arte

En este apartado vamos a repasar los diferentes métodos y herramientas seleccionados para la realización del presente proyecto, así como su funcionamiento y las posibles herramientas alternativas a estas y por qué fueron descartadas.

4.1 Preprocesado de Datos

El primer paso de nuestro proyecto y quizás el más importante consistía en extraer los datos de una base de datos local de la compañía, en dicha base de datos se realizan actualizaciones semanales de los ingresos de las compañías, de las ventas, en definitiva. La base de datos es una base de tipo SQL, un modelo relacional con filas y columnas en el que cada sábado se agregan nuevas columnas a la tabla con la información de la pasada semana.

Lo que se pretende en esta primera parte del proyecto es la extracción de dichos datos y su posterior procesado, para poder así obtener de la tabla resultado los datos para ambos modelos de forma fácil y eficaz. Para dicha tarea hemos elegido la herramienta Alteryx que nos permite realizarlo de forma fácil y visual, sin embargo, existen algunas alternativas, como por ejemplo realizar estas tareas con un código en Python mediante la librería Pandas, lo que probablemente sería una de las primeras opciones para casi cualquier científico de datos.

4.1.1 Pandas

Figura 4.1: Logo librería Pandas

Esta librería es una de las principales formas de crear, modificar, juntar y eliminar tablas y datos, es una librería escrita para el lenguaje de programación Python, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales. Su nombre deriva del término econométrico “panel data”, ya que ofrece datasets que permiten la observación durante varios instantes del tiempo por un mismo individuo.

Esta librería fue creada por Wes McKinney, quien empezó a desarrollar la librería en AQR Capital entre 2007 y 2010, cuando trabajaba en dicha empresa. Esta librería es una de las más populares y usadas en el lenguaje Python, debido a que ofrece la posibilidad de importar datos desde múltiples formatos de datos diferente, como son comma-separated, JSON, SQL. Microsoft Excel, entre otros.

Esto datos se almacenan en dataframes con los que también se permiten múltiples operaciones como son merge, reshaping, selecting, joins, así como limpieza de datos entre otras, en definitiva, una herramienta muy potente que bien podría servir para cumplir el objetivo que nos hemos propuesto en este primer apartado, de forma gratuita y con un uso no demasiado complicado y un resultado a la altura. Aun así, a pesar de todos estos argumentos a favor, decidimos decantarnos por Alteryx, por los motivos que voy a exponer a continuación.

4.1.2 Alteryx Designer

Alteryx es una compañía de software americana fundada en Irving, California, Estados Unidos, en el año 1997 por Dean Stoecker, Olivia Duane Adams and Ned Harding, cuyos productos buscan poner la analítica avanzada al alcance de cualquier trabajador de data del mundo. La firma Gartner reconoció a esta empresa como un leader en su Cuadrante Mágico de 2018 para Ciencia de Datos y Machine Learning.

Figura 4.2: Logo Alteryx Desiger

En concreto, Alteryx Designer, el software que utilizaremos en el proyecto, gira en torno a tres grandes módulos o secciones que se integran dentro de un flujo de trabajo intuitivo para la combinación y manipulación de datos de diferentes fuentes, la creación de modelos mediante esos datos y su posterior representación para mostrar los resultados obtenidos.

En nuestro caso, nos centraremos en la primera de dichas funcionalidades, Alteryx Designer permite la recopilación, mezcla e informe de datos desde casi cualquier tipo de fuente, base de datos, aplicaciones ERP o basadas en la nube, archivos de Microsoft Office o Hadoop. Toda esta cantidad de datos resulta fácil de limpiar y juntar debido a sus secuencias de cambios visuales que permite combinarlos de forma simple e intuitiva, lo que permite a cualquier usuario sin experiencia previa en programación crear un flujo de limpieza de datos mediante su método de arrastrar y soltar.

Al igual que pandas, esta herramienta permite múltiples operaciones con tablas como son merge, join o reshaping entre otras. Algunas de las funcionalidades que ofrece y que emplearemos en nuestro proyecto son las siguientes:

Figura 4.3: Función Browse Alteryx Designer

Browse: Esta funcionalidad permite visualizar los datos en cualquier punto del flujo de ejecución. Además, realiza un perfilado de cada campo de datos, indicando el tipo, el porcentaje en el que se encuentra cada dato y los valores de los datos de cada columna o campo.

Figura 4.4: Función Input Data

Input Data: Permite introducir datos en el workflow de datos, tanto de bases de datos mediante una query, como de una tabla escogida de un fichero local.

Figura 4.5: Función Text Input

Text Input: Permite introducir datos al workflow de forma manual

Figura 4.6: Función Output Data

Output Data: Permite guardar el flujo de datos resultante en un archivo de cualquier extensión o incluso en una base de datos mediante una query.

Figura 4.7: Función Data Cleansing

Data Cleansing: Esta función permite llevar a cabo varias funciones básicas de limpieza de datos como son reemplazar los null values, quitar la puntuación o las mayúsculas.

Figura 4.8: Función Filter

Filter: Esta función permite dividir un flujo de entrada en dos flujos de salida, basándose en si los datos satisfacen o no una expresión de comparación.

Figura 4.9: Función Formula

Formula: Esta es una de las funcionalidades más potentes, permite modificar una o varias columnas mediante una amplia variedad de operaciones.

[1]

Figura 4.10: Función Select

Select: Permite cambiar el nombre, orden y tipo de columnas, así como seleccionarlas y deseleccionarlas dentro del workflow de datos en función de las operaciones que deseamos realizar.

Figura 4.11: Función Join

Join: Esta función permite combinar dos flujos de datos de entrada, basándose en los campos comunes seleccionados previamente, de esta forma la salida, cada fila contiene las columnas de ambas entradas.

Figura 4.12: Función Union

Union: Combina dos o más flujos de entrada con estructuras similares, basado en los campos o en la posición, de esta forma, en la salida, las columnas contienen los datos de todas las entradas.

Figura 4.13: Función DateTime

DateTime: Esta función permite transformar un tipo de dato de entrada, ya sea string o timestamp a una amplia gama de formatos de tipo fecha.

Figura 4.14: Función Summarize

Summarize: Permite agrupar los datos mediante agrupados, suma, máximos, procesado espacial, entre otras. El resultado que obtenemos de esta función no es otro que el resultado de las operaciones que se aplican en ella a cada columna.

La combinación de todas estas funcionalidades permite crear flujos en el espacio de trabajo de Alteryx Designer, para ello, solo es necesario arrastrar la funcionalidad necesaria dentro de dicho espacio uniéndolo con las funcionalidades previas para combinarlas y poder crear el mencionado flujo hasta obtener la salida deseada por el usuario, como vemos en Figura 4.15.

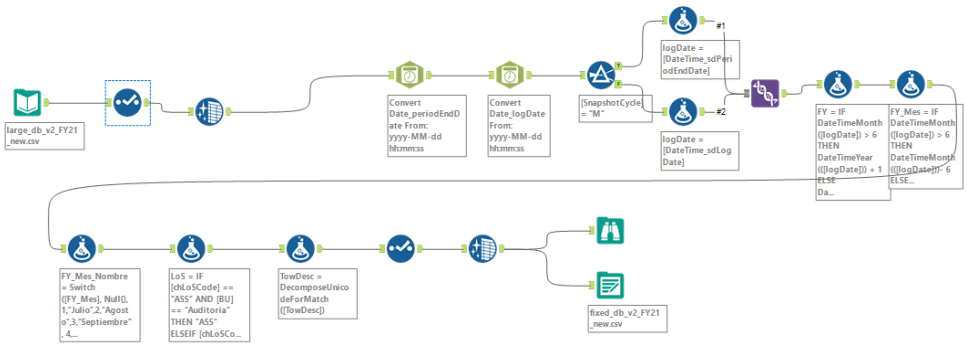


Figura 4.15: Flujo creado con Alteryx Designer

Por todo esto podemos considerar a Alteryx Designer como una herramienta a la altura de Pandas en lo que a tratamiento de datos se refiere, sin embargo, la facilidad con la que en esta herramienta se implementan todas las funcionalidades y su sistema de pincha y arrastra para crear flujos visuales que permitan su fácil creación, interpretación y modificación hacen que esta herramienta convierta el trabajo pesado de la adquisición y limpieza de datos en algo más ameno, siendo esta la razón fundamental por la que elegimos esta herramienta para la tarea del preprocesado de datos.

4.2 Predicción de ventas

Una vez completada la primera fase de extracción y limpieza de los datos ya podremos comenzar a elaborar nuestros modelos de machine learning, en primer lugar, llevaremos a cabo un modelo de series temporales como ya hemos comentado.

El objetivo es llevar a cabo una predicción de ventas del año fiscal, el cual, en el caso de PwC abarca desde julio hasta julio del año siguiente, donde se cierra el año. Esta predicción se elabora semanalmente y pretende estimar el valor de ventas de la compañía, tanto semanalmente como acumuladas hasta el junio siguiente en el que se cerraría el año.

Una vez que hemos definido esto, el primer paso consiste en la elección de un modelo para realizarlo, de primeras, parece que existen dos tipos de algoritmos que deberían lograr un buen resultado en esta situación, debido a su capacidad de aprender de datos pasados, estos son las redes neuronales recurrentes y las series temporales.

4.2.1 Redes Neuronales Recurrentes

Las redes neuronales recurrentes o RNN (Recurrent Neural Networks) son un tipo de red neuronal que se caracteriza por ser la única de estas que es capaz de tratar la dimensión del "tiempo", gracias a sus neuronas recurrentes, en las que la salida de dichas neuronas se introduce como dato de entrada de esta o de una anterior (Figura 4.16). De esta forma, este tipo de redes son capaces de analizar datos de series temporales.

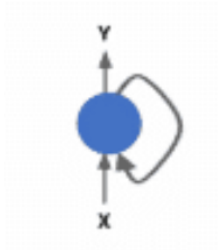


Figura 4.16: Neurona Recurrente

Al implementar estas neuronas recurrentes logramos que este tipo de redes tenga "memoria", ya que las salidas de una neurona especifican no es más que una función de los estados anteriores, por lo que, al introducir esta salida como entrada de la misma neurona, se puede decir que estas neuronas disponen de memoria, esta parte de la neurona que preserva el estado a través del tiempo se conoce además como memory cell.

Este tipo de algoritmo son muy populares y, sin duda, podríamos lograr una predicción muy satisfactoria mediante su implementación, sin embargo, la interpretabilidad de estos algoritmos es muy complicada, al igual que todas las redes neuronales, se consideran cajas negras en las que no es posible interpretar los resultados y el porqué de estos, lo que resulta critico en la finalidad de una previsión de ventas, donde el mayor valor se obtiene al saber el porqué de las predicción, porque se predicen perdidas o ganancias y que se puede realizar al respecto.

4.2.2 Forecasting

El forecasting consiste en hacer predicciones a futuro mediante los datos históricos, este método se basa en la idea de que existen patrones o tendencias en los datos pasados que tienden a repetirse en el futuro con una componente de ruido.

Para el forecasting el futuro es completamente desconocido y simplemente se asume que lo que ha sucedido en el pasado, o que la aproximación que hemos hecho de nuestra serie en base a los datos de pasado se repetirá en el futuro, de esta forma, podemos ser capaces de predecir dichos valores a futuro.

Las predicciones mediante series temporales es una forma de forecasting, una de las más eficaces y se descomponen en cuatro componentes principales, que permiten llevar a cabo la predicción a futuro que comentaba:

- **Tendencia:** Es la línea base de la que parte nuestra serie, el valor inicial al que se suman todos los demás y que resulta especialmente relevante desde un punto de vista de negocio, ya que permite interpretar si nuestra serie crece o decrece con el tiempo, es decir, si los ingresos de la empresa crecen con los años o por el contrario decrecen.

- **Estacionalidad:** Estos son los patrones opcionales que se repiten con el tiempo, pueden ser diarios, semanales, anuales… y no todas las series tienen por qué tenerlos, aunque sí que deben tener al menos uno para considerarse una serie temporal.

- **Ruido:** Es la componente de la serie que somos incapaces de explicar mediante los dos anteriores y que determina el error de nuestra serie.

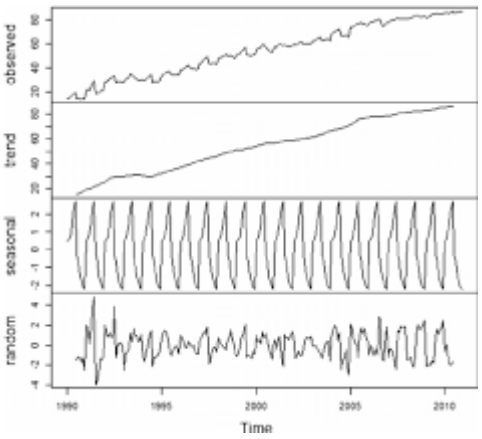


Figura 4.17: Componentes de la serie temporal [2]

En la Figura 4.17 podemos observar las componente que acabamos de describir, donde vemos como la serie en cuestión se trata de una serie con tendencia ascendente, algo muy positivo desde el punto de vista de un negocio; una estacionalidad anual, al igual que las series que vamos a tratar de modelar en este proyecto; y una componente de ruido que somos incapaces de explicar con las dos anteriores

La posibilidad de realizar esta descomposición hace que estos algoritmos seas muy intuitivos y sus resultados visuales y fáciles de interpretar, algo que como he dicho resulta crucial en nuestro caso de uso, por lo que las series temporales son el algoritmo que decidimos seleccionar para llevar a cabo nuestra predicción.

4.2.3 Series ARIMA Y SARIMA

Las series temporales ARIMA y SARIMA son algunos de los métodos más populares a la hora de realizar series temporales. El termino ARIMA proviene de AutoRegresive Integrated Moving Average y es un tipo de serie temporal que captura una serie de estructuras temporales estándar de las series temporales. Por su parte, SARIMA proviene de Seasonal ARIMA, una evolución del modelo ARIMA.

El acrónimo de ARIMA es bastante descriptivo y captura los aspectos principales del modelo como hemos mencionado son los siguientes:

- Auto regresión: Es un modelo que emplea la dependencia del valor actual con un número determinado de instantes justo anteriores a este

- Integrado: Este término se refiere al uso de diferencian las observaciones para tratar de desestacionalizar la serie, esto se consigue, por ejemplo, restando a una observación la observación del instante anterior.

- Media móvil: Es un modelo que usa la dependencia entre una observación y el error residual de una media móvil aplicada a las observaciones con cierto retardo.

Estos tres son los componentes principales del modelo, los cuales se especifican mediante parámetros, que se especifican como tres números enteros que permiten obtener el modelo ARIMA concreto.

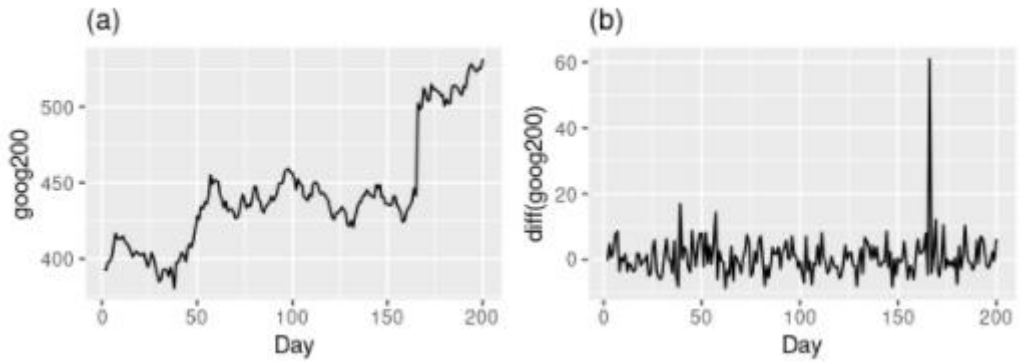


Figura 4.18: Acciones de Google durante 200 días consecutivos. [3]

En primer lugar, los datos se preparan por cierto grado de diferenciación para conseguir que estos sean estacionarios, eliminando así la tendencia y la estacionalidad que afectan negativamente al modelo de regresión. Este proceso lo podemos apreciar en la Figura 4.18, donde la imagen a representa el valor en dólares de las acciones de Google durante 200 días consecutivos, mientras que, la imagen b representa la misma serie una vez hemos realizado la diferenciación y convertido la serie en estacionaria.

Después, se emplean los instantes pasados para calcular la correlación entre estos y el instante actual y se realiza ese mismo calculo para cada instante de la serie, una vez hecho eso se

calcula la correlación de cada instante pasado de tiempo, obteniendo lo que podemos observar en la Figura 4.19.

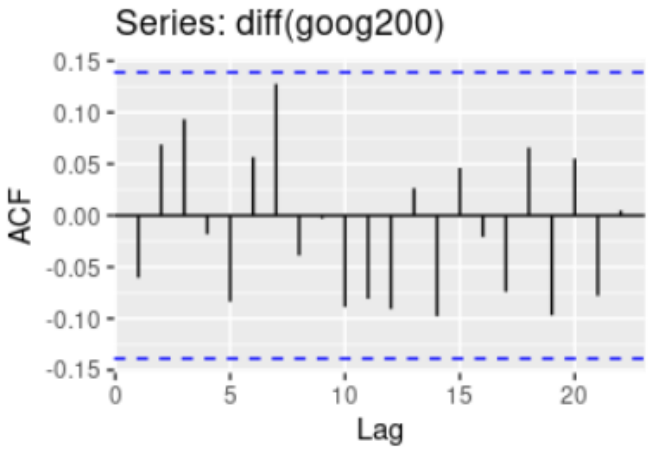


Figura 4.19: Correlaciones de instantes pasados entre valor acciones de Google. [3]

Este grafico representa en qué medida cada instante actual se puede explicar mediante los instantes pasados en el modelo ARIMA del que disponemos en ese momento, como vemos todos los valores se encuentran entre los valores 0.15 y -0.15, por lo que consideramos esas relaciones como puramente aleatorias y diríamos que el error de la serie es únicamente "ruido blanco", puramente aleatorio, por lo que no es posible explicar nada mas de nuestra serie mediante los instantes pasados, y por lo tanto, el modelo estaría completo.

En el caso de las series de tipo SARIMA, a todo lo mencionado anteriormente es necesario añadirle la componente estacional, es decir, a la serie ARIMA habría que añadirle otros tres parámetros (autoregresión, diferenciación y media móvil), pero en este caso para la componente estacionaria, repitiendo el proceso mencionado anteriormente para esta componente.

Ambos modelos de series temporales ARIMA y SARIMA son dos de las formas más populares de construir este tipo de modelos, sin embargo, estos modelos son complejos en cuanto a implementación y requieren un conocimiento de estos y de series temporales más amplio y un mayor esfuerzo para construirlos correctamente. Por esto, decidimos que existían mejores y más simples formas de llevar a cabo el objetivo final de la predicción, como el que veremos a continuación, la librería Prophet de Facebook, la cual sería la forma seleccionada en nuestro proyecto.

4.2.4 Prophet

Prophet o "Facebook Prophet" es una librería open-source para series temporales desarrollada por Facebook, Prophet lleva a cabo lo que ellos llaman un modelo aditivo de forecasting por series temporales.

Este modelo pretende ayudar a sus usuarios a construir series temporales de confianza de una forma fácil y rápida, lo que ellos aseguran es algo no tan fácil de conseguir, pero que puede resultar de ayuda en muchos negocios.

Esta librería utiliza el modelo de descomposición de series temporales (Harvey & Peters 1990) con tres componentes principales: la estacionalidad, la tendencia y lo que ellos llaman holidays. Estos tres términos se combinan como vemos en la siguiente ecuación:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

Ecuación 4.1

En esta ecuación g(t) es la tendencia que modela cambios no periódicos en la serie como ya hemos explicado, s(h) representa cambios periódicos o estacionalidades, mientras h(t) representa el efecto de los mencionados holidays, que ocurren en dias potencialmente irregulares que pueden o no repetirse en el tiempo. El termino épsilon representa el error, es decir, cualquier cambio que no haya sido modelado por nuestro modelo.

De esta forma, abordamos el problema del forecasting como un proceso de ajustar curvas, lo que resulta muy diferente de los modelos que cuentan de forma explícita con la estructura de dependencia temporal en los datos, como por ejemplo el modelo ARIMA. Por supuesto Prophet tiene sus desventajas con respecto a ARIMA, pero algunas de las ventajas que nos ofrece son: flexibilidad, es fácil acomodar la estacionalidad con periodos diferentes y dejar que el modelo haga sus suposiciones sobre la tendencia; a diferencia de ARIMA, las medidas no tienen por qué estar medidas en periodos regulares y no es necesario interpolar los valores faltantes; el entrenamiento es muy rápido y los parámetros son muy fácilmente interpretables.

La forma en la que Prophet calcula la tendencia se basa en una suposición de crecimiento saturado, al igual que el crecimiento de la población o de la gente que utiliza Facebook, los cuales se saturan al llegar a cierto punto, como por ejemplo la gente que usa internet. Es te comportamiento se modela con la siguiente ecuación:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))},$$

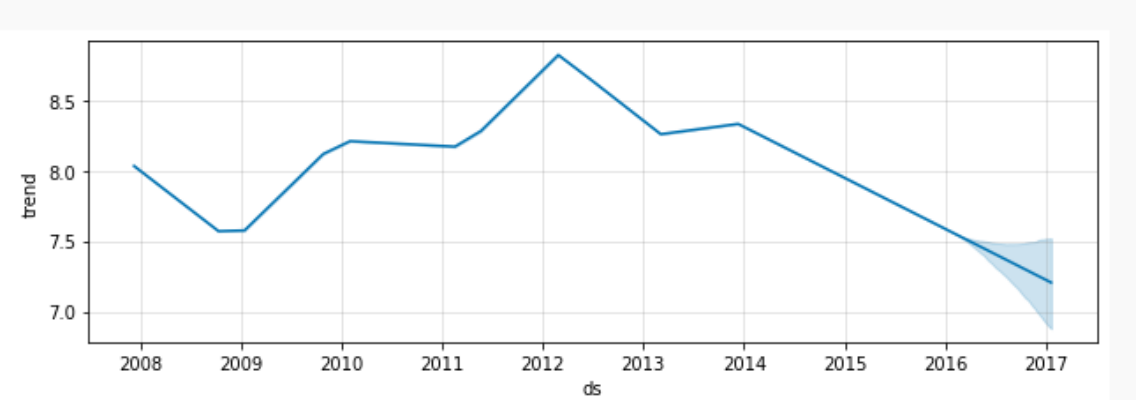
Ecuación 4.2

En esta ecuación es el máximo número al que satura el crecimiento, mientras que k es el ratio de crecimiento y m es un parámetro de offset. No obstante, hay dos aspectos que esta fórmula no

contempla, la posibilidad de un techo al que saturar variable, que crezca a la vez que lo hace el número de individuos, como la gente que usa internet sería el techo de la gente que usa Facebook, pero sería un techo que crece con la medida cuyo crecimiento estamos estudiando. Y, en segundo lugar, ese crecimiento no tiene por qué ser lineal, para eso Facebook desarrolla los changepoints o puntos de cambio, en los que se define un cambio claro en la tendencia o en el ratio de crecimiento y que se definen de forma automática. Teniendo en cuanta estas dos premisas, la fórmula de la tendencia seria la siguiente:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})))}.$$

Ecuación 4.3



Gráfica 4.1: Grafica de tendencia en la documentación de Prophet [4]

Por su parte, la estacionalidad que pretende modelar comportamientos humanos puede ser realmente complicada, ya que llevan a tendencias de 5 dias que modelan una semana de trabajo, lo que produce efectos en la serie que se repiten cada semana, mientras que las vacaciones o parones escolares provocan efectos que se repiten cada año. Para modelar este comportamiento utilizan modelos de estacionalidad utilizan funciones periódicas de t.

En concreto, utilizan series de Fourier para proporcionar modelos flexibles de efectos periódicos, estos modelos se basan en sumas de funciones armónicas, combinadas por una suma ponderada, como vemos en la siguiente ecuación se representa una serie de Fourier de orden N:

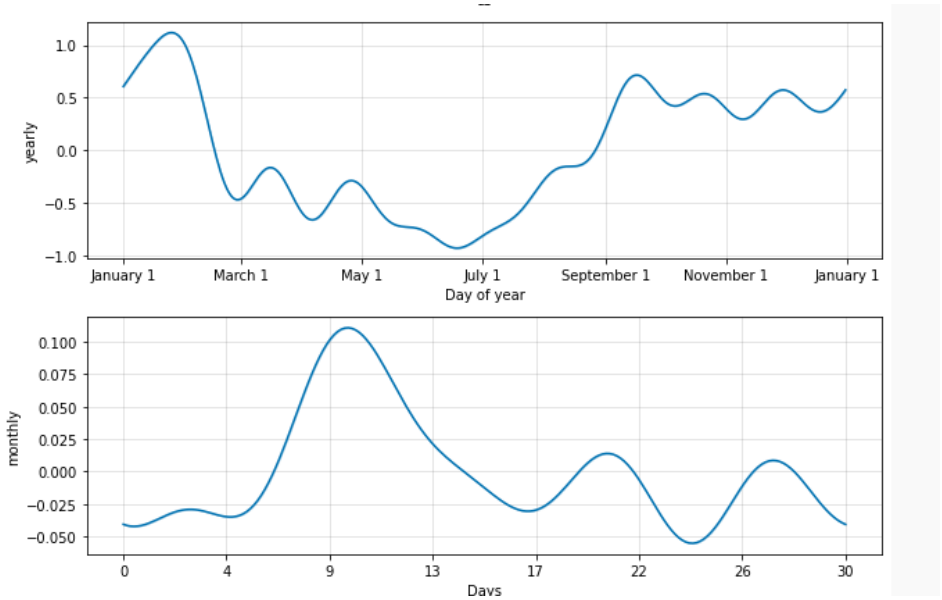
$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right)$$

Ecuación 4.4

Donde P es el periodo de nuestra estacionalidad y N el orden de nuestra serie de Fourier.

Resolver esta función para hallar la estacionalidad, requiere de estimar los 2*N parámetros que la componen, lo que se hace construyendo una matriz de vectores de estacionalidad para cada valor de t en los datos históricos y futuros.

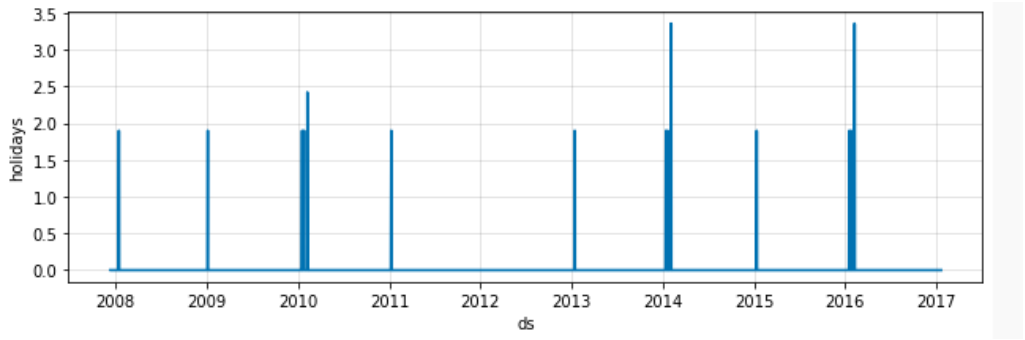
Truncar la serie en N implica un filtro de paso bajo, por lo que, aumentar la N permite ajustar estacionalidades que cambian más rápidamente con el tiempo, lo que aumenta el riesgo de un posible sobreajuste. Los valores por defecto de N son 10 para estacionalidades anuales y 3 para estacionalidades semanales, ya que afirman que funcionan de forma optime en la mayoría de las situaciones, pero estos valores se pueden aumentar en función de lo que sea necesario en cada caso de uso.



Gráfica 4.2: Grafica de estacionalidad semanal y anual en la documentación de Prophet [4]

Por último, están los holidays, son vacaciones o eventos que se producen de forma regular pero que es difícil modelar por su comportamiento, ya que no suceden siempre en exactamente los mismos dias del año, como por ejemplo la semana santa.

Para estos eventos, Prophet da la posibilidad de proporcionar una lista con fechas exactas en las que se produce una vacación o un evento en el pasado y futuro con un nombre identificativo único, de tal forma que es posible tener varios tipos, para los que se calcula la diferencia del valor real con la estimación de la serie en esos dias, de esa forma es posible saber cuál es el impacto de ese día especial en el valor de la serie.



Gráfica 4.3: Grafica de holidays en la documentación de Prophet [4]

Además de ofrecer una forma fácil de modelar series temporales, Prophet ofrece una documentación simple y muy intuitiva en la que explica como es posible optimizar modelos con un tuneo de hiperparámetros y como funciona su calculo del error mediante validación cruzada.

Prophet basa su calculo del error mediante validación cruzada en tres parámetros a los que llama initial, que representa el periodo de los datos que se utilizara para entrenamiento siempre, el period, que es el espacio de tiempo que vamos a añadir a nuestro entrenamiento en cada uno de los folds; y por último, el horizon, que es el tiempo en el que se va a realizar la predicción y en los que se calculara el error.

De esta forma, si disponemos de dos años de histórico de datos y establecemos que el initial se un año con horizon y period ambos de dos meses, se realizara una validación cruzada de 6 folds, en los que se realizaran seis entrenamientos de 12, 14, 16, 18, 20 y 22 meses respectivamente, con una predicción de dos meses en cada uno.

Una vez explicados todos estos conceptos podemos observar como Prophet es una herramienta muy intuitiva que permite construir series temporales de forma relativamente fácil, pero que brilla especialmente por su fácil interpretabilidad, por lo que es la herramienta seleccionada para nuestro proyecto.

4.3 Recomendador de Proyectos

Para el segundo objetivo de nuestro proyecto, que consiste en desarrollar un recomendador de proyectos, necesitaremos llevar a cabo dos algoritmos diferentes, ya que el primero simplificara y ayudara a lograr mejores resultados en el segundo, el proceso de clustering nos permitirá agrupar a nuestros clientes encontrando similitudes en los proyectos que contratan y entre las propias empresas, lo que simplifica claramente el proceso de recomendación, ya que esos clientes serán más propensos a contratar servicios similares.

4.3.1 Clustering

Clustering es el proceso por el cual se divide la población de individuos en grupos, de tal forma que los individuos que están dentro de un mismo grupo son más similares entre sí que lo serian con los miembros de otro grupo. En otras palabras, el objetivo es crear grupos con rasgos similares y asignarlos a un clúster.

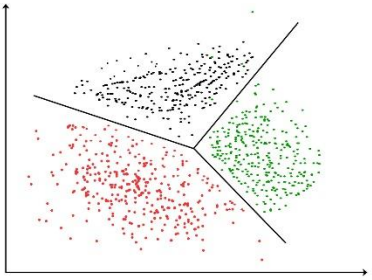


Figura 4.20: Clustering de población 2D [5]

A la hora de diseñar uno de estos algoritmos de clustering suelen surgir varias preguntas o puntos a definir que resultan clave para el devenir de esta primera parte del objetivo, en primer lugar, hay que decidir si nuestro clustering será "soft" o "hard", esto quiere decir que, si dejaremos que algunos individuos queden sin agrupación o, por el contrario, si todos los individuos de nuestra población pertenecerán a un grupo. En nuestro caso, parece evidente pensar que optaremos por la segunda opción, ya que queremos recomendar un proyecto a cualquier cliente, aunque esta recomendación no sea optima, aunque no existe ningún cliente con unas características claramente similares.

Además, existen diferentes técnicas a la hora de agrupar la población, lo que provoca que existan diferentes tipos de clustering:

- **K-Means Clustering:** Es probablemente la técnica de clustering más conocida y popular por su simplicidad y rapidez de computación. Esta técnica consiste en definir un número de clústeres en los que se desea agrupar los datos y se definen tantos puntos aleatorios como clúster se desean. Después, se calcula la proximidad de cada punto a los centros definidos, de tal forma que cada punto pertenece al clúster del centro más próximo. Por último, se recalcula el centro a partir de la media de todos los puntos del grupo y se repiten todos los pasos durante un numero de iteraciones o hasta que los centro no varíen demasiado.

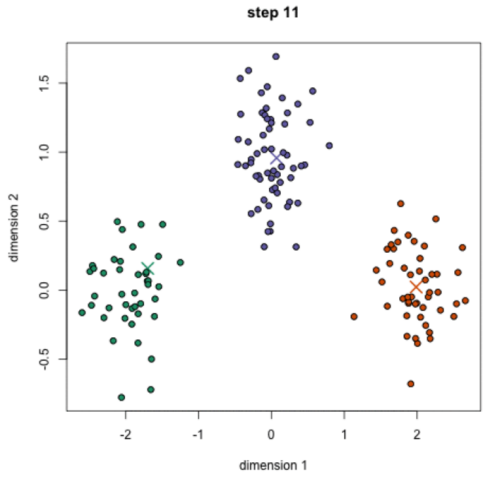


Figura 4.21: Iteración 11 de K-Means Clustering [6]

El método de K-Means tiene algunas ventajas muy claras que lo hacen popular, ya que es muy rápido, puesto que lo único que hacemos es calcular distancias a un centro, y, además, es fácil de implementar y de entender. Sin embargo, K-means tiene algunas desventajas claras, como, por ejemplo, que es necesario preseleccionar el número de clústeres que se desea crear, idealmente queremos que esto lo averigüe el algoritmo de forma independiente. También es un método que comienza de una forma totalmente aleatoria por lo que puede darse el caso de que si corremos este algoritmo de forma idéntica dos veces seguidas obtengamos resultados diferentes.

- **Mean-Shift Clustering:** Este tipo de clustering se basa en la idea de ventanas móviles que pretenden encontrar áreas densas donde los puntos se aglutinan. Es un algoritmo basado en centroides, lo que quiere decir que el objetivo final es encontrar el centro de cada nube de puntos, lo que se consigue actualizando los candidatos a punto céntrico haciendo la media de los puntos en el interior de nuestra ventana móvil (Figura 4.22).

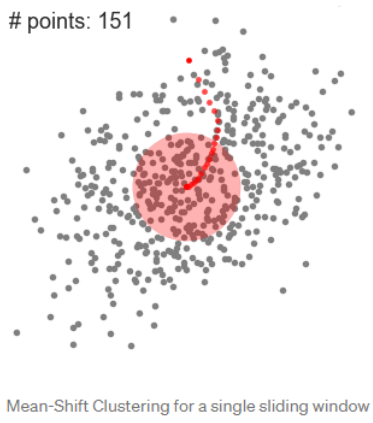


Figura 4.22: Calculo de centros mediante ventana móvil [6]

Por último, en el estado final de post procesado, se filtran estos centros eliminando duplicados o puntos casi duplicados para conseguir los puntos céntricos finales y los grupos que abarcan como vemos en la Figura 4.23.

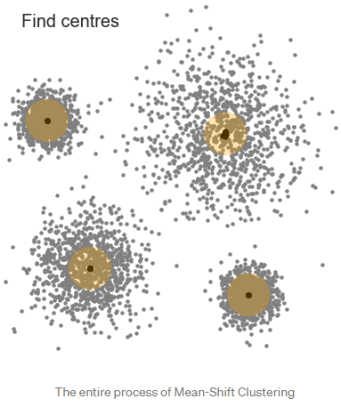


Figura 4.23: Estado final Shift-Mean Clustering [6]

A diferencia de en K-Means, en este método no es necesario seleccionar el número de clústeres, el algoritmo lo encuentra de forma independiente, lo que supone una gran ventaja, además, este método también resulta simple y fácil de entender, sin embargo, requiere que se defina el tamaño de la ventana móvil, lo que resulta clave en el resultado y suele ser una elección no trivial.

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN):** Es un algoritmo basado en la densidad, al igual que el Shift-Mean clustering, pero con algunas ventajas significativas.

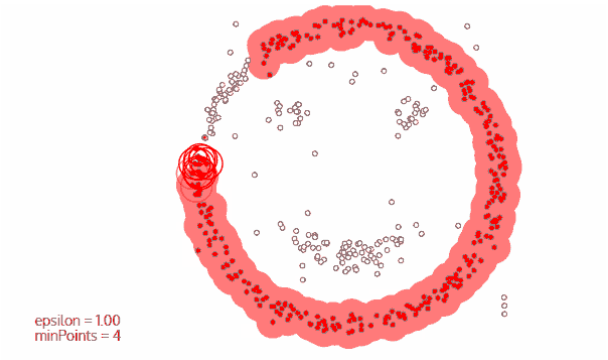


Figura 4.24: Proceso de clustering de DBSCAN [6]

Este algoritmo necesita que se especifiquen dos parámetros, épsilon y mínimo número de puntos. Una vez definidos, el algoritmo selecciona un punto aleatorio que no ha sido visitado

y encuentra todos sus vecinos dentro de la distancia especificada épsilon, si el número de puntos supera el especificado el proceso de clustering empieza, sino el punto se clasifica como ruido (Figura 4.25). En caso de que el proceso de clustering comience, se visitan todos los puntos que han sido añadidos para encontrar sus vecinos en el rango épsilon, verificando la condición de mínimo de puntos (Figura 4.24). Una vez se realiza esta comprobación, los puntos se marcan como visitados y este proceso se repite con todos los puntos hasta que se visitan todos ellos.

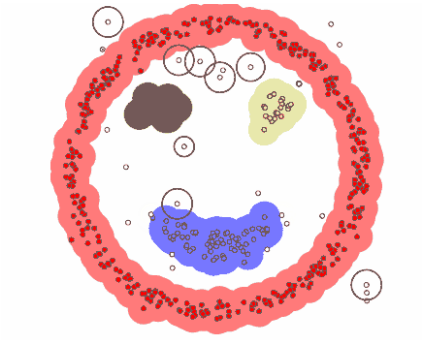


Figura 4.25: Clasificación de puntos sin vecinos como ruido [6]

Este algoritmo de clustering tiene algunas ventajas claras, como que no necesita que se especifique el número de clústeres, y también identifica ruido y valores atípicos que pueden quedar fuera de algún clúster, y, por último, es un algoritmo que se adapta a formas extrañas y es capaz de modelar figuras más complejas. Aun así, tiene una desventaja clara, que es que no se comporta bien con densidades de puntos alternantes, ya que, al tener que predefinir épsilon y el mínimo de puntos no es posible que estos se adapten a las diferentes densidades, como sería el idílico. Lo mismo ocurre si los datos son de una dimensión muy alta, ya que el parámetro épsilon resulta complicado de definir correctamente.

- **Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM):** Una de las principales fallas del método de K-Means es el uso del valor medio para el centro del clúster, ya que esto suele ser un problema cuando los centros de dos clústeres diferentes están relativamente próximos o cuando la forma de los clústeres no es circular como vemos en la Figura 4.26.

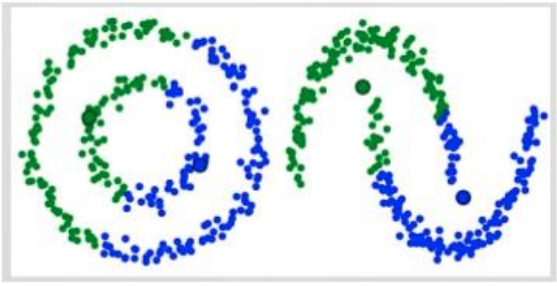


Figura 4.26: K-Means Clustering con casos atípicos [6]

Las misturas de gaussianas nos dan mucha más flexibilidad, ya que por este método asumimos que los puntos se encuentran distribuidos de forma gaussiana, lo que es menos restrictivo. De esta forma, hay dos parámetros que definen los clústeres, la media y la desviación estándar, lo que quiere decir que los clústeres pueden tener cualquier tipo de forma elíptica. Para definir estos dos parámetros del clúster se emplea el algoritmo de Expectation-Maximization (EM), al igual que en K-Means se define el número de clústeres y se inicializan unos centros aleatorios, después se calcula la probabilidad de que cada punto pertenezca a cada uno de los clústeres, por último, se calculan los centros de nuevo mediante una suma ponderada usando la probabilidad de cada punto de tal forma que se procura maximizar las probabilidades de cada punto, y esto se repite hasta la convergencia (Figura 4.27).

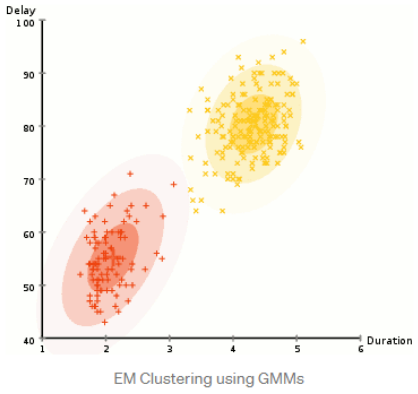


Figura 4.27: Clustering mediante mistura de Gaussianas [6]

Este método tiene dos ventajas claras, son mucho más flexibles que el método de K-Means, ya que, gracias al parámetro de la desviación estándar, los clústeres pueden tener formas variadas, incluso podríamos decir que K-Means es un tipo de mistura de Gaussianas, donde la desviación es igual a cero. Además, pueden tener puntos que pertenezcan a más de un clúster, ya que al calcularse la probabilidad de cada punto podría haber puntos que pertenecen a más de un clúster con mayor o menos probabilidad, aunque esto no resulta especialmente útil para nuestro caso de uso.

- **Agglomerative Hierarchical Clustering:** Este es un tipo de algoritmo conocido como "bottom-up", ya que tratamos cada punto como un solo individuo, partiendo desde la idea de que cada punto conforma un clúster por sí mismo (Figura 4.28).

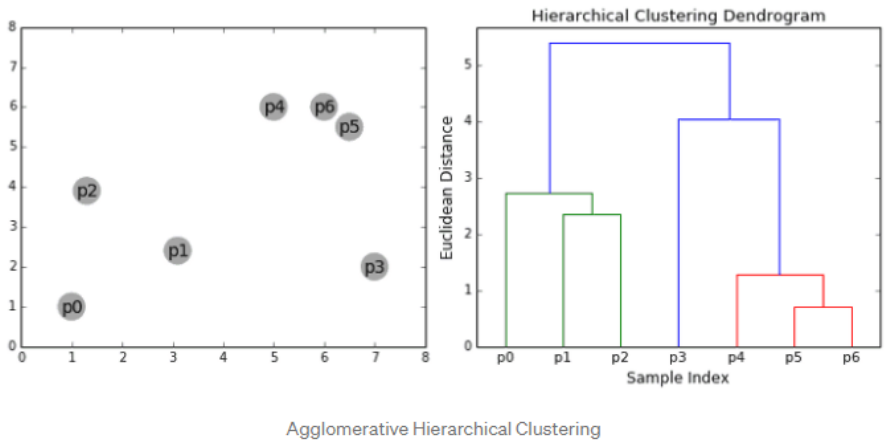


Figura 4.28: Cada punto conforma un clúster en hierarchical clustering [6]

Después, vamos escalando hacia la cima agrupando los puntos por niveles hasta obtener un único clúster que contenga a todos los puntos, de esta forma podemos elegir en que punto de ese proceso parar, obteniendo un resultado final que más se ajuste a nuestros datos (Figura 4.29).

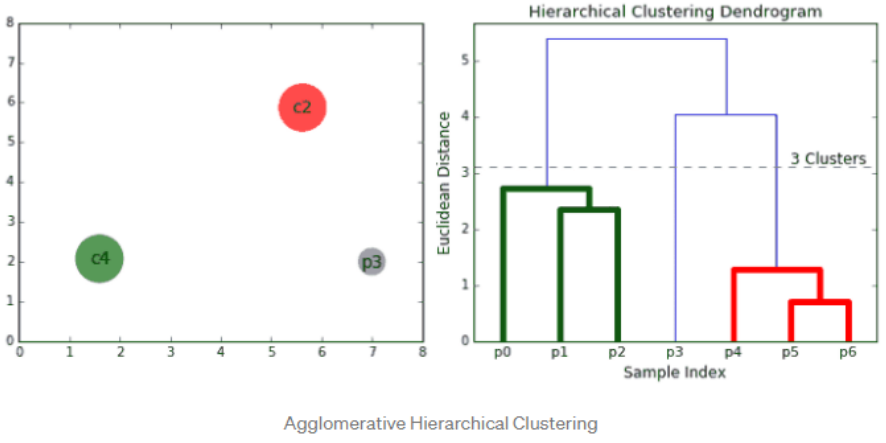


Figura 4.29: Resultado final la seleccionar 3 clústeres [6]

Este algoritmo no necesariamente requiere que especifiquemos el número de clúster, pero si lo podemos seleccionar en base al árbol que se genera al separar los puntos. Además, este algoritmo no es susceptible a una elección de distancia y es especialmente bueno cuando los

propios datos tienen una jerarquía y es necesario mantenerla, mientras que algunos de los otros algoritmos no pueden hacer esto.

Aun así, este método tiene algunas desventajas claras, como, por ejemplo, la dimensionalidad de los datos puede ser critica a la hora de seleccionar el corte de clústeres y este método requiere una gran computación y largos tiempos de entrenamiento, ya que tiene una alta complejidad.

En resumen, hemos mencionado y explorado los algoritmos más populares y eficaces a la hora de realizar un clustering, además, en la Figura 4.30 podemos observar cómo estos algoritmos se comportan ante situaciones muy variadas.

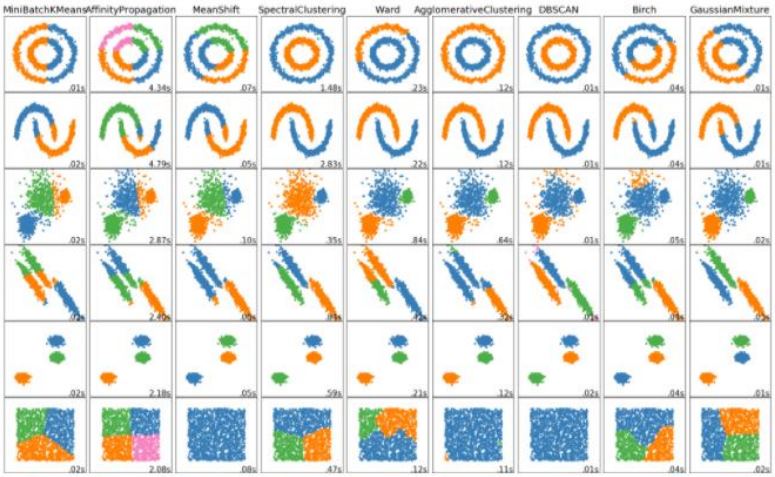


Figura 4.30: Desempeño de los diferentes algoritmos con diferentes configuraciones de datos [7]

Como podemos observar, existe mucha diferencia entre los diferentes algoritmos y parece que los jerárquicos y DBSCAN son los que tienen mejores resultados, sin embargo, debido a que nuestro problema será de una dimensionalidad alta y no queremos que existe ruido, sino que todos los clientes formen parte de un grupo, nos lleva a decantarnos por los algoritmos de mistura de gaussianas y K-Means.

En principio, cabría pensar que debido a que la mistura de gaussianas es una mejora de K-Means, este sería el algoritmo seleccionado, sin embargo, al tratarse de un paso intermedio de un proceso mayor con un amplio número de datos decidimos decantarnos por el algoritmo de K-Means por su rapidez de cómputo y su interpretabilidad en caso de errores.

4.3.2 Recomendadores

Una vez hemos completado el paso previo de crear un clustering de nuestros clientes, usaremos esa información, esos grupos de clientes para desarrollar un recomendador de productos que permita sugerir proyectos a los diferentes clientes en base al tipo de cliente del que se trate y a los servicios que haya contratado de nuestra firma.

Un sistema de recomendación es un algoritmo de machine learning que permite predecir si un proyecto o producto es de interés para un cliente o usuario, mediante un perfilado de los diferentes usuarios este algoritmo es capaz de sugerir productos que te deberían de gustar en base a tus gustos, los gustos de usuarios similares y el tipo de producto o productos similares, como vemos en la Figura 4.31. De esta forma podemos captar la atención de los clientes hacia ciertos productos o servicios en los que no habían plantado y que en ocasiones contratan con algunos de nuestros competidores.

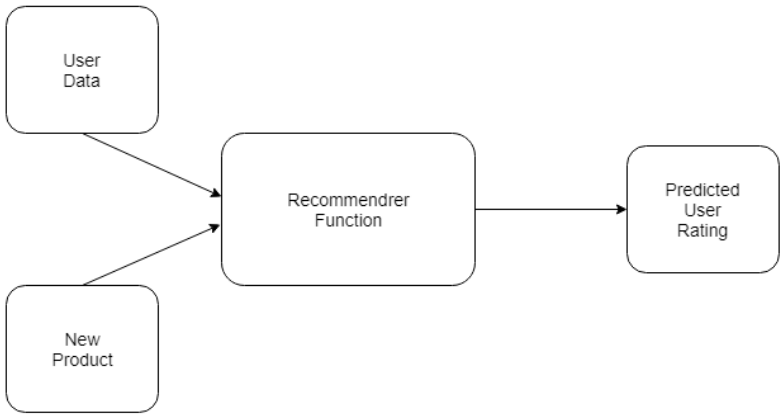


Figura 4.31: Cómo funciona un sistema de recordación [8]

El elemento más importante de estos algoritmos es la función de recomendación, el bloque central de la Figura 4.31, la cual toma los datos de los clientes y da una predicción del rating que dará ese cliente a un producto en concreto, o en nuestro caso, si dicho producto es de su interés o no.

A la hora de construir un sistema de recomendación resulta clave entender que existen tres tipos de relaciones:

1. **Usuario-Producto**: Esta relación ocurre cuando un usuario tiene preferencia por ciertos tipos de productos, como, por ejemplo, un jugador de futbol y aficionado del deporte tendrá tendencia a comprar y consumir productos relacionados con el futbol.

2. **Producto-Producto:** Esta relación ocurre cuando existen productos con naturalezas similares, ya sea por su apariencia o por sus características, como pueden ser libros o canciones del mismo género.

3. **Usuario-Usuario:** esta relación ocurre cuando algunos usuarios tienen gustos muy parecidos con respecto a los productos o servicios que consumen, por ejemplo, cuando esos usuarios son amigos o tienen entornos similares.

Existen muchas formas de saber o intuir si a un usuario le gusta un producto o servicio, estos métodos pueden ser explícitos, por el propio usuario indicando que cierto producto le gusta o no; o, por el contrario, pueden ser implícitos, son los que se generan al interactuar con el producto. En nuestro caso, no disponemos de ningún dato explicito por el que indicar al algoritmo si un producto le gusta o no, solo sabemos que servicios ha contratado y deberemos asumir que, si lo ha hecho, el producto le gusta.

Existen dos tipos de sistemas de recomendación:

- **Basados en contenido:** estos sistemas utilizan su conocimiento de los productos para recomendar otros nuevos, estas recomendaciones se basan en las características de los diferentes productos. Estos sistemas funcionan muy bien cuando se tiene una información descriptiva de los datos de antemano.

Este método requiere de esos datos descriptivos de los productos, ya que necesita una matriz de características de los productos para elaborar una predicción. Por ejemplo, pongamos que tenemos las valoraciones de los usuarios de la Tabla 4.1.

User\Movies	Interstellar	Inception	The Shining	Alien
Clark	5 stars			
Bruce			5 stars	
Tony				
Steve		2 stars		1 star

Tabla 4.1: Ratings de los usuarios de nuestros productos [8]

Como vemos, se trata de una matriz en la que la mayoría de los campos están vacíos y nuestro algoritmo basado en contenido necesita una segunda matriz para poder rellenar alguno de los huecos de nuestra matriz. Esa matriz será la matriz de películas, en la que se indican algunas de las características de las películas que ofrecemos y que podemos observar en la

Movie\Attributes	Science	Adventure	Horror	Mystery
Interstellar	5	4	1	2
Inception	5	3	1	2
The Shining	1	2	5	4
Alien	2	3	5	3

Tabla 4.2: Tabla de contenidos [8]

Como podemos ver, esta matriz contiene el género de cada una de las películas y nos permite identificar que películas son de contenido similar, de ahí el nombre del algoritmo. Viendo esta matriz resulta evidente que al usuario Clark le gustan las películas de ciencia ficción y sería lógico recomendarle la película Inception, mientras que a Bruce le gusta la película El Resplandor, que se trata de una película de terror, por lo que sería lógico recomendarle la película Alien, con un contenido similar. Esto es justo lo que hace este tipo de algoritmos como vemos en la Tabla 4.3.

User\Movies	Interstellar	Inception	The Shining	Alien
Clark	5 stars	5 stars		
Bruce			5 stars	5 stars
Tony				
Steve		2 stars		1 star

Tabla 4.3: El algoritmo predice algunos de los ratings desconocidos. [8]

Este método de recomendación tiene una ventaja muy clara, ya que funciona incluso cuando no existen reviews o ratings de un producto o servicio, pero requiere datos descriptivos de cada producto lo que resulta costoso y difícil de conseguir, además son menos populares por su difícil implementación en bases de datos amplias, ya que, no todos los usuarios tienen la misma opinión de los productos, aunque el contenido sea similar.

- **Recomendadores de filtro colaborativo:** Estos algoritmos hacen sus recomendaciones basándose en las recomendaciones que han hecho los usuarios de cada uno de los productos sin tener en cuenta los productos en sí. Simplemente trata de predecir qué productos son los que el usuario puede querer en base a lo que ya ha valorado y le ha gustado, algo que se parece notablemente a nuestro objetivo en este proyecto.

User\Movies	Interstellar	Inception	The Shining	Alien
Clark	5 stars	2 stars	1 star	
Bruce	1 star	5 stars		1 star
Tony	5 star	3 stars	1 star	5 stars
Steve	1 star	5 stars	5 stars	1 star

Tabla 4.4: Tabla de ratings de películas de nuestros usuarios [8]

En este caso, el algoritmo no se fijará en las otras películas que yo he valorado, sino que mirara los ratings que le han dado a esa película usuarios similares a mí. De esta forma, si observamos la Tabla 4.4, podemos ver como Bruce aún no ha visto El Resplandor y sus gustos son significativamente similares a los de Steve, por lo que resulta evidente pensar que le dará la misma puntuación a esa película que Steve. Lo mismo ocurre con Clark y Tony al valorar Alien.

User\Movies	Interstellar	Inception	The Shining	Alien
Clark	5 stars	2 stars	1 star	5 stars
Bruce	1 star	5 stars	5 stars	1 star
Tony	5 stars	3 stars	1 star	5 stars
Steve	1 star	5 stars	5 stars	1 star

Tabla 4.5: Recomendación mediante filtro colaborativo [8]

En la Tabla 4.5 vemos como esas dos películas obtienen la misma puntación que la de los usuarios similares al pasarlas por nuestro sistema de recomendación. Este método no requiere de ningún tipo de descripción de productos, pero no puede recomendar productos si no existe ningún usuario que haya valorado dicho producto o no puede hacer recomendaciones precisas si no existe ningún usuario similar a mí.

En nuestro caso, no disponemos de productos que vender, ni de valoraciones de los usuarios de los proyectos que han consumido, ni descripciones de los proyectos que nos permitan asociarlos entre si dentro de categorías, como ya hemos comentado. Por este motivo, este segundo método de filtro colaborativo parece el más asequible para nuestros intereses, ya que lo que pretendemos es recomendar proyectos en base a lo que han consumido clientes parecido al cliente al que queremos recomendar.

Al tratarse de este método sería necesario una matriz que cruzase todos los clientes y proyectos de PwC, lo que supondría una matriz enorme, difícil de tratar, lo que suele ser uno de los mayores problemas de estos algoritmos. Para solucionar esto, este método recurre a la trasposición de matrices, como vemos en la Figura 4.32, donde se intenta descomponer la matriz principal R en dos matrices P y Q.

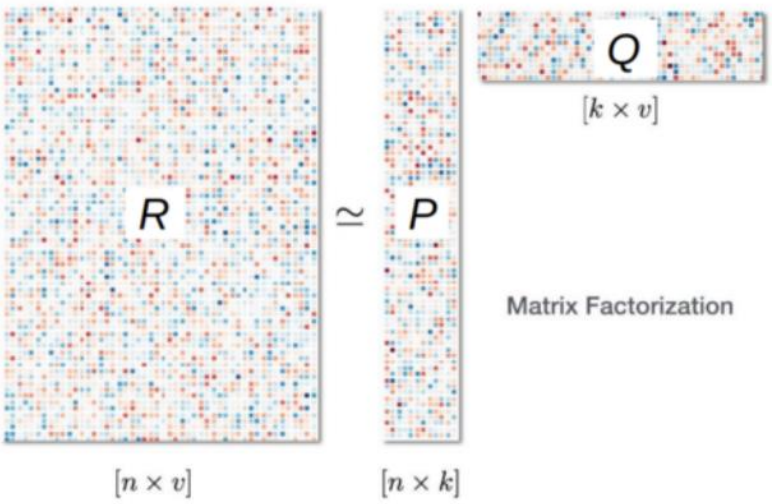


Figura 4.32: Trasposición de matrices de sistema de recomendación [9]

De esta forma se reduce significativamente la dimensionalidad del problema, algo que resulta clave en una empresa como PwC. De esta forma, obtendremos dos matrices, una de clientes que pretende simplificar las filas y que comprime la información de los clientes y otra de productos, que pretende simplificar la información de los proyectos, simplificando la información de las columnas.

$$L = ||R - P \times Q^T||_2 + \lambda \left(||P||_2 + ||Q||_2\right)$$

$$\hat{r}_{ij} = p_i^T q_j = \sum_k p_{ik} q_{kj}$$

$$argmin_{q,p} \sum_{i,j} (r_{ij} - p_i^T q_j)^2$$

Ecuación 4.5

Para lograr esta factorización de matrices solo es necesario desarrollar un modelo que sea capaz de construir estas dos matrices a partir de las principales mediante las ecuaciones de Ecuación 4.5, así, solo es necesario multiplicar una fila por la columna correspondiente para obtener el rating de una película, o en nuestro caso, si un cliente consume o no un proyecto.

Además, sería posible introducir una penalización en la función de pérdidas de ese modelo para tratar de conseguir que esas dos matrices sean lo más simples posibles para tratar de solucionar al máximo el problema de la dimensionalidad.

Por último, otra de las técnicas que conviene mencionar son las reglas de asociación. El proceso de encontrar objetos o, en este caso proyectos, que se consumen juntos de forma habitual se conoce como minado de relaciones o patrones y tiene un campo de estudio dedicado a ello exclusivamente.

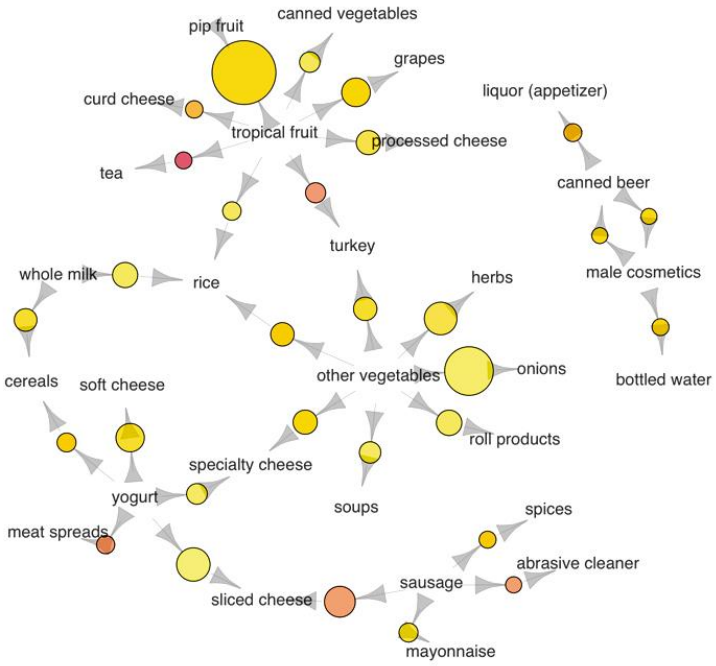


Figura 4.33: Example of association rules [10]

Estas relaciones se encuentran explorando un conjunto de datos de transacciones o consumiciones de productos, estas reglas de asociación se basan en dos conceptos, el soporte, que es con la frecuencia con la que se produce una transacción y la confianza que es la fortaleza de la relación. El objetivo es encontrar relaciones entre productos que se consumen de forma habitual juntos, es decir, que su soporte y confianza sea lo más alto posible.

Algunos de los beneficios de incluir estas relaciones es que son fácilmente interpretables y simples, lo que se valora muy positivamente desde un punto de vista de negocio.

5. Resultado Obtenidos

Durante este apartado vamos a realizar una descripción de los pasos de desarrollo que ha seguido nuestro proyecto durante los últimos meses. Para ello, seguiremos el orden que hemos llevado en el propio proyecto y en la sección anterior, comenzando por como realizamos la extracción de los datos y posteriormente los modelos mediante las técnicas ya comentadas.

5.1 Extracción de los datos mediante Alteryx

Este primer episodio del proyecto duro aproximadamente 1 mes y estaba comenzado una vez yo me incorpore a este proyecto.

Para nuestro proyecto necesitaremos los datos de una base de datos en las que se lleva registro de todas las ventas de la compañía a la que llamaremos WOP. Esta base de datos se actualiza por los managers de la compañía cada vez que logran una venta y se trata de una base muy amplia en cuanto a dimensiones y que nos planteara muchos problemas, no solo en la extracción sino en también en los posteriores modelos como veremos.

En primer lugar, el primer mensaje con el que nos enfrentamos que es en la base de datos existen múltiples tablas, en una de ellas se almacenan los datos del año fiscal 17, en el que se empezaron a tomar datos hasta el año fiscal 20, que es el último año que está cerrado; y, por otro lado, tenemos otra tabla con el año en curso, es decir, el año fiscal 21.

Esto supone el primero de nuestros problemas ya que necesitaremos lógicas independientes, ya que cada base de datos es independiente, y aunque, ambas son muy similares si tienen pequeños detalles que las diferencian y que nos obligan a elaborar lógicas diferentes.

Por simplicidad, decidimos elaborar varios módulos de Alteryx, lo que permitirá tener el trabajo granularizado y encontrar errores de forma más simple en una lógica tan completa. El primero de esos módulos consiste únicamente en leer los datos y eliminar las posibles mayúsculas y espacios que puedan contener como vemos en la Figura 5.1.

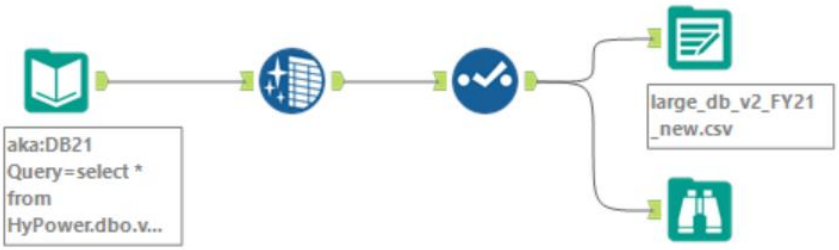


Figura 5.1: Primer módulo de extracción de datos

Este primer módulo lanza una query muy simple de tipo SELECT * FROM contra la base de datos de la compañía, de esta forma obtenemos todos los registros de ambas tablas y los podemos almacenar en un csv para poder manipular como queramos la tabla, la cual dispone de un total de 107 campos que habrá que seleccionar y manipular para obtener los campos y datos que nos interesen.

Al realizar esta extracción observamos que existen algunos otros problemas, ya que como sabemos, nuestra tabla de datos tiene un registro por cada proyecto que se registra, lo que suponen un gran número de registros a lo largo de los 4 años fiscales ya completados. Aun así, resulta poco probable que la compañía haya firmado un total de más de 3 millones y medios de registros a lo largo de esos cuatro años. Esto se debe a que los datos se almacenan en snapshots que se realizan de nuevo semanalmente y en cada uno de esos se incluyen los proyectos que ya han sido registrados en semanas anteriores.

Este problema no es especialmente complicado de resolver, ya que con un simple group by es posible solucionarlo, sin embargo, decidimos dejar esto para el código previo de los modelos, ya que dicha agregación puede depender de campos que aún no están creados o que hay que modificar aún.

Especialmente sensible para el primer modelo será el campo de la fecha en la que se hace registro del ingreso en cuestión, esto resulto un problema grande al descubrir que existían un total de siete campos que Alteryx identifica con formato fecha, como vemos en la Figura 5.2, muchos de estos campos son irrelevantes para nuestros modelos, ya que indican la fecha de creación, la fecha de cierre o la última fecha en la que se modificó el estado del proyecto.

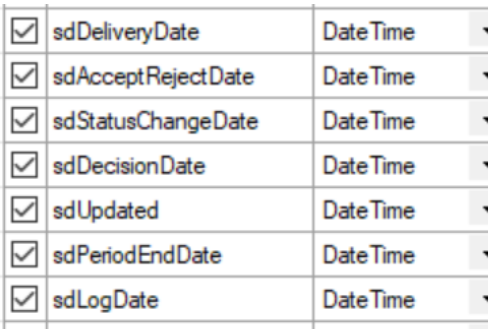


Figura 5.2: Campos identificados como DateTime por Alteryx

Aun así, si hay dos campos que pueden resultar útiles en el futuro, estos son los dos últimos que veíamos en la Figura 5.2, el campo sdPeriodEndDate indica el ultimo final de periodo, en este caso de mes, en el que se ha registrado el proyecto, por lo que este campo nos permite agrupar los registros por mes, mientras que sdLogDate indica la semana en la que se ha realizado el pago.

Estos dos campos parecen no tener sentido por sí mismos, ya que aún no hemos explicado que dentro de esta base de datos existen registros de tipo semanal y registros de tipo mensual, por

lo que será necesario distinguirlos y elaborar una lógica para tener un campo único de fecha teniendo esta premisa en cuenta.

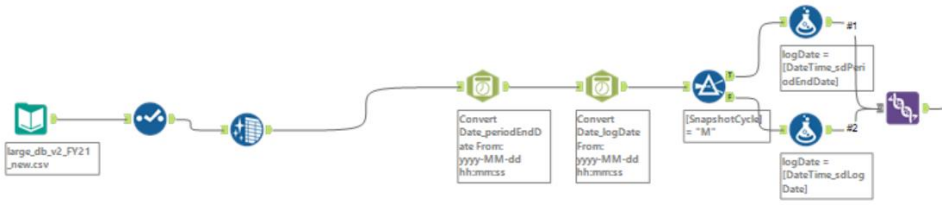


Figura 5.3: Lógica de limpieza de fechas en Alteryx

En la Figura 5.3 vemos cómo hacemos justo esto que comentamos en el inicio del segundo de los módulos que hemos desarrollado en Alteryx, comenzamos leyendo y limpiando los campos de la base de datos que hemos extraído en el primer módulo, para después convertir los dos campos de fechas que nos interesan al formato de fecha deseado y que mejor se adapta a nuestras necesidades, después de esto separamos el flujo de dato en dos mediante una lógica IF en la que usamos el campo SnapshotCycle, donde se especifica si el registro es de tipo mensual o no. Por último, creamos el campo de fecha definitivo logDate, donde tomamos la fecha adecuada en función del tipo de registro, para después juntar ambos flujos con el nuevo campo logDate.

La segunda parte de este módulo consistirá en crear nuevos campos que serán de gran importancia para ambos modelos, en primer lugar, el campo FY que determinara el año fiscal en el que nos encontramos, el FY_mes que determina el mes del año fiscal en el que nos encontramos y el FY_mes_nombre que da el nombre real del mes en lugar de un número. Como vemos en la Figura 5.4, estos tres campos se obtienen a partir de campo fecha que acabamos de crear, ya que recordamos que el año fiscal de la compañía comienza en julio por lo que el FY21 de la compañía comenzó en julio de 2020 y termina ahora en julio de 2021, siendo además julio el mes número 1.

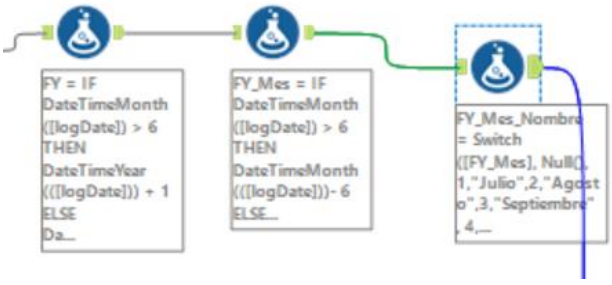


Figura 5.4: Creación de campos a partir de la fecha

Para finalizar este segundo módulo creamos un último campo que diferencia los de la empresa y volvemos a limpiar los campos, guardando el resultado en un nuevo CSV (Figura 5.5).

Figura 5.5: Final segundo módulo Alteryx

Por último, creamos un último módulo en el que los datos modificados se agrupan para evitar duplicados y en el que se reparten los datos a dos tablas diferentes, una mensual y la otra semanal. Para esto, el primer paso es volver a asegurarnos de que la fecha está en formato fecha correcto y posteriormente, agrupar los registros como vemos en la Figura 5.6 y la Figura 5.7.

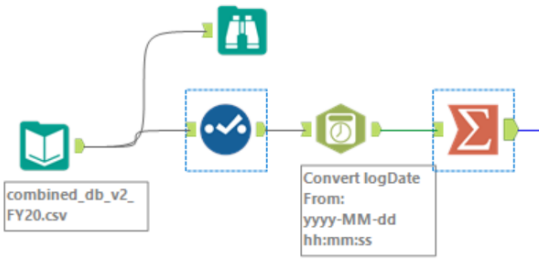


Figura 5.6: Inicio de ultimo módulo Alteryx

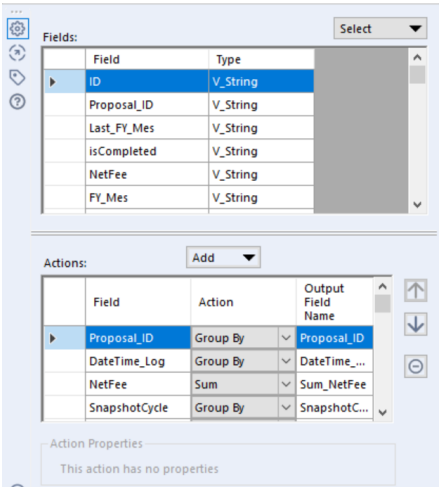


Figura 5.7: Agrupación para eliminar duplicados del mismo proyecto en una misma fecha

Una vez hecho esto, a posteriori, decidimos agregar dos campos nuevos, cada uno importante para uno de los modelos, el primero de ellos, hasAuditCode indica si el registro se refiere a una

empresa que está siendo auditada, esto es de vital importancia en el recomendador, ya que una empresa que está siendo auditada no puede recibir ningún otro tipo de servicio de consultoría por la misma empresa. El segundo campo es WeekNum que indica el número de semana en el año fiscal al que corresponde cada uno de los registros, clave en la serie temporal, cuyos intervalos de tiempo son semanales (Figura 5.8).

Por último, separamos de nuevo los campos por timo de registro, mediante un IF, con el campo SnapshotCycle para separar los registros mensuales de los semanales.

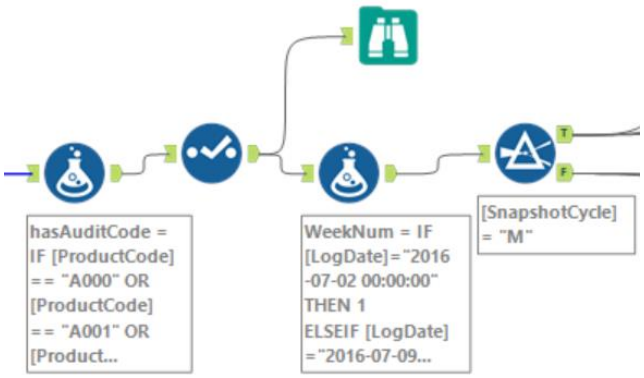


Figura 5.8: Creación de los campos hasAuditCode y WeekNum y separación por tipo de registro

Para finalizar este último modulo se guardan los datos definitivos en un CSV, tanto los mensuales como los semanales. Además, debíamos juntar ambas tablas que hemos obtenido entre los años 17 y 20 y el año 21 que va por libre al tratarse de una tabla que se actualiza cada semana, este paso lo hicimos en pandas y almacenamos el resultado en una tabla data_global con todos los campos que tenemos disponibles de ventas de la compañía desde agosto del año 2016 hasta la actualidad.

Una vez empezamos a emplear estos datos en los modelos que vamos a presentar a continuación este proceso completo resultaba bastante pesado, ya que cada semana sería necesario leer y aplicar las transformaciones necesarias a toda la tabla de año fiscal 21, el actual. Como sabemos esta tabla es bastante grande, ya que cada semana se acumulan los registros de todo el año, agregando en las semanas finales casi 50 mil registros y con un total de registros estos últimos meses superior al millón.

Por este motivo y debido a su baja complejidad, decidimos elaborar un delta de este Alteryx, es decir, que cada semana se extrajesen única y exclusivamente los registros de esa semana nuevos que pegaríamos a la tabla completa cada vez que realicemos una extracción. Para lograr esto solo era necesario modificar ligeramente la query de consulta a la tabla de datos, como se ve en la Figura 5.9.

```
select *
from HyPower.dbo.vwtblStatInfWOPDesdeFY21
where (HyPower.dbo.vwtblStatInfWOPDesdeFY21.sdPeriodEndDate > Convert(
datetime, '2021-06-06 00:00:00', 120))
  or (HyPower.dbo.vwtblStatInfWOPDesdeFY21.sdLogDate > Convert(datetime
, '2021-06-06 00:00:00', 120))
```

Figura 5.9:Query de extracción del nuevo archivo delta

Por otro lado, debemos leer y crear usar una función unión que permita pegar las dos tablas justo antes de la escritura en un csv del producto final, como se muestra en la Figura 5.10.

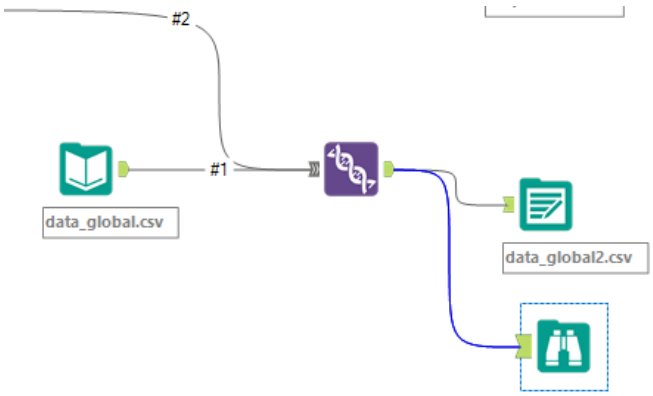


Figura 5.10: Union de tabla global con los datos nuevos semanales

Una vez hecho esto, ya tenemos una extracción semanal consistente que se puede realizar de forma rápida y que nos permite tener los datos actualizados para nuestros modelos en un breve periodo de tiempo y con los campos necesarios. Por esto, podemos decir que estamos listos para comenzar con los modelos que usaran estos datos.

5.2 Serie Temporal

5.2.1 Prophet

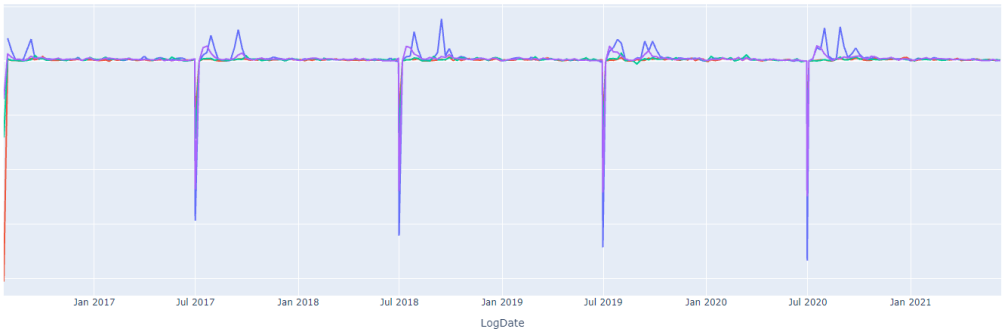
Como ya hemos explicado, en el primero de estos modelos vamos a usar la librería Prophet para la creación de una serie temporal de predicción de ventas de la empresa, en un principio esta parte del proyecto consistía en desarrollar 5 modelos de serie temporales independientes, uno por cada categoría de la empresa (por temas de confidencialidad no puedo revelar el nombre de estas categorías y al nombrarlas lo hare con numeros) y un último global, cuyos datos se obtienen como suma de los otros cuatro.

Lo primerio que debíamos hacer era agrupar los datos y extraer lo que realmente necesitamos de la tabla de datos original que tenía demasiados campos, ya que para el algoritmo de Prophet

realmente solo requerimos de 3 campos, la fecha o LogDate, el valor o NetFee y el LoS que usaremos para dividir los datos para cada modelo.

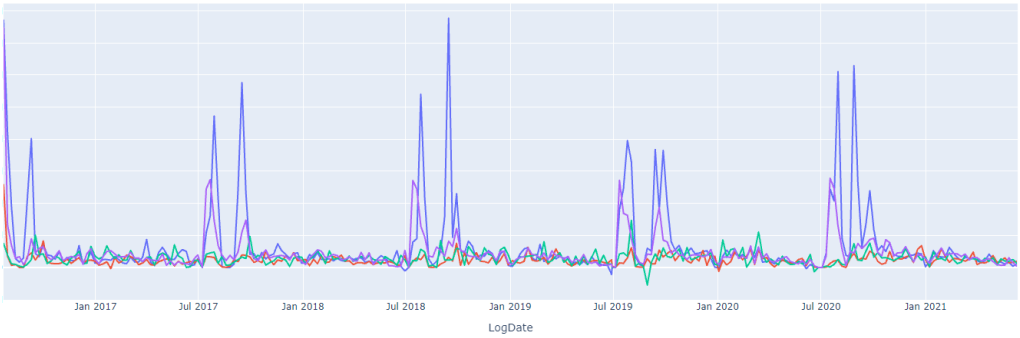
Gráfica 5.1: Ingresos por LoS agrupados

La Gráfica 5.1 muestra el resultado de agrupar los datos que extraemos mediante Alteryx por fecha y por categoría, como vemos, estos datos son agrupados y para el modelo con Prophet sería mucho más interesante obtener los datos sin acumular, el valor neto de cada semana, para poder así, obtener estacionalidades y tendencias más clara para los expertos de cara a negocio. Por lo que decidimos modificar estos datos mediante código con una simple función diff, obteniendo lo que vemos en la Gráfica 5.2



Gráfica 5.2: Datos de ventas de la compañía sin acumular por LoS

Ahora si vemos como estos datos tienen más sentido, no obstante, sigue habiendo un valor al principio de cada año fiscal que no parece normal por su gran valor negativo, ese valor se corresponde con el cambio de año, en el que el valor del año anterior se reinicia y al hacer la función diff queda muy negativo. Para solucionar esto, simplemente sustituimos el valor previo que teníamos acumulado por el valor que obtenemos al hacer la diferencia, ya que, al tratarse del primer día, el valor acumulado es igual que el valor de ese día (Gráfica 5.3).



Gráfica 5.3: Datos sin acumular y limpias para la serie temporal

Una vez hecho esto, ya estamos listos para comenzar a desarrollar nuestra serie temporal basándonos en la documentación de Prophet para poder así, llegar a un producto mínimo viable lo antes posible, del que después escalaremos para obtener el resultado final.

En esa primera iteración de nuestro modelo solamente teníamos dos funciones principales que habíamos hecho a partir de la documentación que provee Facebook sobre su librería Prophet, estas dos funciones eran una función de optimización en la cual se realizaba un tuneo de hiperparámetros para tratar de encontrar el modelo óptimo con los datos disponibles. Esta función simplemente crea una serie de combinaciones de parámetros que deseamos probar, después se elige un numero de optimización que creamos suficientemente elevado y se basa en la teoría de que es imposible probar todos los modelos, por lo que, elegimos el mejor de los probados, asumiendo que igual existe uno aún mejor, pero con no demasiada mejora disponible.

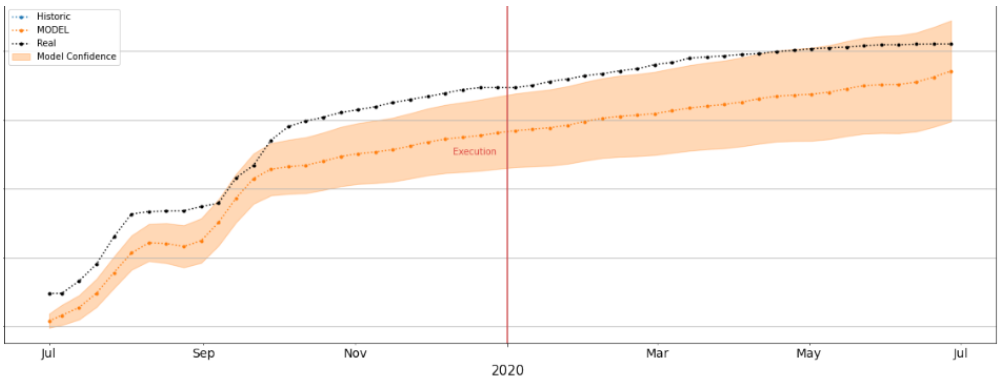
Como vemos en la Figura 5.11: Ejemplo de Facebook de función de optimización, Facebook proporciona una documentación sobre cómo realizar esta función que permite explorar los parámetros para poder así escoger el modelo más completo.

```python
# Python
import itertools
import numpy as np
import pandas as pd

param_grid = {
    'changepoint_prior_scale': [0.001, 0.01, 0.1, 0.5],
    'seasonality_prior_scale': [0.01, 0.1, 1.0, 10.0],
}

# Generate all combinations of parameters
all_params = [dict(zip(param_grid.keys(), v)) for v in itertools.product(*param_grid.values())]
rmses = []  # Store the RMSEs for each params here

# Use cross validation to evaluate all parameters
for params in all_params:
    m = Prophet(**params).fit(df)  # Fit model with given params
    df_cv = cross_validation(m, cutoffs=cutoffs, horizon='30 days', parallel="processes")
    df_p = performance_metrics(df_cv, rolling_window=1)
    rmses.append(df_p['rmse'].values[0])

# Find the best parameters
tuning_results = pd.DataFrame(all_params)
tuning_results['rmse'] = rmses
print(tuning_results)
```

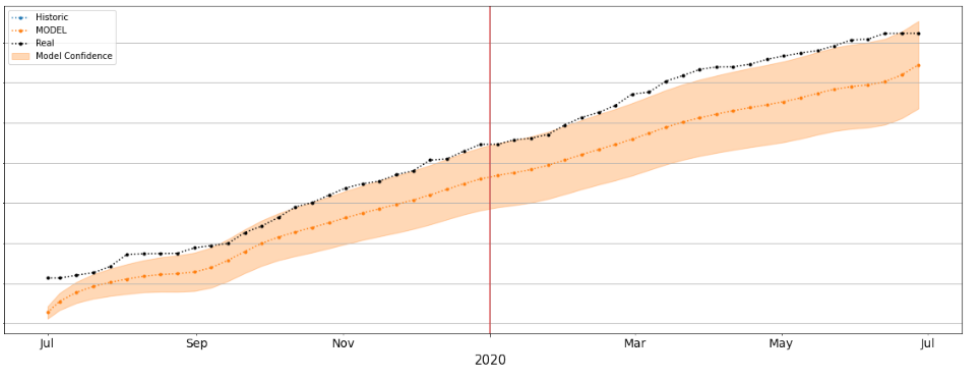
Figura 5.11: Ejemplo de Facebook de función de optimización [11]

La segunda función era una función train que simplemente escoge el mejor de los modelos que hemos probado, toma sus hiperparámetros y entrena un modelo con ellos del que después se elabora una predicción y se visualizan unos resultados.

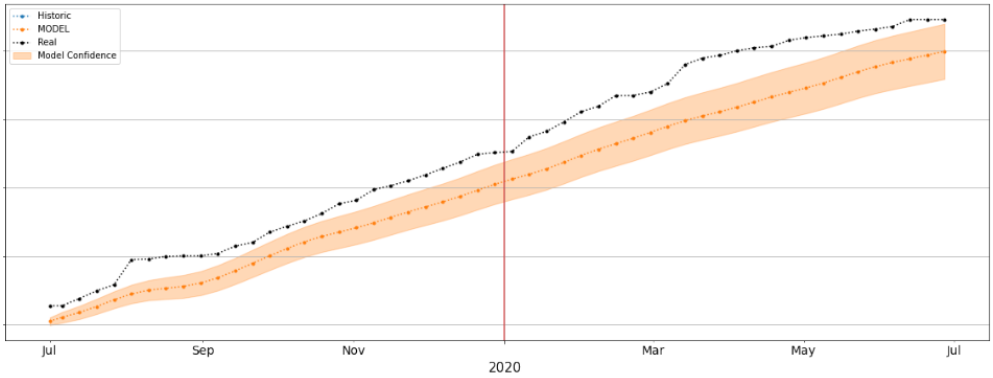
Con estas dos funciones somos capaces de elaborar las primeras predicciones durante la primera semana de marzo, realizando una predicción de año fiscal 2021 completa, desde julio, por lo que, disponíamos de medio año conocido, obteniendo los resultados que voy a mostrar a continuación:



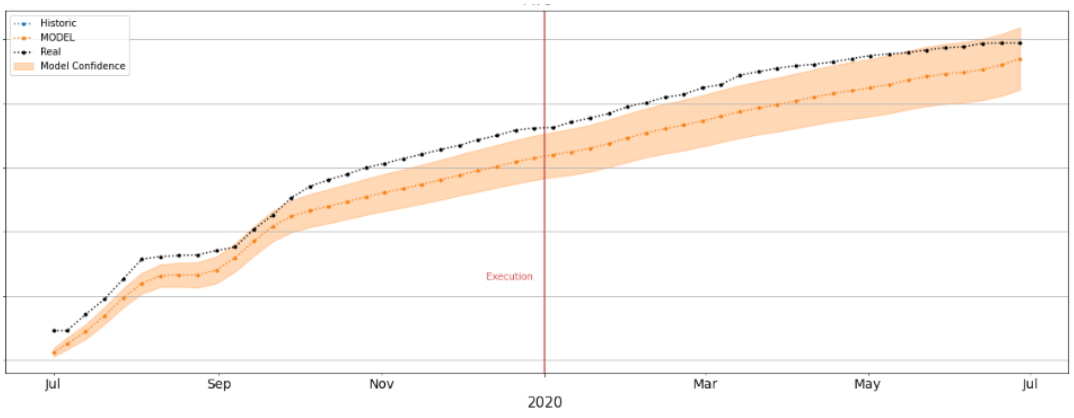
Gráfica 5.4 Resultados de primera predicción categoría 1



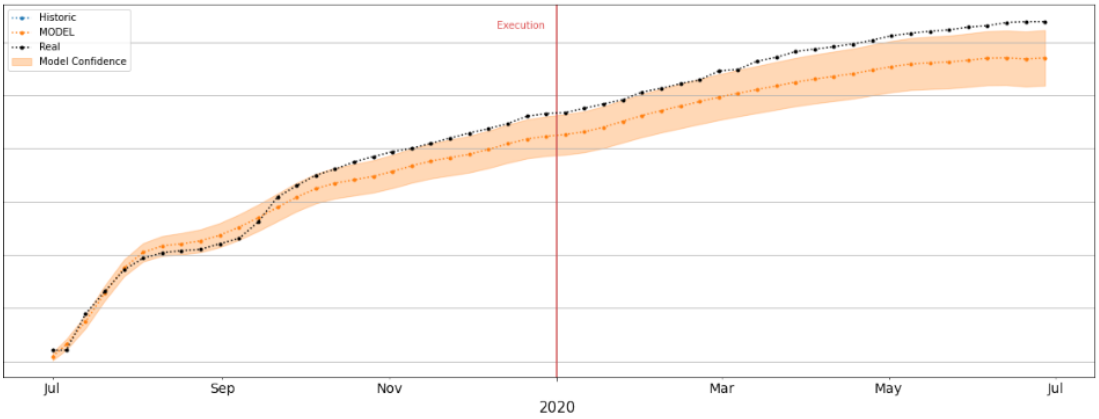
Gráfica 5.5: Resultados primera predicción categoría 2



Gráfica 5.6: Resultados primera iteración categoría 3



Gráfica 5.7: Resultados primera predicción categoría 4



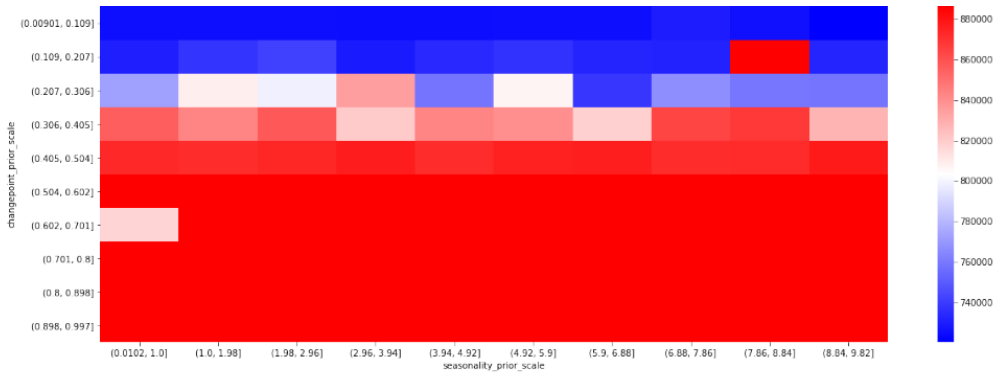
Gráfica 5.8: Resultados primera predicción categoría 5

Cabe aclarar que para la presente memoria solo se mostraran las predicciones del año completo ya que estas son las más complejas de realizar y las que realmente nos daban una idea de cómo de buenas eran la predicciones que estábamos realizando semanalmente. Por esto, y para evitar disponer de demasiada información repetida, solo se mostrarán los resultados de las predicciones de todo el año.

En estas primeras predicciones vemos como existen dos fallos que son extremadamente preocupantes. En primer lugar, estas predicciones anuales parecen que comienzan un escalón por debajo del valor inicial en el que deberían, y, en segundo lugar, parece que no es capaz de modelar correctamente las estacionalidades de los modelos, por lo que aún hay mucho margen de mejora en nuestro modelo, sus parámetros, etc.

Para solucionar estos problemas, decidimos en primer lugar, elaborar dos nuevos gráficos mediante la librería seaborn, la cual se especializa en realizar todo tipo de visualizaciones y que nos permitían interpretar de mejor manera los resultados.

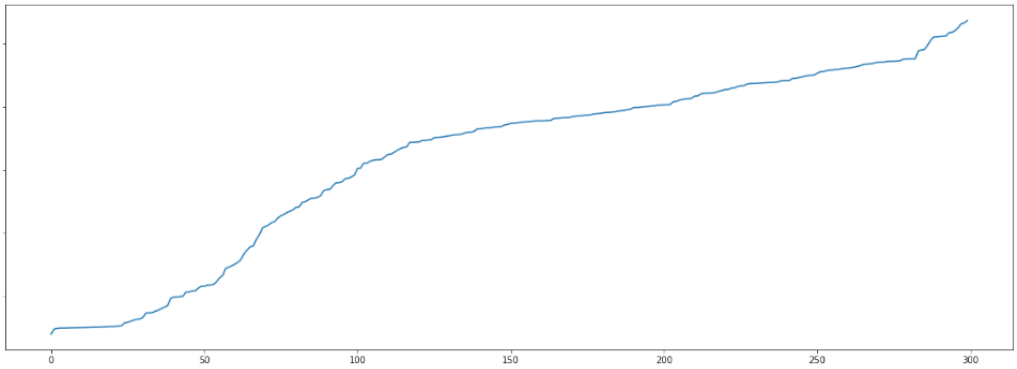
El primero de estos gráficos es un mapa de calor con los dos parámetros que estamos tuneando, changepoint_prior_scale, el cual puede ser quizás el más importante, ya que determina la flexibilidad de las tendencias, en concreto, cuanto cambia la tendencia cuando se encuentra con un punto de cambio de esta a lo largo del tiempo. Por su parte, el seasonality_prior_scale controla la flexibilidad de la estacionalidad, es decir, como permitimos que nuestra tendencia tenga grandes fluctuaciones. Un ejemplo de este grafico lo podemos observar en la Gráfica 5.9: Mapa de calor de la optimización de



Gráfica 5.9: Mapa de calor de la optimización de categoría 2

Como vemos de esta forma resulta muy evidente si la optimización de un modelo en concreto ha sido eficaz o no y en que zonas están siendo mejores los modelos por si interesase mover el rango de las variables hacia esa zona en una optimización futura.

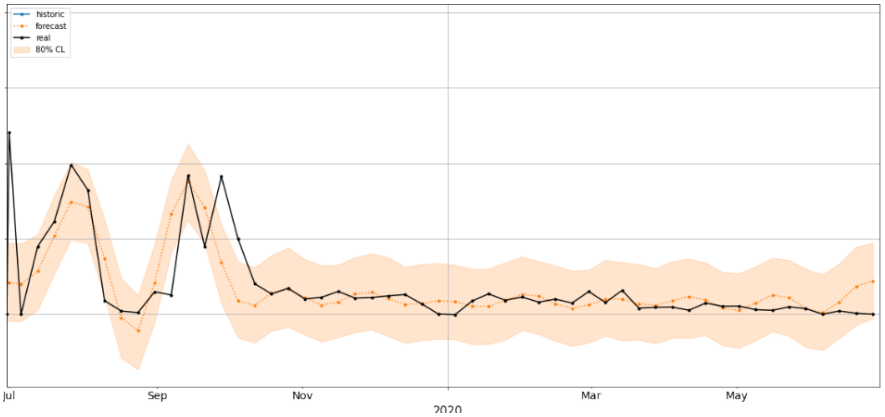
En segundo lugar, un gráfico de línea que represente los errores de los modelos en validación cruzada ordenados de mayor a menor (Gráfica 5.1), de esta forma es posible ver si la diferencia entre los modelos buenos es muy amplia, y sobre todo, si ha encontrado una serie de modelos en los que el error no mejora demasiado, lo que verificaría nuestra teoría de que, aunque no encontremos el mejor modelo, encontraremos uno cercano con un valor alto de optimizaciones.



Gráfica 5.10: RMSE de los diferentes modelos en la optimización de categoría 2.

Una vez hemos implementado estas dos mejoras en nuestro modelo para que resulte más fácil interpretar los resultados pudimos empezar a tratar de arreglar los dos errores que aparecían en nuestro modelo.

El primero de ellos resulta evidente y sencillo de solucionar ahora a posteriori, recordamos que parecía que las gráficas siempre comenzábamos por debajo del valor real de ventas, para explicar esto vamos a observar la Gráfica 5.11

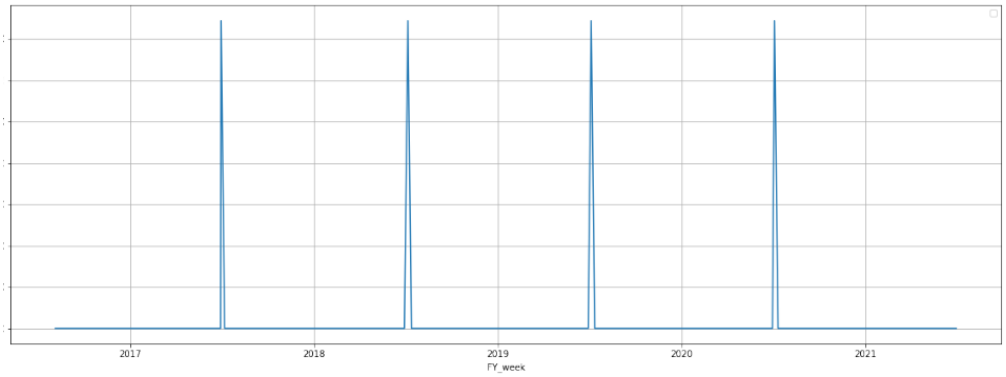


Gráfica 5.11: Resultados primera predicción sin acumular categoría 1

Si observamos el primer valor de esta grafica observamos como el valor de la predicción es significativamente inferior al valor real, mientras que por el contrario vemos como los valores finales son significativamente mayores. Para entender por qué ese primer valor era tan alto en todos los años tuvimos que recurrir a los expertos del negocio, que nos explicaron que en algunos años las últimas semanas del año fiscal se mueven a el primer día del año siguiente, por eso ese valor aparece tan alto, porque es el valor de tres o cuatro semanas en uno solo.

La solución a este problema tampoco resulto trivial, ya que lo primero que se ocurre es simplemente sumar el valor de ese primer día a la predicción, o, por el contrario, igualar ese valor a mano a 0. En principio, ambas soluciones deberían de ser perfectamente validas, sin embargo, ninguna de ellas parece la más correcta u ortodoxa, sin embargo, no éramos capaces de modelar ese comportamiento con las tendencias, ya que se trata de un valor extremadamente elevado y que sucede de forma repentina, por lo que no solo afectaba al comienzo de año, sino al final del anterior, donde el modelo trata de aumentar el valor para compensar ese pico.

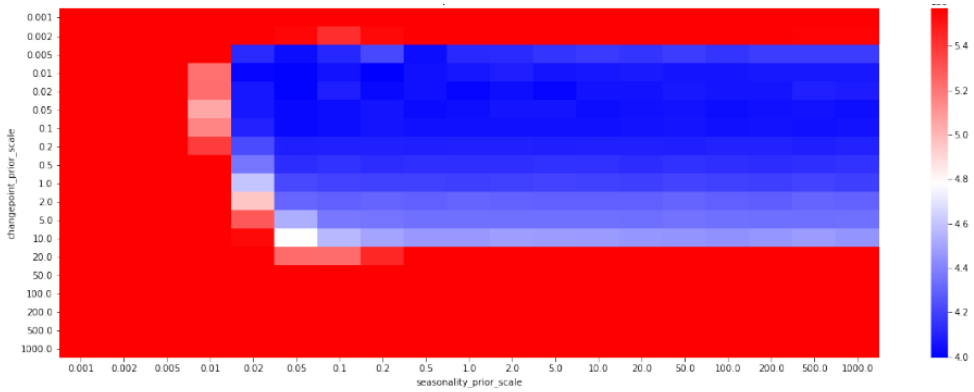
Finalmente, descubrimos que Prophet cuenta con una funcionalidad a la que llama holidays o vacaciones en español, la cual consiste en especificar al modelo una serie de días en los que el comportamiento del modelo es especial, como los fines de semana en una serie de demanda eléctrica, o la superbowl en una serie de espectadores de TV e EE. UU. A priori, esta funcionalidad parece que es justo lo que buscábamos, tenemos un día con un comportamiento especia y que se repite cada comienzo de año, lo que podemos crear unos holidays para cada día de comienzo de año, que permita al modelo saber que tiene que añadir un valor a esa primera semana, de tal forma que la predicción no sería solo la tendencia y la estacionalidad, sino que tendría una tercera componente que serían los holidays (Gráfica 5.12).



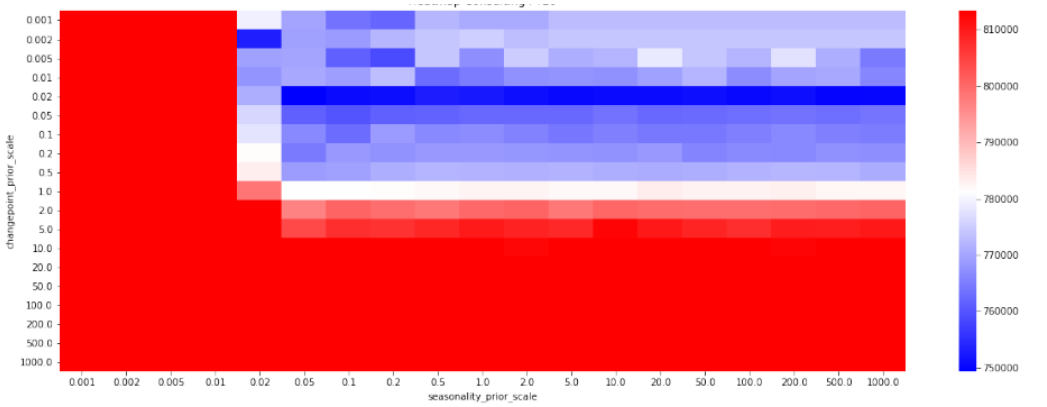
Gráfica 5.12: Grafica de holidays de categoría 1

Para tratar de mejorar las tendencias de nuestro modelo lo que decidimos fue llevar a cabo un estudio en profundidad de las dos variables que hemos comentado antes, de esta forma, podremos afinar mucho más la búsqueda de nuestro modelo optimo, lo que debería desembocar en una mejoría en el modelado de las tendencias de nuestro modelo.

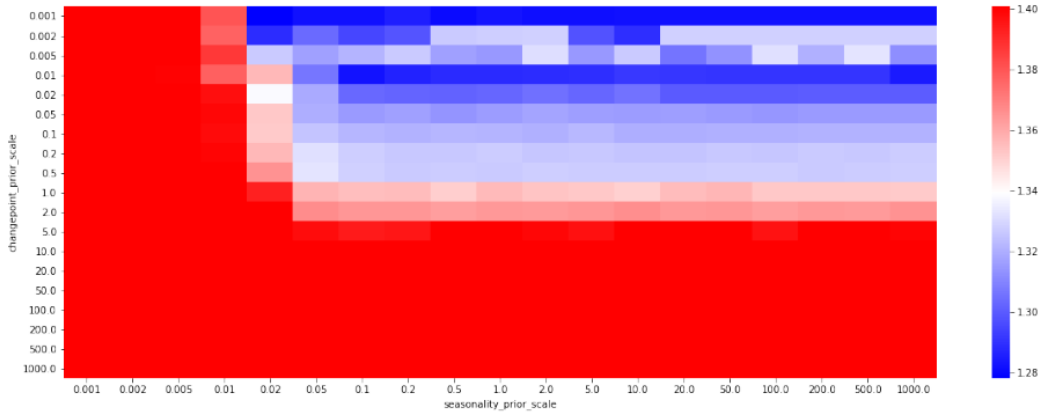
Para llevar a cabo esta segunda tarea, en primer lugar, realizamos una prueba de escala de la variable, en la que esta aumenta su valor de forma exponencial y así, resulta fácil evaluar en que rango de valores deben de estar las variables para cada modelo, lo que sería nuestro segundo paso, desarrollar un diccionario en el que se crea un rango de valores específico para cada uno de los cinco modelos. Los resultados de esta prueba de escala fueron los siguientes:



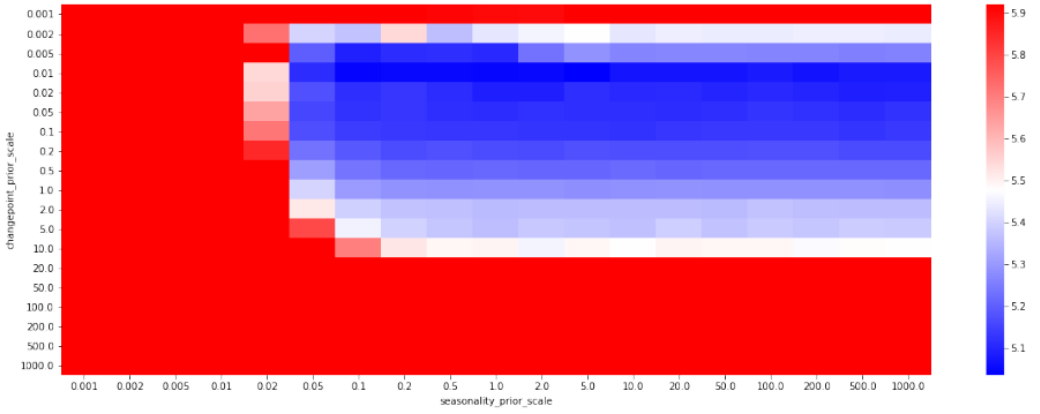
Gráfica 5.13: Prueba de escala categoría 1



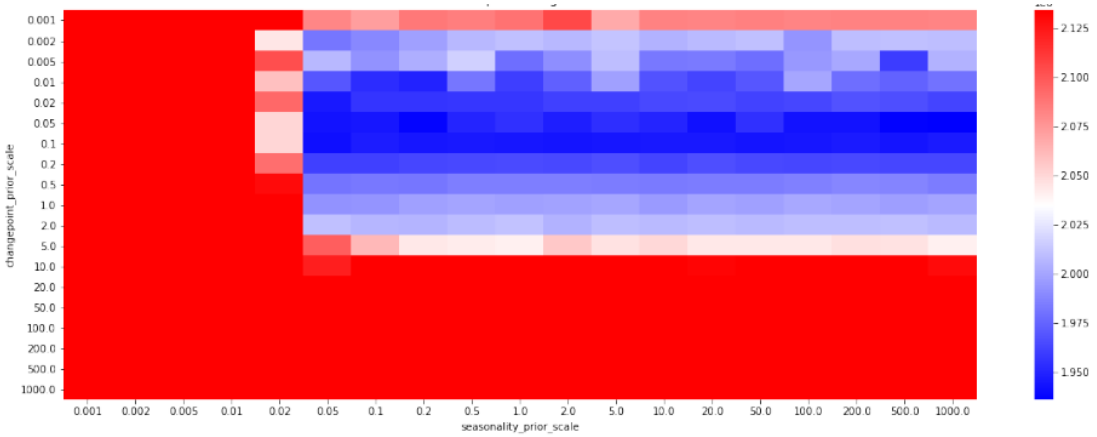
Gráfica 5.14: Prueba de escala categoría 2



Gráfica 5.15: Prueba de escala categoría 3



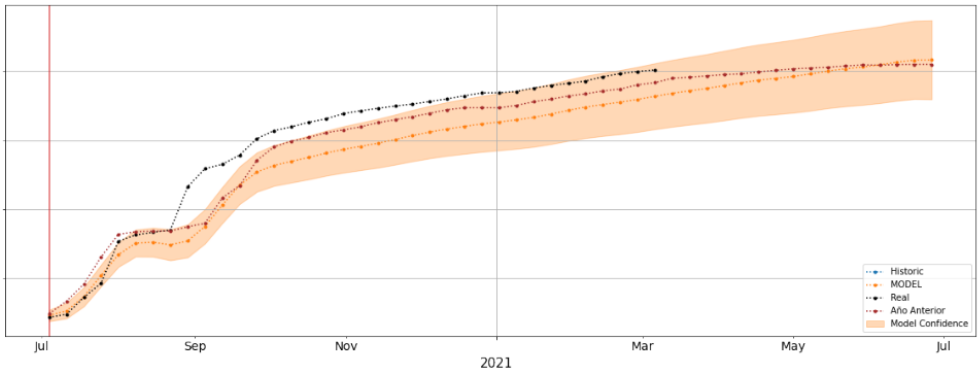
Gráfica 5.16: Prueba de escala categoría 4



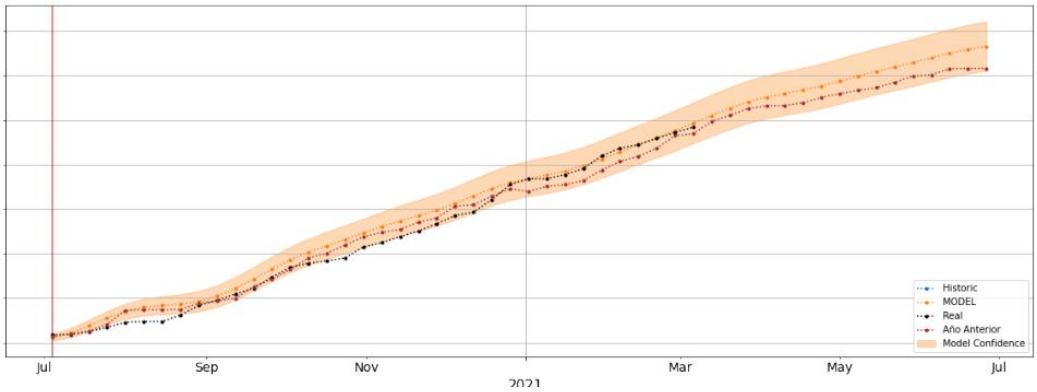
Gráfica 5.17: Prueba de escala categoría 5

Como vemos, estas pruebas han servido para justo lo que queríamos ya que en los cinco modelos es posible ver una zona clara en la cual el error cuadrático medio disminuye de manera clara y que podemos tomar como zonas de rango de tuneo de parámetros para cada uno de los modelos.

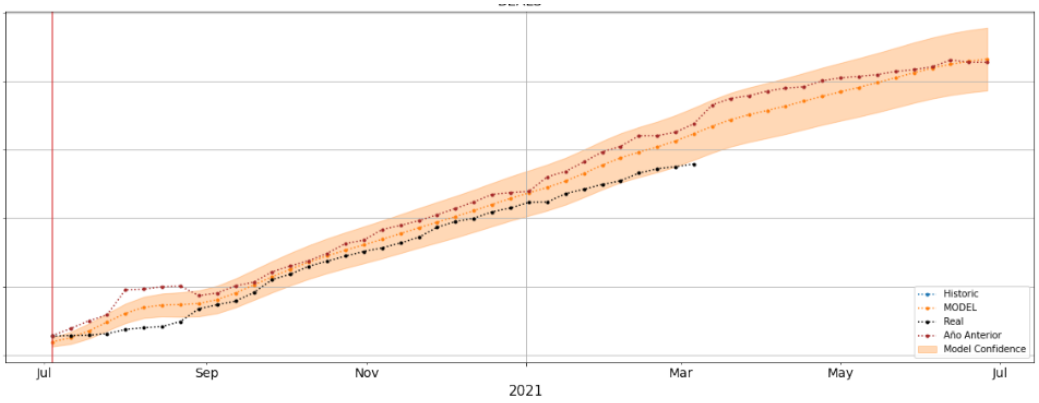
Una vez realizamos todas estas mejoras, realizamos una nueva prueba de predicción de todo el año fiscal 2021, obteniendo unos resultados mucho más satisfactorios como vemos a continuación:



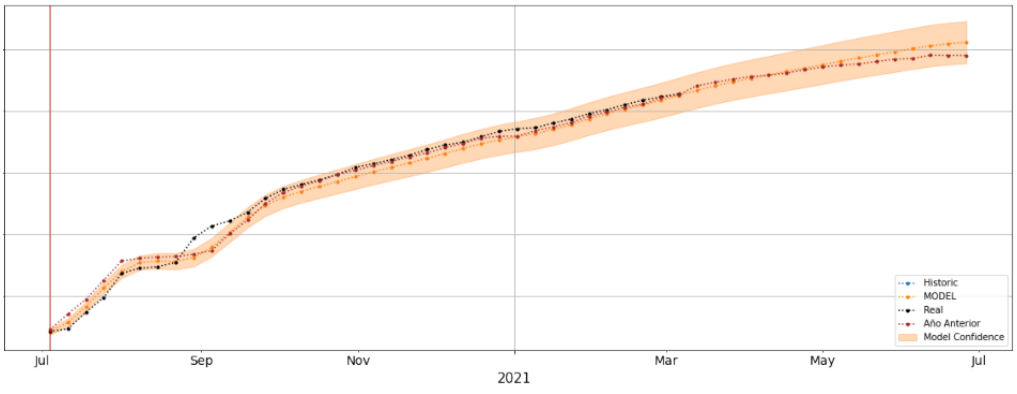
Gráfica 5.18: Resultados de predicción de la categoría 1 tras las mejoras



Gráfica 5.19: Resultados de predicción de la caegoría 2 tras las mejoras



Gráfica 5.20: Resultados de predicción de la categoria 3 tras las mejoras

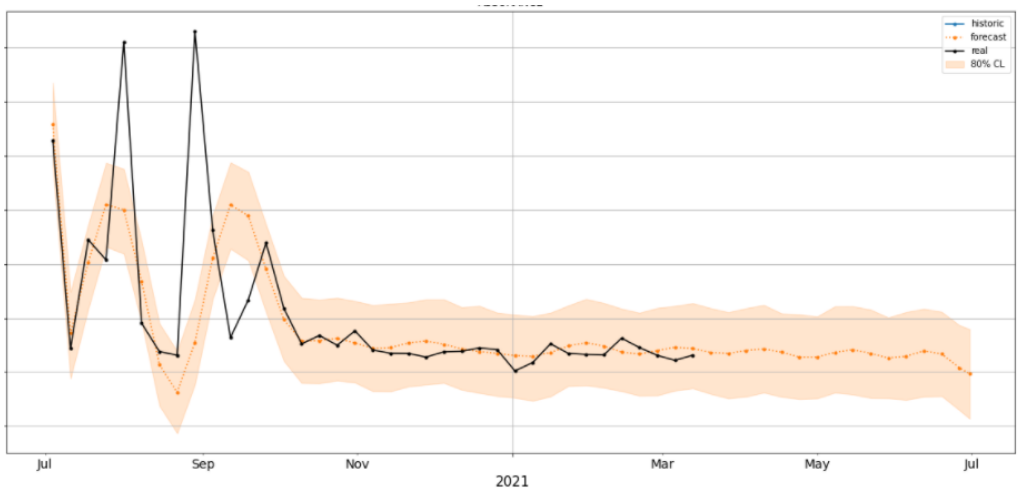


Gráfica 5.21: Resultados de predicción de la categoria 4 tras las mejoras

Gráfica 5.22: Resultados de predicción de la categoria 5 tras las mejoras

Podemos observar una clara mejoría en general en todos los modelos, si bien hay algunos que parecen que incluso se podrían considerar definitivos como la categoría 3 o la categoría 2, otros como la categoría 4 parecen estar casi bien a excepción de alguna tendencia que no modela correctamente, y por último, la categoría 1 es sin duda el que peor desempeño conseguimos, ya que no conseguimos modelar bien la estacionalidad anual ni aun después de las mejoras realizadas, a pesar de que el valor final de acumulado es correcto, y por lo tanto, nuestra predicción valida, pero se observa claramente como el modelo no tiene el comportamiento deseado, ya que no modela la tendencia correctamente la tendencia al comienzo de año y lo compensa al final, por lo que el acumulado del error es cero, pero el error es alto todo el año.

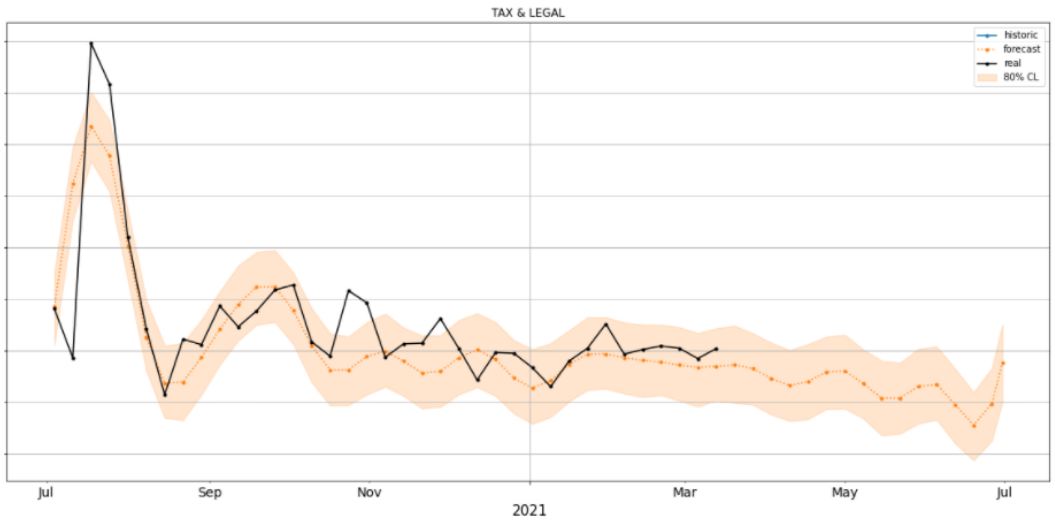
Para hallar una solución a este problema, de nuevo, debemos recurrir a la gráfica del valor no acumulado, el cual observamos en la Gráfica 5.23.



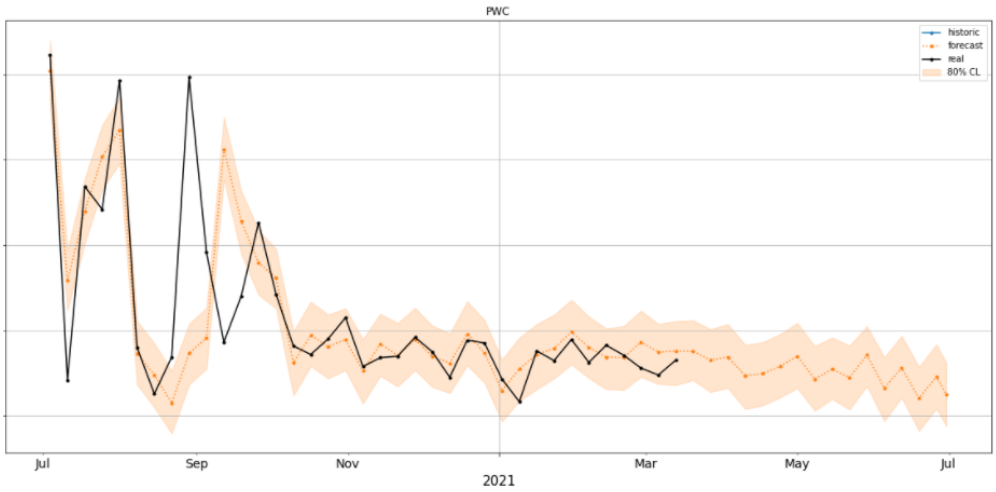
Gráfica 5.23: Predicción mejorada la categoria 1 sin acumular

Como vemos en esta gráfica, las ventas de la categoría 1 son muy descompensadas a lo largo del año, ya que tiene dos periodos de firmas de proyectos, en los consigue vender dos o tres proyectos realmente importantes que le suponen un gran ingreso en esos picos que observamos a principio de año, sin embargo, el resto del año las ventas son considerablemente menores y la gráfica se estabiliza. Desde un punto de vista del modelo, esto quiere decir que tiene que permitir fluctuaciones muy grandes en ciertos momentos del año, no obstante, estos picos no se producen siempre en la misma semana del año, sino que suelen variar una semana arriba o abajo.

Esto supone un problema, puesto que como vemos el modelo no predice un valor mucho más grande a los demás, sino que predice tres valores mayores que de lo normal. Esto tampoco debería de suponer un problema, ya que si así fuese y la predicción fuese correcta nuestro modelo se desviaría únicamente las tres semanas en la que se produce el pico, lo que sería aceptable, pero nuestro modelo esta desviado todo el año, lo que quiere decir que no somos capaces de modelar correctamente las fluctuaciones. Algo similar ocurre con categoría 5 (Gráfica 5.24) y con la categoría 4 (Gráfica 5.25)., pero no en tanta medida.

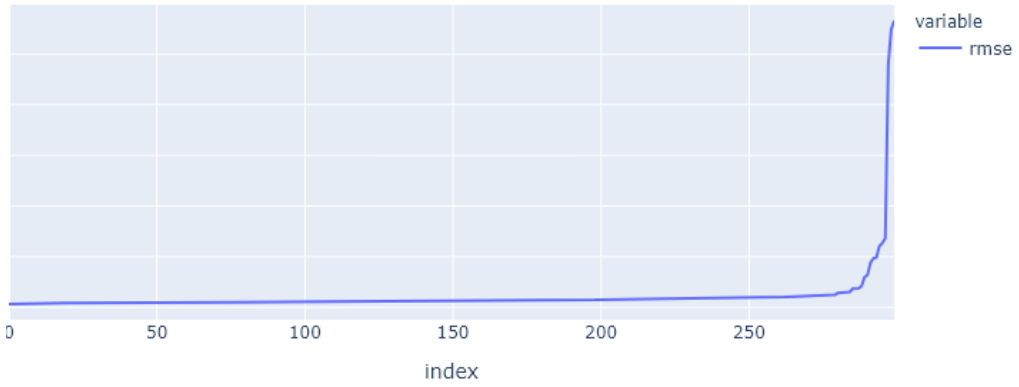


Gráfica 5.24: Predicción mejorada categoría 5 sin acumular

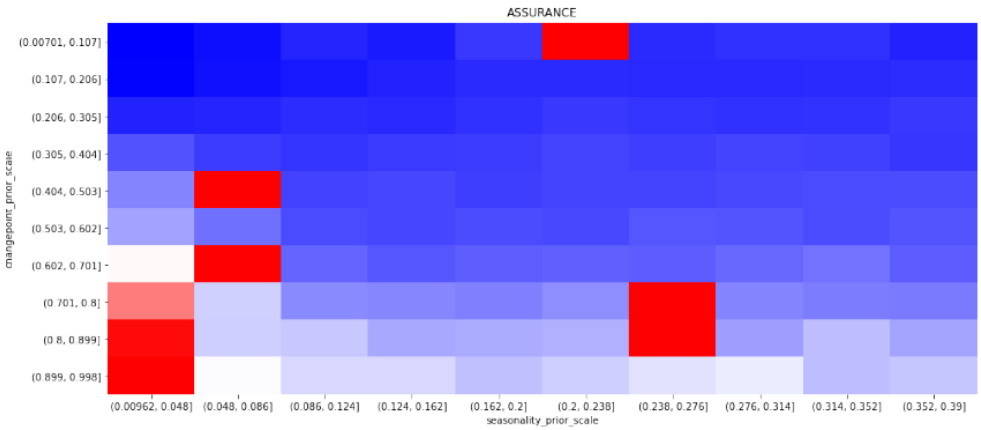


Gráfica 5.25: Predicción mejorada categoría 5 sin acumular

Viendo estas tres graficas de los modelos que no llegan a ser del todo perfecto parece que el problema ocurre con la estacionalidad anual, la cual nuestro modelo no logra representar de forma satisfactoria debido a las fluctuaciones. Este problema se puede deber al seasonality_prior_scale, parámetro que como hemos dicho antes modela en qué medida permitimos a nuestro modelo tener fluctuaciones, sin embargo, como hemos indicado, ya hemos ajustado los rangos óptimos de cada parámetro para cada modelo individualmente y este rango parece estar acotado correctamente como vemos en las gráficas del error de la categoría 1 en la Gráfica 5.26 y la Gráfica 5.27.



Gráfica 5.26: RMSE de los modelos de optimización de categoría 1

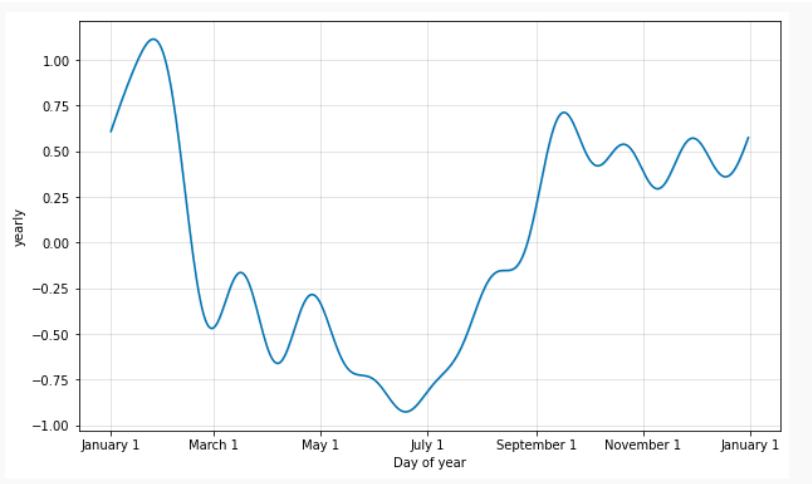


Gráfica 5.27: Heatmap de las variables de optimización de categoría 1

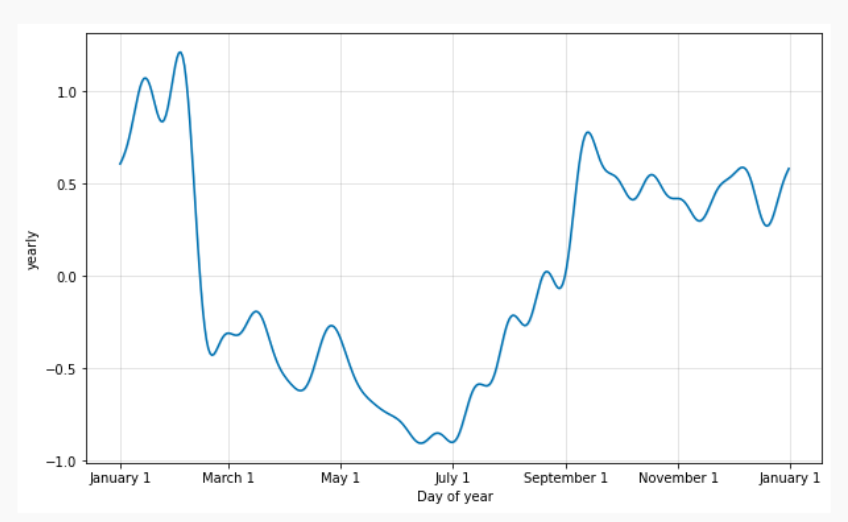
Vemos como nuestro rango de optimización está perfectamente acotado y como los modelos apenas difieren en su error, siendo la mayoría de ellos perfectamente validos como modelo final. Estos resultados nos llevaron a buscar otro parámetro que pudiese ser el responsable de que el modelo no sea capaz de modelar ese comportamiento.

Tras muchas pruebas para solucionar este error y tras muchos fracasos intentando modelar estos comportamientos con holidays, al igual que el punto inicial, dimos con una posible solución, como hemos visto, las estacionalidades de Prophet se construyen a partir de una suma parcial de Fourier, por defecto este orden de Fourier, indica el número de términos en la suma parcial, por lo que también indica como de rápido puede cambiar la estacionalidad.

Sabiendo esto, resulta bastante lógico pensar que nuestra estacionalidad no es capaz de modelar estas fluctuaciones debido a que no hay suficientes términos en nuestra serie de Fourier y una posible solución a este problema sería aumentar el valor de 10 (Gráfica 5.28) a 20 términos (Gráfica 5.29) en la suma de términos de la serie.



Gráfica 5.28: Estacionalidad construida a partir de 10 términos de Fourier [4]

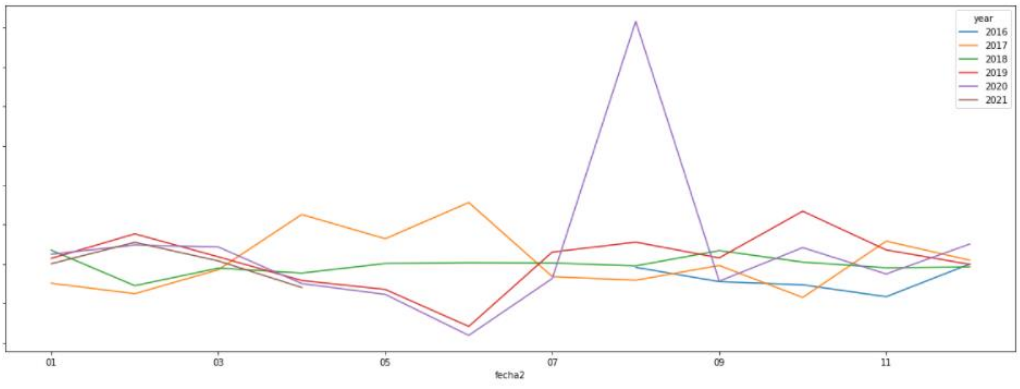


Gráfica 5.29: Estacionalidad construida a partir de 20 términos de Fourier [4]

Como vemos en estas dos últimas gráficas, aumentar el número de términos que conforman nuestra suma de Fourier provoca una curva con unos picos mucho más pronunciados, que parece ser una posible solución a nuestro problema, aun así, un valor excesivamente alto también puede provocar un sobreajuste, por lo que habrá que hallar el valor optimo en cada uno de los modelos.

Como se especifica en la documentación de Facebook, este es un parámetro que no conviene tunear como hacemos con los otros dos, ya que para modelar la estacionalidad ya está el seasonality_prior_scale, por lo que tener ambos parámetros tuneándose al mismo tiempo puede ser claramente contraproducente, y por esto, decidimos probar este parámetro en cada modelo en diferentes optimizaciones hasta obtener el valor óptimo de cada uno de ellos.

Una vez realizado este cambio solo nos faltaba una cosa por explicar, las tendencias eran incapaces de modelar la caída de tendencia que tenían tanto el modelo de la categoría 1 como el de la categoría 4, ya que nuestro modelo siempre daba mayores ventas en estas últimas semanas del año, como se ve claramente en la Gráfica 5.21, sin embargo, esto no se debía a un fallo de nuestro modelo o a una mala optimización, ya que ese resultados tenía sentido.



Gráfica 5.30: Datos reales de la compañía por año desestacionalizado

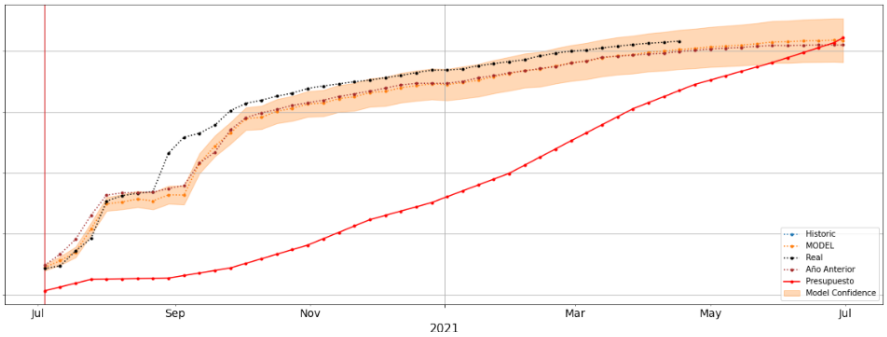
En la Gráfica 5.30 observamos un gráfico de líneas en los que se representas los ingresos de la compañía, los cuales han sido desestacionalizados, primero quitándoles la tendencia global, dividiendo cada valor por la media de los valores de la compañía, y después se han dividido cada mes por la media de los valores de cada mes, para quitar así la estacionalidad anual de la serie.

En este gráfico, el eje X representa los meses del año en orden normal, no por año fiscal, por lo que los meses que debemos observar son el 4, 5 y 6 para poder dar una explicación al problema que comentábamos anteriormente. Vemos como en estos meses los años 2019 y 2020 tienen unos ingresos claramente inferiores a 2019 y, sobre todo a 2018. Por esto decíamos que nuestro modelo tenía sentido, ya que, al hacer una media de las estacionalidades, el final de año es mayor que el de los últimos dos.

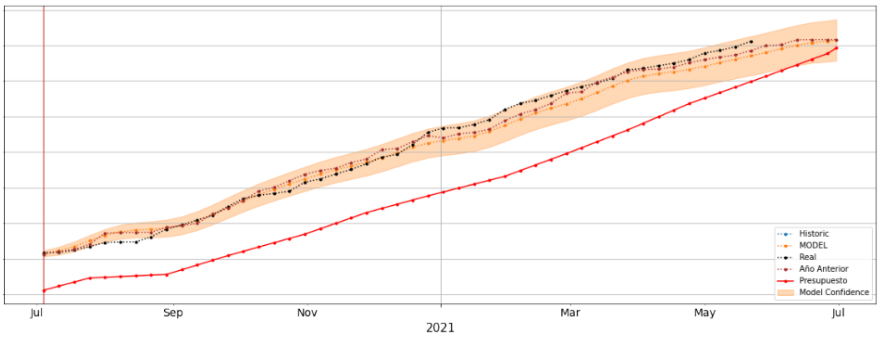
Una vez que hemos aclarado esto, solo falta determinar si el comportamiento habitual es el de los últimos dos años, o si, por el contrario, nuestro modelo tiene razón y ese comportamiento no tiene por qué repetirse.

Para poder responder a esta pregunta es necesario saber cómo funcionan los datos que estamos usando, ya que como hemos comentado, son los managers de la compañía los que apuntan, por así decirlo, las ventas que realizan en la base de datos. Por este motivo, estos dos últimos años los ingresos han sido menores, porque la gente al ver que los objetivos se iban a cumplir con creces deja de archivar la información a final de año, para así poder cargarla al año siguiente y tener un seguro en el futuro.

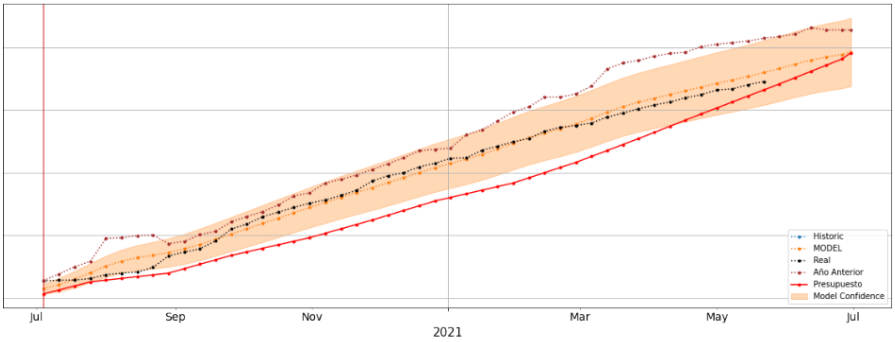
Al ver esto, decidimos tomar la solución de reducir los datos de entrenamiento, entrenando únicamente con los años 2018, 2019 y 2020, puesto que los dirigentes de la compañía nos aconsejaron seguir este procedimiento, asegurando que este era un comportamiento que se iba a repetir consistentemente. Realizando todos estos campos obtuvimos los siguientes resultados:



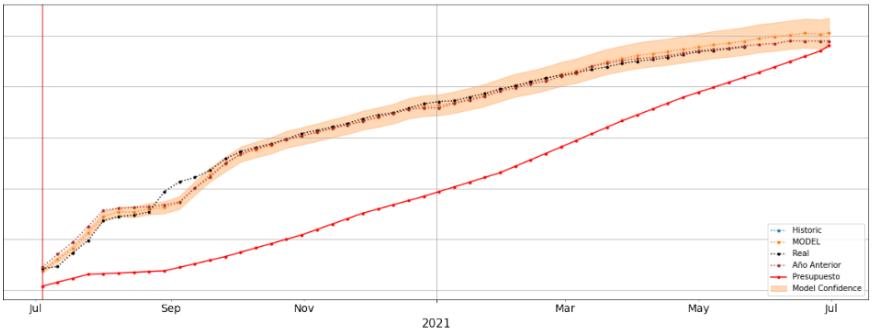
Gráfica 5.31: Predicción de ventas de la categoría 1 tras Fourier y reducción de datos



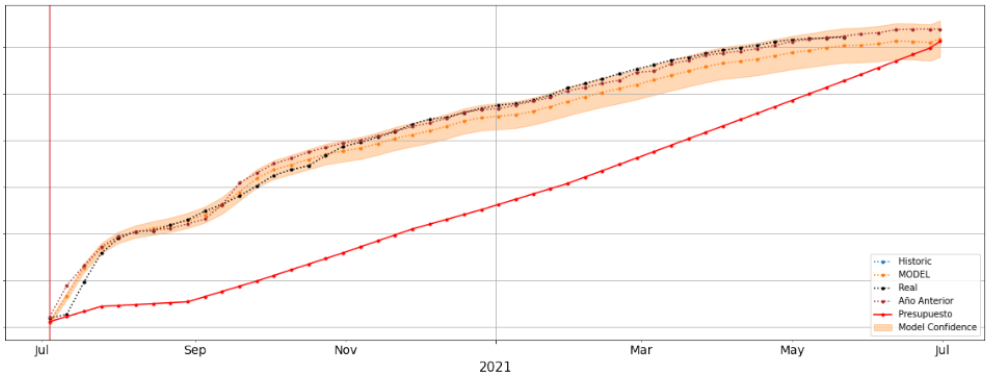
Gráfica 5.32: Predicción de ventas de la categoría 2 tras Fourier y reducción de datos



Gráfica 5.33: Predicción de ventas de la categoría 3 tras Fourier y reducción de datos



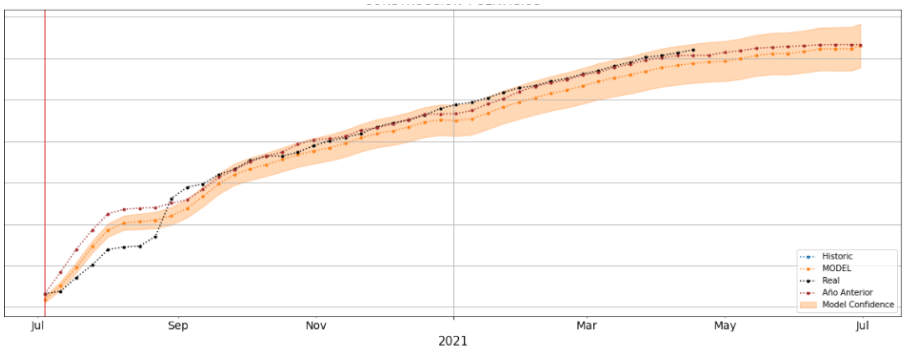
Gráfica 5.34: Predicción de ventas de la categoría 4 tras Fourier y reducción de datos



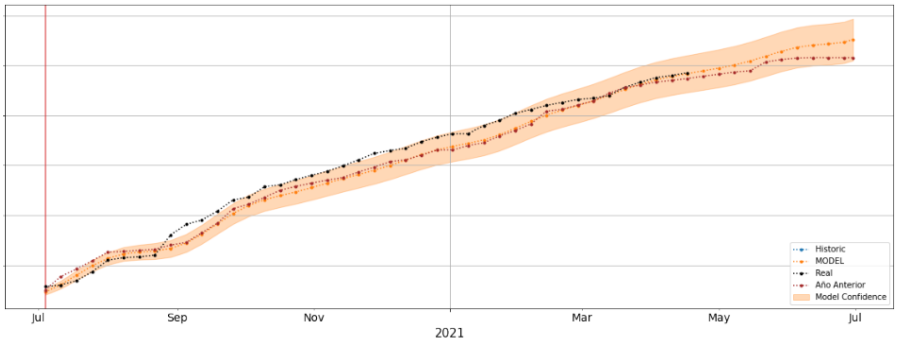
Gráfica 5.35: Predicción de ventas de la categoría 5 tras Fourier y reducción de datos

Como podemos observar, estas predicciones han mejorado considerablemente con respecto a las anteriores, siendo ahora muy satisfactorias en los 5 modelos, por lo que podemos dar el proceso de optimización y desarrollo del modelo por finalizado.

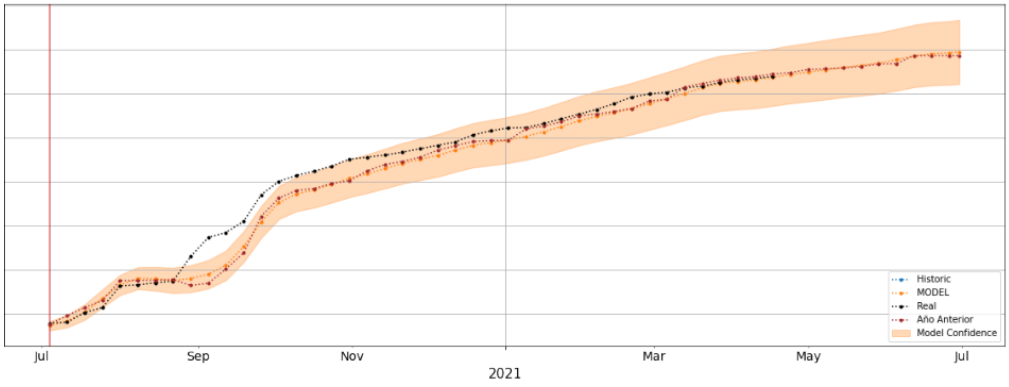
Posteriormente a todo este proceso que hemos comentado se fueron sucediendo una serie de mejoras, todas ellas guiadas por dirigentes de la compañía, los cuales nos daban su opinión sobre el proyecto cada dos semanas y nos hacían sugerencias para mejorar y obtener el producto final que tenemos ahora. Entre estas mejoras están la creación de una serie de modelos extra, ya que nos pidieron que, si podíamos desarrollar estos mismos modelos, pero por sector de la compañía, para hacerlo seguimos los pasos ya comentados, pero de una forma mucho más rápida, ya que no fue necesaria tanta exploración hasta lograr los siguientes resultados:



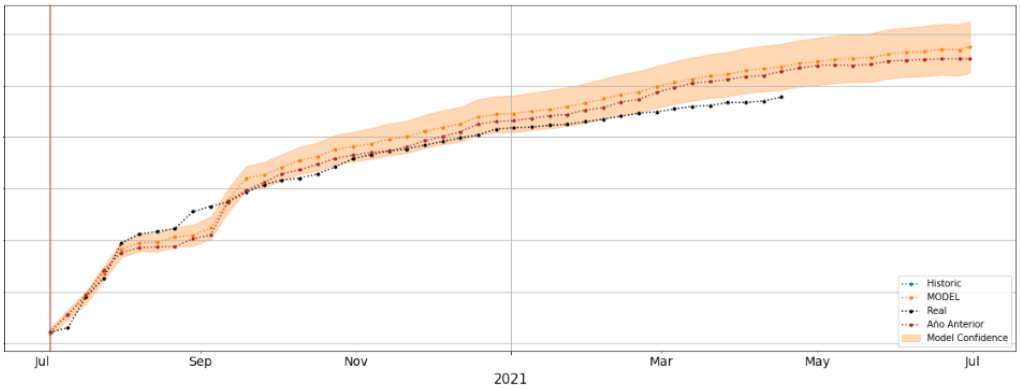
Gráfica 5.36: Resultado modelo sector 1



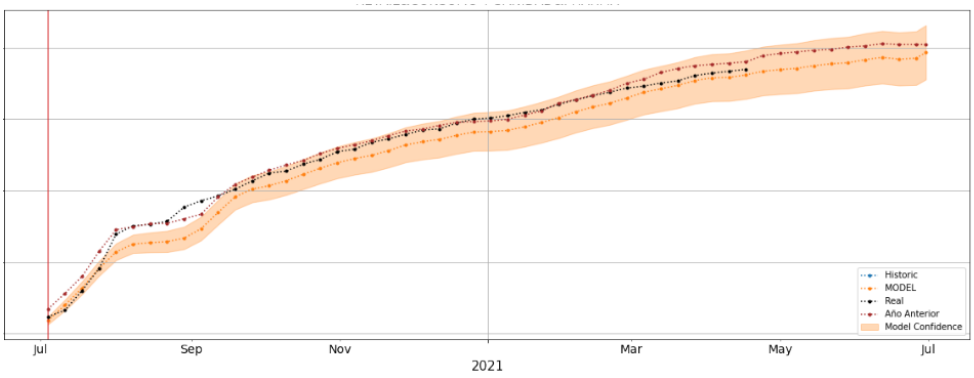
Gráfica 5.37: Resultado modelo sector 2



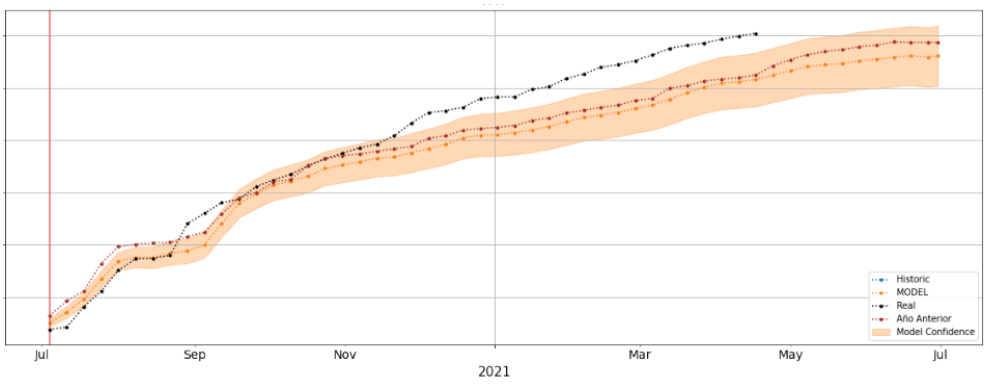
Gráfica 5.38: Resultado modelo sector 3



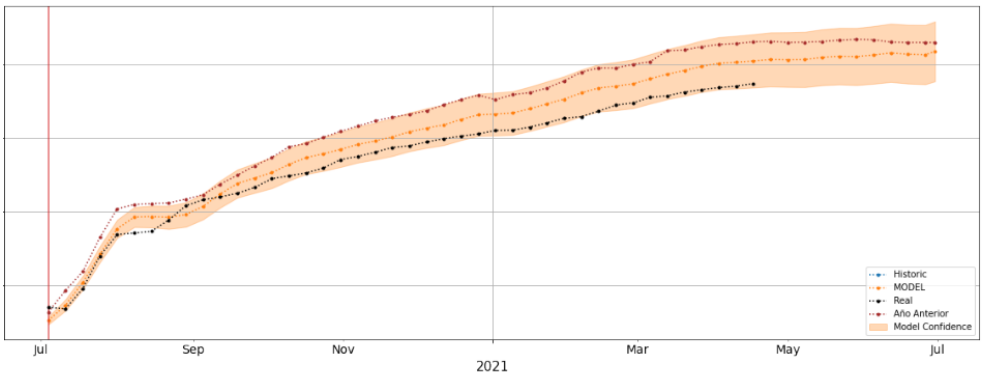
Gráfica 5.39: Resultado modelo sector 4



Gráfica 5.40: Resultado modelo sector 5



Gráfica 5.41: Resultado modelo sector 6



Gráfica 5.42: Resultado modelo sector 7

El resto de las mejoras fueron de cara a la parte de visualización de los resultados, la cual explicare en el siguiente apartado, pero entre ellas esta, desarrollar una comparación entre nuestras predicciones y el presupuesto que elabora la compañía de forma interna, la elaboración de un presupuesto para el siguiente año fiscal, con el actual en curso, pero únicamente a partir de abril y como hemos visto en las gráficas, la inclusión de nuevos datos como son los presupuestos o los años anteriores.

5.2.2 Power BI

A la vez que se desarrollaba estos modelos se desarrollaba una herramienta de visualización para que se pudiesen interpretar los resultados de los modelos de forma fácil y sencilla. Esta visualización se trata de un tablón desarrollado con la herramienta de Windows Power BI y que trataba de ser una herramienta interactiva con la que fuese posible sacar el máximo partido a nuestras predicciones.

En un principio nuestro tablón de visualizaciones incluía solo un par de pestañas, una para la visualización de la compañía por LoS (Figura 5.12) y otra para la visualización de las predicciones, con un seleccionable de él LoS que queríamos visualizar y la fecha de predicción (Figura 5.13), lo que nos permitía visualizar todas las predicciones de cierre de año que habíamos realizado. Como vemos a continuación, estas visualizaciones no son especialmente complejas, aunque cumplían su función de representar los resultados que íbamos logrando con el modelo.

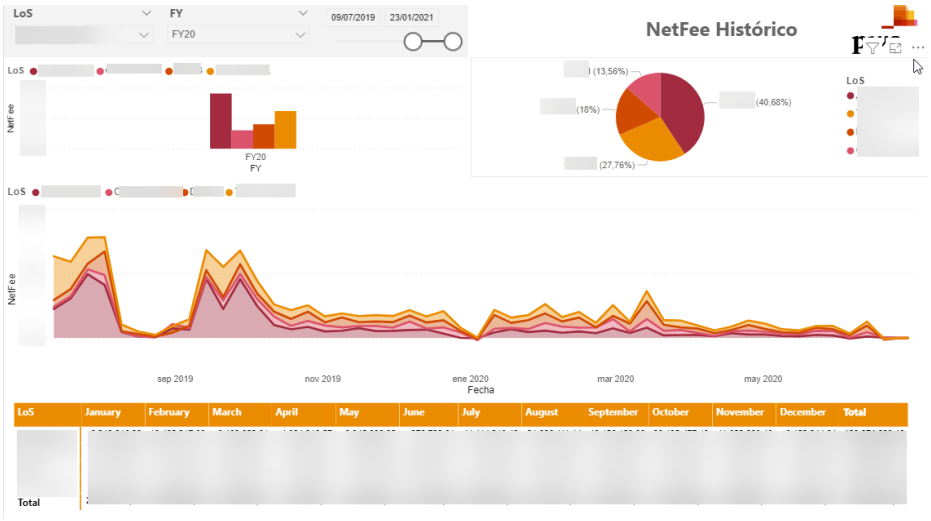


Figura 5.12: Pestaña de datos históricos en primer tablón que realizamos

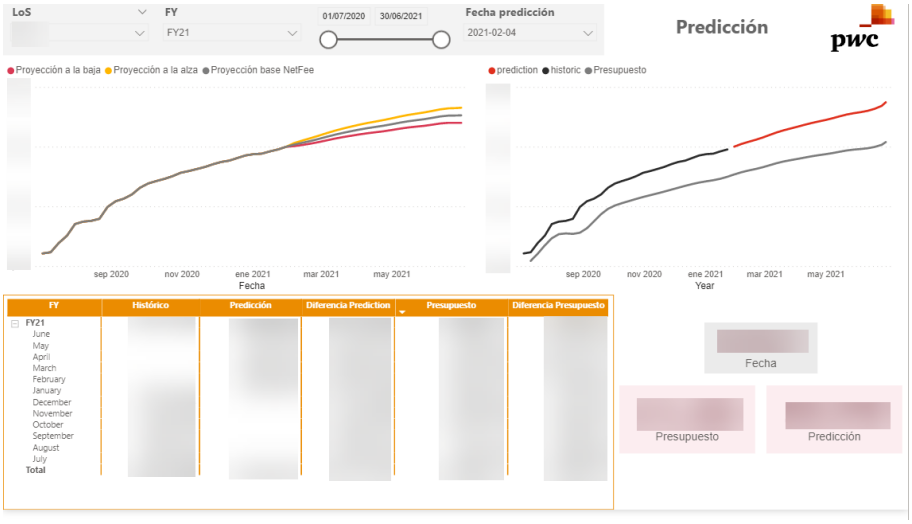


Figura 5.13: Pestaña de visualización de predicciones primer tablón

Como vemos, este primer tablón que elaboramos sirve como visualización, que es su objetivo, pero hay muchas cosas que se podrían añadir incluso en estas dos visualizaciones para hacerlo mas completo y atractivo desde el punto de vista de negocio.

Poco a poco dejamos atrás esa primera iteración del tablón, desarrollando sobre todo la pestaña de las predicciones, ya que era la mas importante a la hora presentarlo y para la interpretación de negocio, en definitiva, es de la pestaña de la que se pueden sacar mas conclusiones y la que mas interesaba cada vez que presentábamos el proyecto.

Por este motivo, poco a poco, a la vez que avanzaba el modelo lo hacían también las visualizaciones, añadimos KPIs para hacerlo más atractivo, porcentajes de crecimiento y decrecimiento hasta obtener la visualización que observamos en la Figura 5.14.

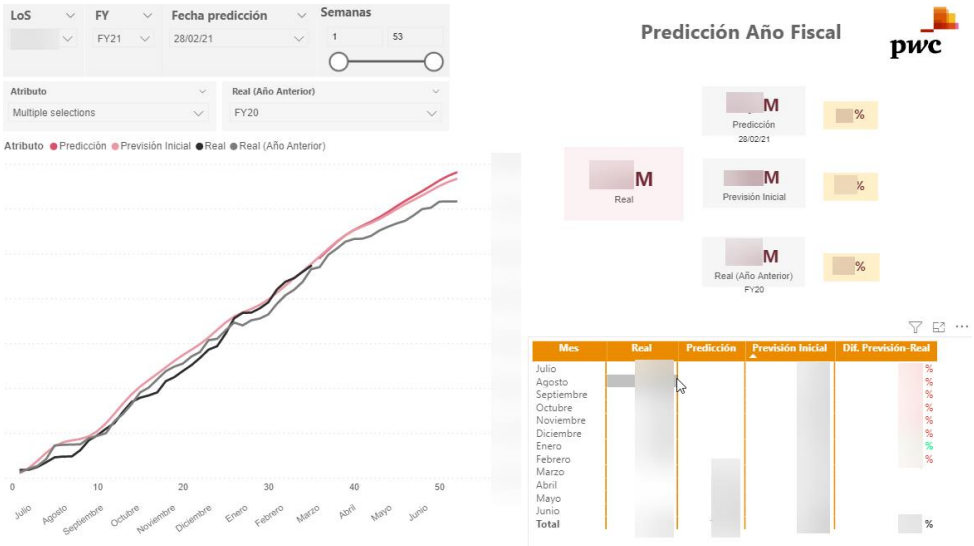


Figura 5.14: Pestaña de predicciones tras los nuevos cambios.

Como vemos, cabe destacar que incorporamos tres nuevos KPIs con los valores más relevantes para que estuviesen de forma visible y fácilmente interpretable. De esta forma en rojo tenemos el valor real del año fiscal en la semana de predicción, y en gris tenemos tres medidas, la predicción de final de año, la predicción que establecimos a principio de año y el real del año anterior global, junto a ellos está el porcentaje que supone el real de este año con respecto a esas tres medidas.

Esta pestaña se parece mucho más al resultado final que tenemos actualmente, no obstante, aún existen un par de cambios que realizamos más adelante, ya que nos parecía que los KPIs no eran del todo significativos, aunque aportan mucha más información de forma visual que la anterior pestaña, sigue sin ser del todo significativo, ya que por la forma en la que esta implementado no resulta del todo significativo y no permite identificar si las predicciones son positivas o no.

Una mejora significativa fue la incorporación de esta misma pestaña, pero sin los datos acumulados (Figura 5.15), lo que permitía realizar estas mismas comparativas, pero para cada uno de los meses en lugar de para todo el año, lo que permite identificar si las predicciones para un mes en concreto son positivas o si no lo es.

Para la realización de todos estos valores visuales y todos los porcentajes usamos las métricas de Power BI que nos permiten seleccionar valores determinados de una columna mediante sumas, máximos y otras operaciones. Para aplicar los filtros que nos interesan creamos las tablas maestro a partir de nuestras tablas de histórico y de resultados que extraemos del modelo, de esta forma podemos filtrar de forma dinámica las tablas a la vez que lo hacen los usuarios con los filtros y seleccionar los valores que nos interesan en función de esos filtros.

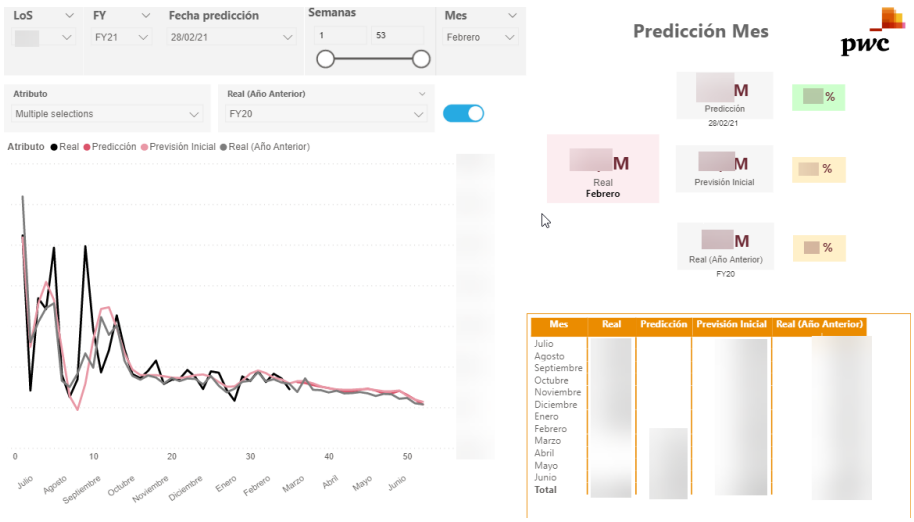


Figura 5.15: Pestaña de visualización de predicciones sin acumular

A medida que iba avanzando el proyecto modificamos los KPIs como hemos comentado, ya que no eran del todo intuitivo y añadimos algunos más. Siguiendo las predicciones de los expertos de la compañía con los que nos reuníamos para exponerles nuestro trabajo, creamos un primer KPI

grande que destaca sobre los demás, en el que se compara el valor real con nuestra predicción para final de año y con respecto a la misma semana del año anterior, para poder saber de forma fácil que porcentaje total llevamos cubierto y como se desarrolla el año con respecto al anterior.

En pequeño tenemos los dos KPIs que ya conocemos y dos nuevos con datos internos de la compañía, la ambición, es decir, lo que a la empresa le gustaría vender este año; y el presupuesto, lo que se espera vender este año. Todos ellos se comparan con la predicción de nuestro modelo, de esta forma tenemos las medidas de predicción, el real del año anterior y nuestra predicción a principio de año, todas ellas comparadas con la predicción de la semana actual, para así poder interpretar como de positivo o negativo está siendo el año actual (Figura 5.16).

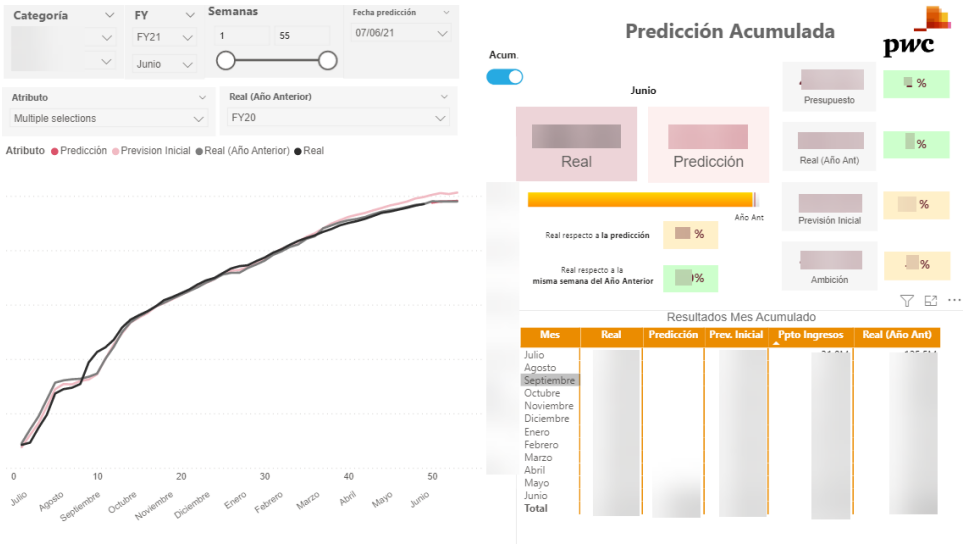


Figura 5.16: Pestaña de visualización de predicciones definitiva

También añadimos la posibilidad de seleccionar una categoría entre LoS o Sector y dentro de cada uno el que se desea seleccionar, ya que recordamos que añadimos siete nuevos modelos que deben de poder visualizarse en el tablón definitivo.

Por último, añadimos un botón azul que se puede observar en la figura donde es posible seleccionar si se desea ver los resultados acumulados o sin acumular y la visualización cambia automáticamente entre una y la otra.

Como hemos comentado en el apartado anterior, hubo una serie de extensiones que añadimos a nuestro modelo para poder realizar nuevas visualizaciones que aportasen valor a la compañía. Una de estas extensiones es que a partir del mes de abril el modelo calcula de forma automática el presupuesto del año siguiente, de tal manera, que es posible visualizar como nuestro modelo cree que se va a desarrollar el año siguiente. En esta visualización del presupuesto del año fiscal siguiente, representamos el año actual y la predicción de nuestro modelo para el año siguiente, y se nos pidió añadir la posibilidad de representar el acumulado del año actual, con un porcentaje de incremento seleccionable al que llamamos presupuesto de Naive (Figura 5.17).

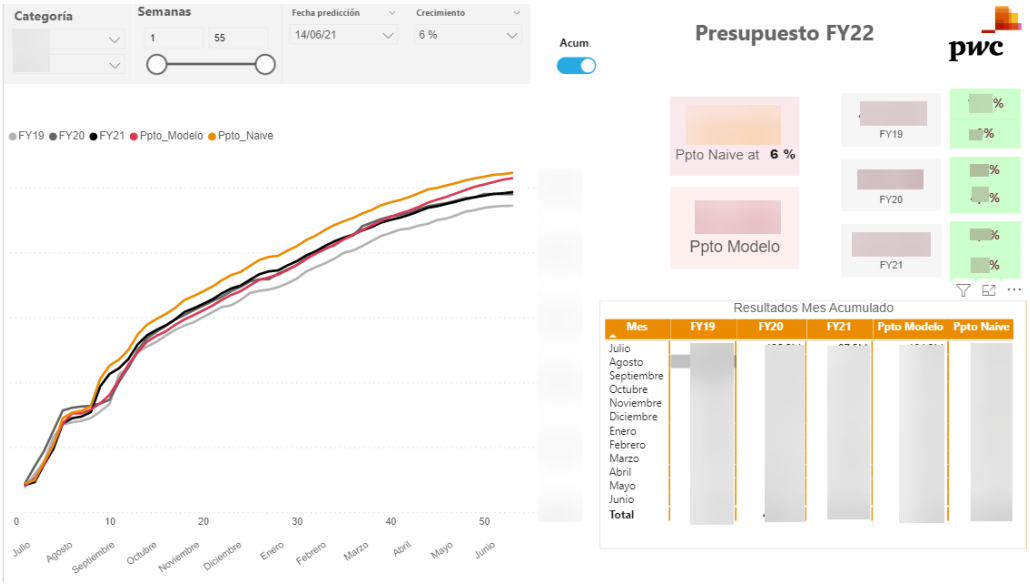


Figura 5.17: Pestaña de presupuestos año siguiente

Esta pestaña también incluye una comparación de los dos modelos con los tres años anteriores, teniendo así un porcentaje de crecimiento y decrecimiento con cada uno realizado también mediante medidas de Power BI. Esta pestaña también ofrece la posibilidad de visualizar los resultados sin acumular, mediante el botón azul de la visualización, obteniendo la Figura 5.18.

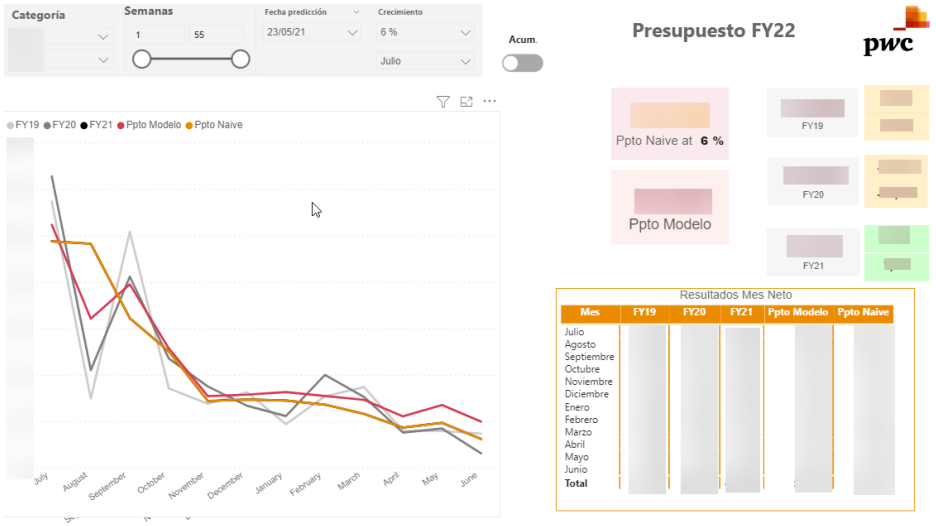


Figura 5.18: Pestaña de presupuestos sin acumular

Por último, se añadieron dos pestañas más que permitían de forma sencilla comparar las predicciones del modelo con la previsión de ventas que hacen de forma interna en la compañía (Figura 5.19) y otra pestaña para comparar las propias predicciones entre sí (Figura 5.20).

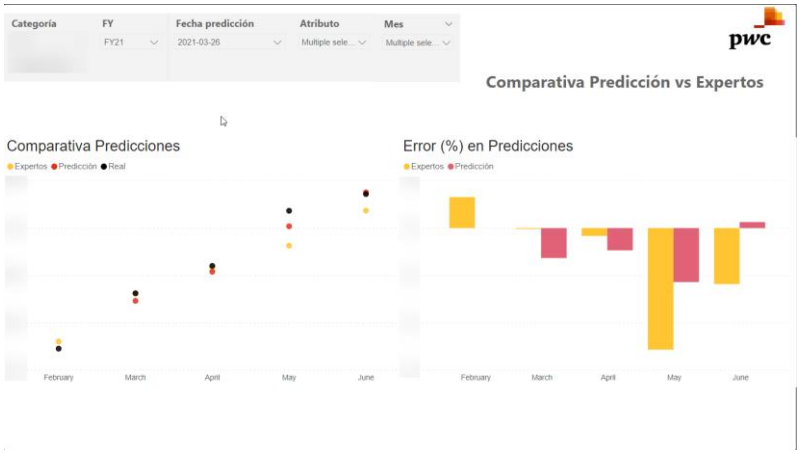


Figura 5.19: Comparativa de predicciones con presupuesto de expertos

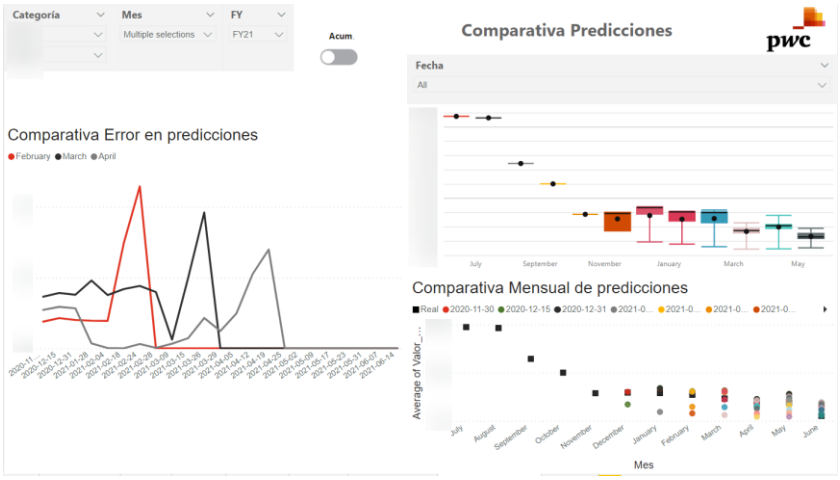


Figura 5.20: Comparativa de las diferentes predicciones) entre si

De esta forma nuestro Power BI queda completo, a día de hoy, y a la espera de sugerencias y posibles extensiones que puedan venir en el futuro.

El conjunto de modelo y tablón de Power BI forman un conjunto que ha gustado mucho de forma interna en la compañía y que considero que forman una herramienta muy completa y que sin duda aportara mucho valor en un futuro.

5.3 Recomendador de Proyectos

Esta es la última fase de nuestro proyecto, como hemos comentado, utilizaremos la información que hemos extraído en el apartado 5.1 que nos permitirá clasificar a los clientes atendiendo a la información transaccional mediante un clustering, para hallar tipologías de comportamiento similares. A partir de esto desarrollamos un algoritmo que recomienda proyectos

a los diferentes clientes basados en su historial de compras y en comportamientos de clientes parecidos a este.

Si, además, añadimos la cronología con la que se consumen los proyectos es posible profundizar en las diferentes tipologías de clientes en "journey" por PwC, de esta forma es posible estudiar la maduración de los clientes a lo largo de su relación con la firma, para así encontrar las propuestas que hayan tenido estadística más éxito a la hora de continuar el "engagement" con los cliente y poder perfilar a los clientes, identificando los más rentable para la firma.

Para lograr estos objetivos teníamos un camino de acción definido, en primer lugar, elaboramos una tabla maestra con la información agregada y consolidada que habíamos extraído anteriormente de la base de datos de la compañía.

De esta forma, la tabla resultado estará formada por un índice único, que lo componen los diferentes clientes de PwC y cuyas columnas son las variables que lo caracterizan en las diferentes dimensiones (monetaria, sectorial, transaccional, temporal…).

Este primer paso puede parecer algo simple al parecer, a priori, un simple pivot y group by de la tabla original agregada que habíamos extraído, que tenía como índice el registro temporal en el que se realizaba cada venta. No obstante, este objetivo no resulto para nada trivial, ya que en la tabla original existían compañías a las que llamábamos matrices, las cuales tenían filiales que habían contratado un proyecto de PwC, por lo que en el registro de nuestra tabla original aparecían los datos de dicho filial, sin embargo, el campo de pago aparecía como cero euros.

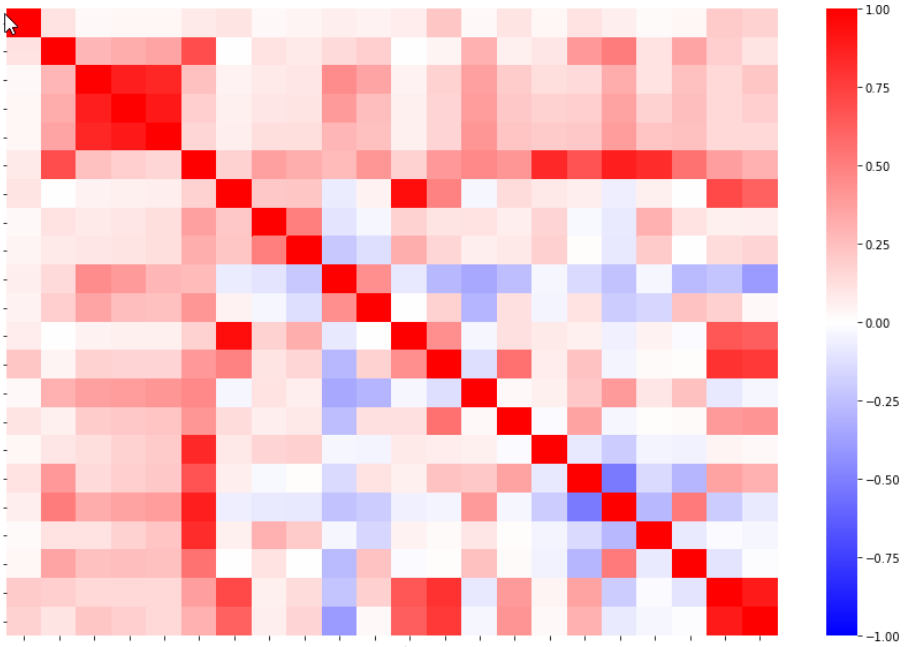
Esto se debía a que eran las filiales las que contrataban los servicios, pero la matriz era la que realizaba el pago en otro proyecto o propuesta. Esto se identificaba porque cada registro o propuesta tenía dos IDs distintos el id de propuesta y el ultimate ID, y, además, tenía dos IDs de cliente, el clientID y el ultimateClientID.

La solución de este problema resulto compleja, ya que no bastaba con eliminar los registros cuyo pago fuese cero, porque nos interesa conocer el número de servicios que contrata cada compañía a la hora de hacer el clustering, no solo el valor de estos. Por este motivo, elaboramos una lógica por la cual se crea un solo campo de ID como combinación de escenarios de los cuatro que hemos comentado, de tal forma que finalmente era posible agrupar por dicho campo y tener una tabla con IDs de clientes únicos. No solo eso, sino que incluso fuimos capaces de enriquecer esta tabla resultado con información extra que nos llegó de forma interna en la compañía.

Recordamos, además, que una vez resolvimos este problema de ID, disponíamos de una información interna que extrae la compañía de el registro mercantil y de otras fuentes publicas que nos permite obtener algunos datos de nuestros clientes que realmente nos proporcionan una visión mucho mas global de estos. Gracias a estos datos extra podremos clasificar a los clientes, no solo por sector de proyecto y por proyectos contratados, ya que esto, podría desembocar en una clasificación meramente superficial en la que solo recomendaríamos proyectos dentro de una misma categoría o sector, lo que nos haría perder mucho valor extra que le podríamos sacar a este modelo. Así, mediante estos datos, podemos conocer datos tan relevantes como el tamaño de la empresa, lugar donde está su sede o el numero de trabajadores, lo que nos permite elaborar un

clustering de estos clientes con una visión de ellos mucho mas global y que nos permitirá desembocar en un recomendador que de verdad encuentre todas las relaciones entre nuestros clientes, siendo capaz de recomendar a cada empresa una amplia variedad de proyectos con los que pueden ser afines y que maximizara la probabilidad de éxito de este apartado.

Una vez realizamos la limpieza y enriquecimiento de nuestra tabla podíamos comenzar a desarrollar nuestro clustering, el cual comenzamos con una ligera exploración de las variables, estudiando, entre otras cosas, las diferente correlaciones entre estas (Gráfica 5.43), lo que nos permite observar el poder discriminatorio y la capacidad generalizadora de las diferente variables.

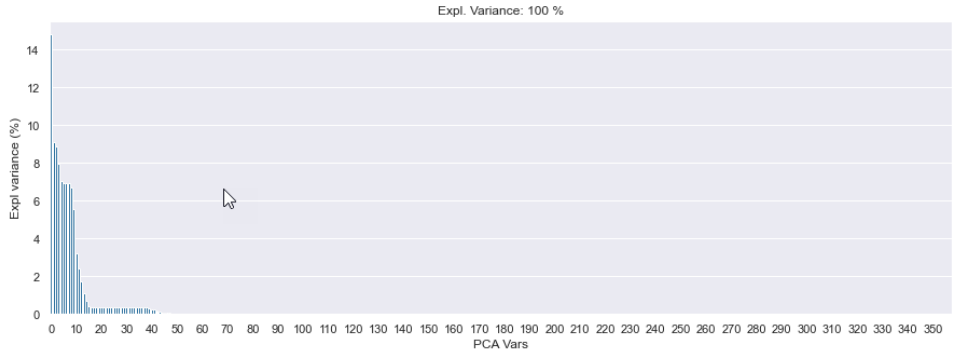


Gráfica 5.43: Correlaciones entre as variables de nuestra tabla maestra

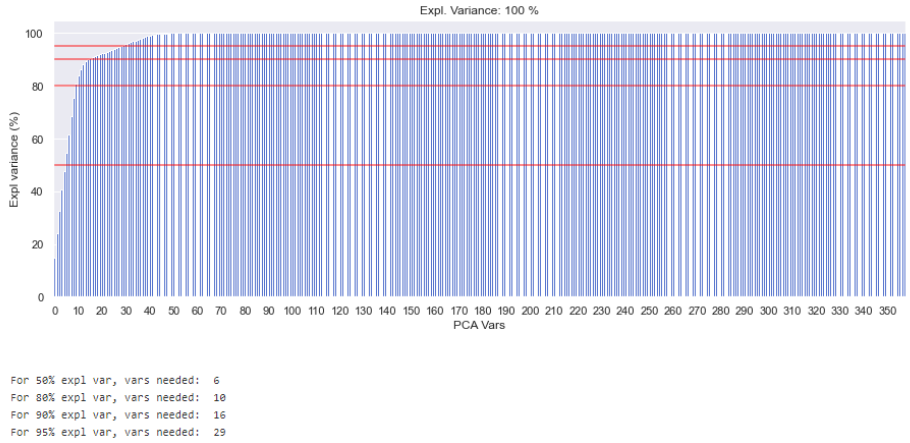
Podemos observar cómo existen algunas variables con una fuerte correlación como las de la esquina superior izquierda que habrá que observar y tratar con cuidado a la hora de desarrollar nuestro clúster.

5.3.1 Clustering

Posteriormente llevamos a cabo un estudio de componentes principales, lo que nos permitirá reducir significativamente la información sin perder demasiada información, ya que nuestro problema actual consta de más de 300 variables, numero demasiado alto y que sin duda podemos disminuir considerablemente mediante las componentes principales.



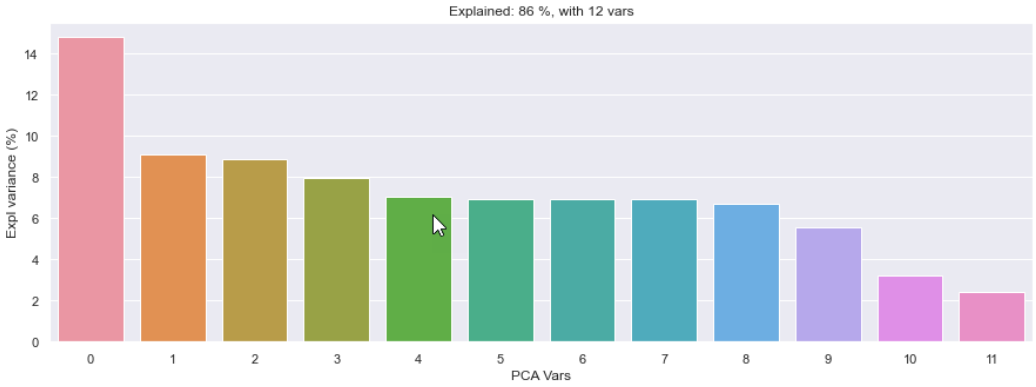
Gráfica 5.44: Explicación que realiza cada PCA de nuestros datos



Gráfica 5.45: Explicación que realiza cada PCA de nuestros datos acumulado

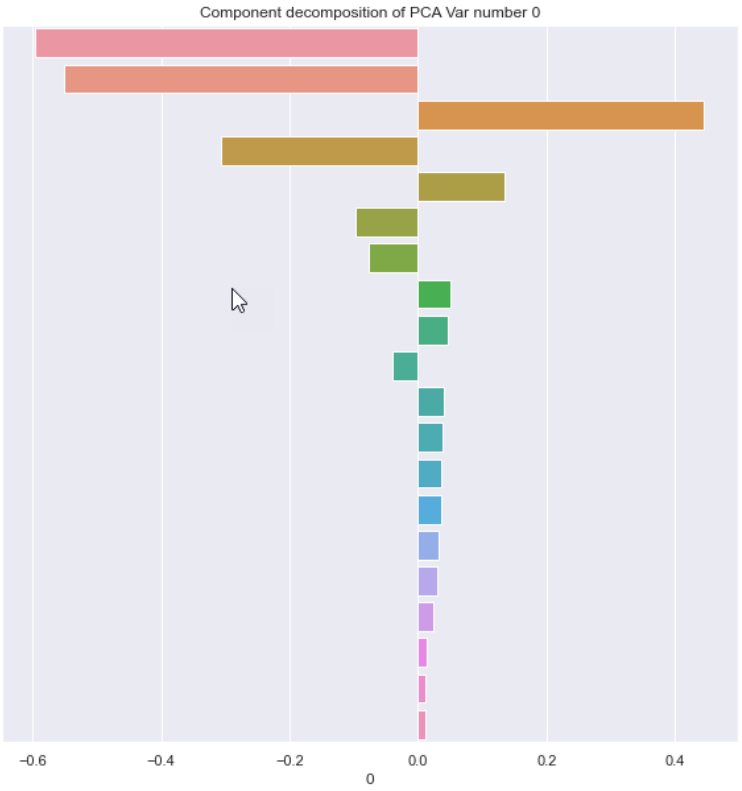
Al ver estos resultados aún resulta más evidente que la reducción de dimensionalidad mediante PCAs era algo necesario, ya que como vemos, es posible explicar el 80% de nuestros datos con tan solo 10 variables, mientras que con 29 explicamos el 95% de estos, frente a las 358 variables que teníamos en el problema inicial, resulta un cambio bastante significativo.

Viendo esto, decidimos seleccionar únicamente 12 variables con las que explicaríamos más del 85% de los datos, un número muy significativo, siendo la



Gráfica 5.46: Explicación de las 12 PCAs seleccionadas

Por último, para dar por finalizado este estudio sería necesario interpretar como se forman cada una de estas PCAs, a partir de cuales de nuestras variables están formadas y que es lo que puede tratar de explicar cada una de ellas, ya que como vemos en la Gráfica 5.47, esta componente principal trata de diferenciar dos grupos principales, aquellos cuya variable tercera es grande con respecto a los que las variables 1, 2 y 4 son altas.



Gráfica 5.47: Composición de la PCA 0 a partir de las variables de nuestra tabla.

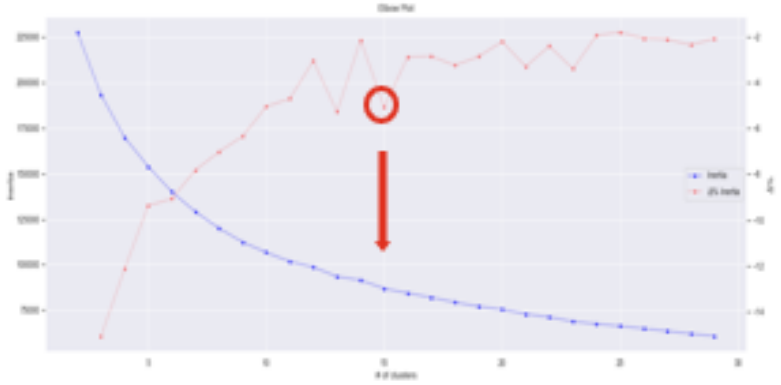
Viendo estos gráficos resulta fácil interpretar que es lo que pretende diferenciar cada una de las PCAs y al tener el porcentaje de los datos que consigue explicar puede resultar realmente útil

para tener una idea general de cómo se organizan tus datos y que características tiene, algo muy valioso a la hora de hacer el clustering y desde un punto de vista de negocio.

Ahora si podemos comenzar con el modelado de nuestro clustering, recordamos que para ello íbamos a usar el método de K-Means, por lo que el primer paso será determinar el número de clústeres que vamos a emplear para nuestro algoritmo. Para ello, utilizamos el método del codo.

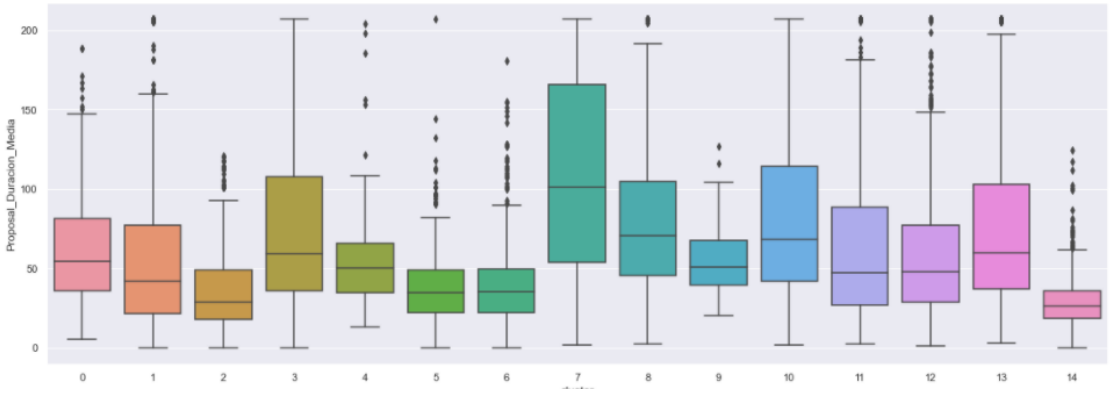
Este método agrupa los datos en un numero de clústeres (k) y calcula una puntuación media para cada valor de k en cierto rango de valores. Por defecto se calcula la inercia, es decir, la distancia de cada punto al centro que ha sido asignado, de esta forma si representamos estos valores en una gráfica, esa grafica se parece a un brazo, por lo que el valor que indica el número de clúster tiene pinta de codo, ya que es el punto en el que al aumentar el número de clúster la distorsión no mejora considerablemente, por lo que no merece la pena seguir aumentando el valor k.

En nuestro caso, se representan dos valores en la Gráfica 5.48, el valor de la inercia en azul al aumentar el número de clústeres, mientras que en rojo tenemos el porcentaje de variación con respecto a la anterior iteración de k. En este caso no está claro en qué punto se produce ese codo que comentábamos, pero si es posible observar como el porcentaje de disminución de inercia tiene un valor alto en el clúster número 15 y a partir de ahí se estabiliza en valores más pequeños (Gráfica 5.48), por este motivo, decidimos que el número de clústeres con el que entrenaríamos nuestro algoritmo de k-Means fuese igual a 15.



Gráfica 5.48: Método del codo para determinar el número de clústeres

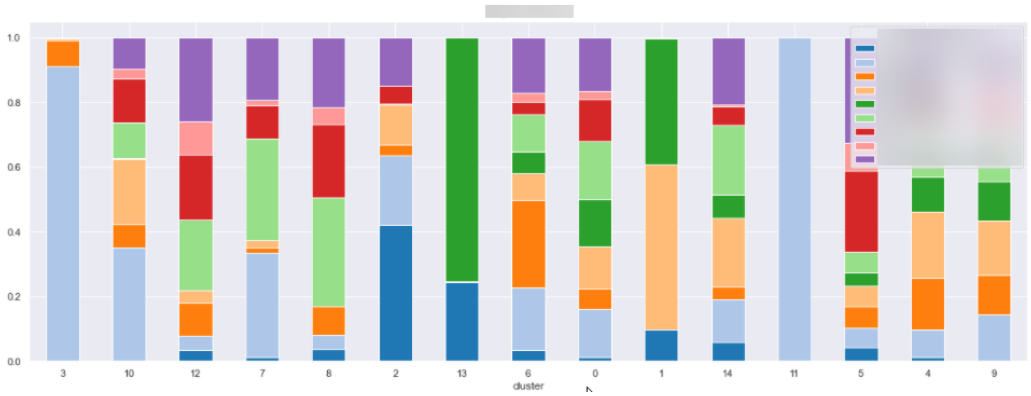
Una vez escogido el número de clústeres solo debemos entrenar nuestro modelo de K-Means, obteniendo unos resultados bastante prometedores y de los que podemos sacar algunas conclusiones con gráficos de nuestros 15 clústeres como estos que vamos a mostrar a continuación:



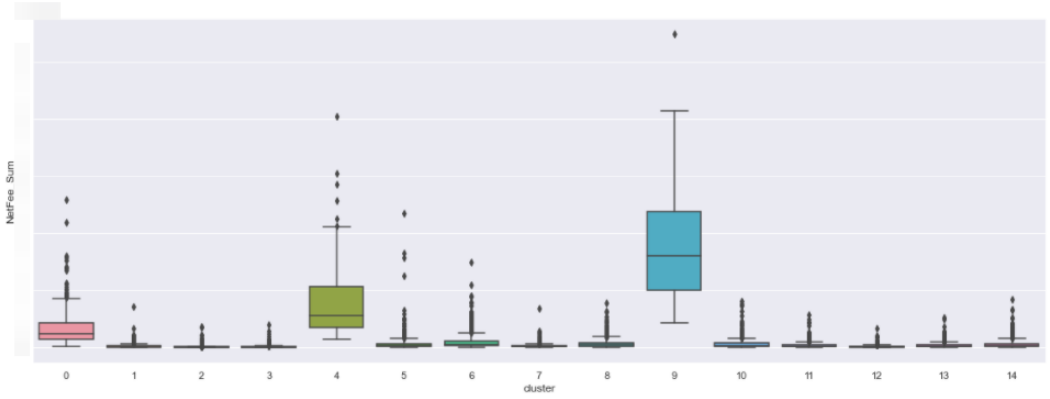
Gráfica 5.49: Boxplot de la duración media de los clústeres obtenidos

En la Gráfica 5.49 podemos observar las distribuciones de las duraciones medias de proyecto de las empresas que conforman cada clúster, de esta forma podemos identificar como en el clúster número 7 se encuentran las empresas con mayor duración de proyecto media asociando esa idea con el clúster número 7.

De igual modo, podemos ver en la Gráfica 5.50, como algunos clúster están conformados por la mayoría de las empresas del mismo sector, como por ejemplo el clúster 11, el 10 o incluso el 13, pudiendo asociar dichos clústeres a ese tipo de empresas.



Gráfica 5.50: Grafico de barras acumulado por sector de PwC



Gráfica 5.51: Boxplot de pago medio de las compañías que forman cada clúster

Por último, en la Gráfica 5.51 observamos como los clústeres 4 y 9 están formadas por empresas que pagan sumas de dinero mucho más cuantiosas en sus proyectos, otro dato muy relevante para nosotros desde un punto de vista de negocio y para nuestro recomendador.

Si nosotros somos capaces de realizar todas estas conclusiones a cerca de los diferentes clústeres que hemos realizado, resulta evidente pensar que un modelo puede extraer una información muy relevante y que será de gran ayuda en nuestro recomendador.

5.3.2 Recomendador

Para realizar el recomendador, recordamos que habíamos decidido usar un recomendador de filtro colaborativo, ya que no necesita información de los productos para funcionar, sino que se basa en los propios datos para averiguar qué objetos recomendar o a que personas les va a gustar el que.

En nuestro caso, tenemos un tipo de calificación muy característica, que recordamos que se conoce como implícita, ya que no disponemos de un número que indica que productos han gustado o no a cada cliente, simplemente disponemos de 1 si el usuario ha consumido nuestro producto y 0 si no.

Este tipo de datos supone un desafío extra, ya que a diferencia de en los algoritmos con data explicita donde nuestro modelo únicamente tiene que rellenar los huecos que faltan con el rating o puntuación esperados, en nuestro caso, los huecos son 0, es decir, que el usuario no ha consumido ese proyecto y que, por esto, puede obtener información de los huecos a diferencia de con los datos explícitos.

Recordamos que este tipo de recomendadores se partía de una matriz donde disponemos de la información de los clientes y los productos como filas y columnas respectivamente (matriz R), siendo el objetivo de nuestro algoritmo factorizar esa matriz enorme para extraer los “gustos” de cada usuario. Para ello, empleamos el métodos de mínimos cuadrados alternativos (ALS), que lo que intenta es en cada iteración acercarse más y más a esa representación factorizada de nuestros datos.

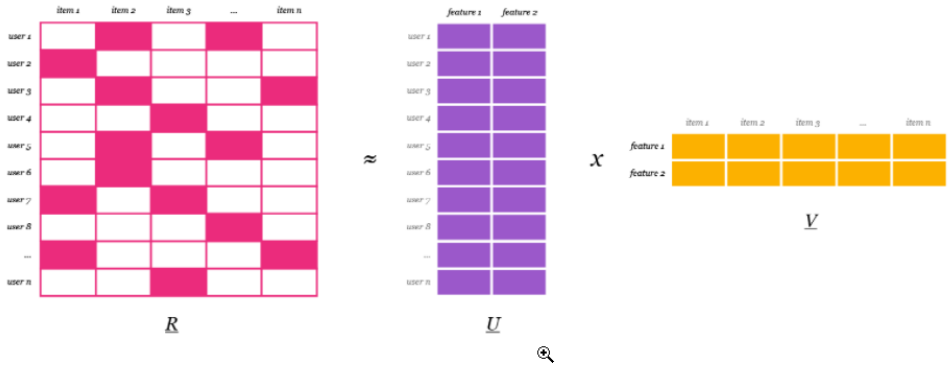


Figura 5.21: Representación de la matriz R mediante U y V [12]

De esta forma, buscamos representar R como una multiplicación de dos matrices U y V. Para ello, ALS asigna valores aleatorios a esas dos matrices y utilizando mínimos cuadrados de forma iterativa podemos lograr una aproximación lo mas precisa posible de R, este método trata de ajustar una línea, en su forma mas básica, a los datos, tratando en cada iteración que la distancia entre los datos y la recta sea lo mas pequeña posible, tratando de conseguir el ajuste óptimo.

Al utilizar ALS seguimos esta misma idea solo que tratamos de optimizar la matriz U, mientras que V se mantiene fija y alternamos entre una y la otra hasta llegar a la solución.

De esta forma construimos un modelo a partir de ALS con el que éramos capaces de generar una lista de proyectos con la probabilidad de que cada empresa fuese propensa a querer contratar dichos servicios. De este modelo únicamente elaboramos un primer producto mínimo viable con el que éramos capaces de extraer dicha lista que era el objetivo final, sin embargo, los socios y dirigentes del proyecto mostraron más interés en la predicción de ventas y este modelo quedo un poco mas estancado en esta primera iteración, la cual no daba malos resultados, simplemente necesitaba mas tiempo de ajuste para ser una herramienta que se pudiese usar de forma automática en un negocio.

6. Conclusiones y Trabajos Futuros

6.1 Resumen y Conclusiones del Proyecto

Este proyecto en el que PwC me ha dado la posibilidad de formar parte se trata de un proyecto que trata muchos de los temas que hemos tratado a lo largo del curso, que me ha servido, no solo para reforzar esos conocimientos, sino para completar mi formación el en mundo de data y analítica.

Dicho esto, en este proyecto hemos desarrollado desde cero dos modelos completamente independientes que pretenden dar una mejor visión a la empresa sobre los cliente y el mercado con los que interactúan para poder así obtener ciertas ventajas competitivas en un futuro aumentando las ventas y mejorando la experiencia de nuestros clientes.

Para ello, hemos utilizado los datos de ventas que posee y actualiza la firma de forma interna, tratándolos y adaptándolos a las necesidades que teníamos por cada uno de los modelos que queríamos desarrollar utilizando para ello la herramienta Alteryx.

Posteriormente hemos realizado un total de doce modelos independiente de series temporales para predecir no solo las categorías de la empresa sino los sectores de esta, logrando unas muy buenas predicciones de cierre y del año próximo, con su posterior visualización en Power BI para poder extraer conclusiones de forma rápida y sencilla.

Por último, hemos clusterizado los diferentes clientes de la compañía pudiendo así agruparlos por perfiles similares y realizar una recomendación de proyectos a esos clientes para tratar de alargar su compromiso y fidelidad con la firma, y aunque, este modelo no ha sido desarrollado hasta su ponto optimo como si que lo esta el resto del proyecto, tenemos un gran porcentaje de este hecho pudiendo incluso realizar recomendaciones a las compañías con una lista ordenada por probabilidad de interés.

En resumen, se podría decir que este proyecto ha resultado ser un gran desafío, no solo a nivel personal sino como grupo en los que han ido surgiendo múltiples desafíos y situaciones que parecían un laberinto en si mismas: A pesar de todo, hemos sabido sobreponernos a todas estas situaciones hasta lograr el producto que tenemos ahora, que aunque no esté acabado del todo, sin duda es motivo para estar orgulloso por el trabajo que he realizado junto a mis compañeros en los últimos meses.

En cuanto a los modelos, considero que la serie temporal sin duda será un modelo de amplio interés en la firma como se ha demostrado por los dirigente a lo largo de estos meses, con unos resultados mas que satisfactorios y unas visualizaciones finales mediante Power BI que sin duda dejan obsoleto al anterior método de realizar el presupuesto de forma interna a mano preguntando a todos los sectores y categorías por sus ingresos esperados y que llevaba varias semanas elaborarlo, dado que de esta forma es posible lograr unos resultados parecidos o incluso mejores en apenas una hora y mediante un clic de ratón.

Por su parte, el recomendador no se parece a nada que hubiese anteriormente en la compañía y de ahí venia la ilusión de todos con este modelo en concreto, sin embargo, el interés mostrado por este modelo no fue el esperado y se quedó un poco estancado. Aun así, sus resultados son bastante buenos para el tiempo que se le ha dedicado y sin duda, con un poco más de trabajo podríamos lograr que este modelo fuese de gran ayuda en la compañía, no solo a nivel ventas, sino relación con el cliente y experiencia con nuestros servicios.

Por todo esto, considero que este proyecto me ha servido para completar mi formación, no solo siendo capaz participar de forma activa en un proyecto de Big Data y Analítica real, sino por ser capaz de reforzar alguno de los conocimientos vistos en clase, sino por ser capaz de aprender muchos nuevos, todo ello mediante el trabajo en grupo y la colaboración que supone realizar un proyecto de este calibre.

6.2 Trabajos Futuros

Aunque hemos conseguido desarrollar productos bastante buenos en todos los ámbitos de este proyecto, aun nos quedan algunas cosas por hacer y dadas las magnitudes de este proyecto también se entiende que no ha sido posible realizarlo todo en tan solo 6 meses, por ello aun quedan detalles por pulir antes de que podamos dar este proyecto por finalizado.

En primer lugar, en cuanto a la serie temporal es importante terminar su puesta en producción lo antes posible para que de esta forma sea posible actualizar los datos, el modelo y las visualizaciones de forma automática cada semana sin que una persona deba estar ejecutando el Alteryx, después el modelo y por último pasando los nuevos datos al Power BI.

Sobre la serie temporal también se ha hablado de modificar el periodo de tiempos, ya que el hecho de que los datos sean semanales supone algunos problemas para los datos de los directivos y expertos, puesto que muchos meses difieren mucho en sus ventas de un año para otro debido a pe por la configuración d ellos dias igual de un año para otro tienen una semana mas o menos que el año anterior, lo que supone una gran diferencia de ventas. Por esto, seria interesante tratar de realizar el modelos de forma diaria, continuando con los entrenamientos a nivel semanal, pero teniendo los datos por día en lugar de por semana, esto solucionaría el problema de las agrupaciones por meses y de cualquier otro tipo de problemas que la agrupación por semanas pueda acarrear.

Por último, en la serie temporal es prioritario garantizar el cambio de año fiscal que se produce en apenas diez dias, para que todas las herramientas desarrolladas sigan funcionando una vez finalice de forma automática y con la misma eficacia.

En cuanto al recomendador, el trabajo que queda por hacer es mas complejo y costoso, ya que seria necesario optimizar aun mas el proyecto incluyendo algunos conceptos básicos como las reglas de asociación o incluso una prueba d ellos parámetros que hemos seleccionado en el método de mínimos cuadrados aleatorios.

En cuanto al clustering, seria necesario añadir la nueva información que nos pasaron de forma interna de la compañía, ya que cuando se realizó esta primera iteración del modelo solo el 40% de los clientes cruzaba con la información que habíamos añadido, en la actualidad hemos logrado subir esa cifra hasta el 77% de cruce lo que enriquecería de forma muy valiosa nuestros datos y seria necesario una nueva ejecución dl modelo de clustering. Incluso seria posible estudiar otros métodos para la construcción del clustering que no se probaron en su momento, como por ejemplo la mistura de gaussianas que podría ser capaz de mejorar incluso los buenos resultados obtenidos.

Por último, faltaría construir una librería de todo este código de la misma forma que lo hicimos con la serie temporal, de esta forma seria posible ejecutar este algoritmo de forma automática, aunque no resulta tan prioritario como lo es en la serie temporal que requiere una ejecución cada semana.

7. Presupuesto

En este capítulo llevaremos a cabo un desglose global de todos los costes que ha supuesto este proyecto, para mantener un orden y tener una mejor comprensión de los costes estos se han dividido en tres subgrupos: materiales, licencias y personales.

7.1 Materiales

Vamos a realizar una recapitulación de todos los costes materiales que ha supuesto este proyecto, comenzando por nombrar todos los recursos empleados en este.

- Ordenador portátil de gama alta HP EliteBook 840 G4 con procesador i5 de septima generación y 8GB de memoria RAM.
- Espacio de trabajo acondicionado de luz, calefacción y aire acondicionado para poder llevar a cabo las tareas de búsqueda y desarrollo del proyecto durante los 9 meses de duración de este, con un precio medio mensual de unos 100 euros para este proyecto.
- Material de oficina como los empleados para la escritura, cuadernos y tinta de impresora.

La fórmula 5.1 expresa como se han obtenido los costes de cada uno de estos materiales, donde A representa el tiempo de uso de dicho material, T es el tiempo de devaluación del producto vida útil, C es el coste de compra de dicho producto y U es el uso del producto medio en porcentaje.

$$C_{am} = \frac{A}{T} CU \tag{7.1}$$

Para la siguiente tabla se considera que el espacio de trabajo y los cursos son recursos dedicados a este proyecto y el precio que aparece de estos es acorde a esta suposición, por ello son amortizados al 100%.

Material	A[mese]	T[meses]	C[€]	U[%]	C_{am}[€]
Ordenador	6	24	450	90	101.25
Espacio de trabajo	6		100		600
Material de oficina	6	12	40	100	20
				TOTAL	721.25

Tabla 7.1: Coste materiales

7.2 Licencias

En este caso, las licencias que hemos empleados son las de Alteryx Designer, cuya licencia asciende a unos 4400 euros al año, por lo que el coste total de las licencias seria de 2200 euros, ya que se ha empleado durante 6 meses únicamente.

7.3 Personal

En este apartado vamos a considerar el coste personal que ha supuesto para mí, para otro becario y para mi tutor el desarrollo de este proyecto, a pesar de que el volumen de personal en este proyecto ha variado durante los seis meses, podemos considerar eso como la media.

Por esto, vamos a suponer que a lo largo de este proyecto de seis meses he dedicado unas 5 horas diarias, trabajando al mes 22 días laborables, y las horas de mi tutor serán una aproximación de alrededor de las mismas horas que yo.

Persona	Horas de trabajo	Coste[€/hora]	Coste total [€]
Trainee (2)	660	5.5	7260
Senior Ass.	660	18	11880
		TOTAL	19140

Tabla 7.2: Coste de personal

De esta forma el coste total del proyecto sería:

	COSTE [€]
Materiales	721.25
Licencias	2200
Personal	19140
TOTAL	22061.25

Tabla 7.3: Coste total del proyecto

Dando lugar a una cantidad de presupuesto añadida de unos 22061.25€

8. Bibliografía

[1] A. Company, «Alteryx Documentation,» [En línea]. Available: https://help.alteryx.com/es-419/current/designer/get-started-designer.

[2] M. Mirzavand y R. Ghazavi, «A Stochastic Modelling Technique for Groundwater Level Forecasting in an Arid Environment Using Time Series Methods,» Water Resources Management, November 2014.

[3] R. J. H. a. G. Athanasopoulos, Forecasting: principles and practice, 2nd edition, Melbourne, Australia: OTexts, 2018.

[4] Facebook, «Prophet - Seasonality, Holiday Effects, And Regressors,» Facebook, [En línea]. Available: https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#fourier-order-for-seasonalities.

[5] S. Priy, «Clustering in Machine Learning,» GeeksforGeeks, 23 febrero 2020. [En línea]. Available: https://www.geeksforgeeks.org/clustering-in-machine-learning/.

[6] G. Seif, «Towards data science,» 5 febrero 2018. [En línea]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

[7] S. Learn, «Clustering,» [En línea]. Available: https://scikit-learn.org/stable/modules/clustering.html.

[8] B. Shetty, «AN IN-DEPTH GUIDE TO HOW RECOMMENDER SYSTEMS WORK,» Bult in, 24 junio 2019. [En línea]. Available: https://builtin.com/data-science/recommender-systems.

[9] P. Kordík, «Machine Learning for Recommender systems — Part 1 (algorithms, evaluation and cold start),» Medium, 3 junio 2018. [En línea]. Available: https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed.

[10] «Association Rules and the Apriori Algorithm: A Tutorial,» [En línea]. Available: https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html.

[11] Facebook, «Prophet Diagnostics,» Facebook, [En línea]. Available: https://facebook.github.io/prophet/docs/diagnostics.html#hyperparameter-tuning.

[12] Victor, «ALS Implicit Collaborative Filtering,» Medium, 2017 august 23. [En línea]. Available: https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe.

[13] A. Bronstein, «A Quick Introduction to the "Pandas" Python Library,» 18 April 2017. [En línea]. Available: https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673.

[14] Alteryx Company, «About Us,» [En línea]. Available: https://www.alteryx.com/company/about-us.

[15] The Information Lab, [En línea]. Available: https://www.theinformationlab.es/que-es-alteryx-y-para-que-sirve/.

[16] J. Torres, 22 Septiembre 2019. [En línea]. Available: https://torres.ai/redes-neuronales-recurrentes/.

[17] J. Brownlee, «Machine Learning Mastery,» 15 Agosto 2020. [En línea]. Available: https://machinelearningmastery.com/time-series-forecasting/.

[18] D. Burba, «Towards data science,» 3 octubre 2019. [En línea]. Available: https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb.

[19] J. Browlee, «Machine Learning Mastery,» 17 agosto 2018. [En línea]. Available: https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/.

[20] S. Kaushik, «An Introduction to Clustering and different methods of clustering,» Analytics Vidhya, 3 noviembre 2016. [En línea]. Available: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/.

[21] J. Wu, «Content-Based Recommender Systems and Association Rules,» Medium, 27 mayo 2019. [En línea]. Available: https://medium.com/@jwu2/content-based-recommender-systems-and-association-rules-599843cb2fd9.

9. Anexo A

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

I.C.A.I.

PROYECTOS FIN DE MÁSTER
CURSO:

Ficha de proyecto fin de máster

Titulación y optatividad: Ingeniería de Tecnologías Industriales (Electrónica)

Alumno 1º Apellido: Gurtubay

2º Apellido: Regulez

Nombre: Ander

Teléfono de contacto: 670287798

e-mail: 202016918@alu.comillas.edu

Título del Proyecto Fin de Máster: Proyecto de optimización de los servicios y ventas mediante Machine Learning

Director (nombre y dos apellidos): CARLOS BLANCO JABARES

Teléfono de contacto: e-mail: 688510966 carlos.blanco.jabares@pwc.com

Breve descripción del proyecto (5 o 6 líneas): El objetivo del proyecto consiste en predecir el comportamiento del mercado objeto de estudio, así como los potenciales clientes que se integran en el y las necesidades de estos potenciales cliente. Para ello, construiremos una serie de modelos de machine learning apoyados en fuentes de información internas, sobre las que se realizara una extracción y preprocesador de los datos, con los que luego desarrollaremos unos

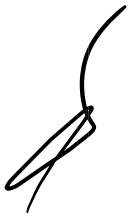
Aceptación del Director (firma y fecha)

17/02/21

modelos de clustering de clientes, un recomendador de potenciales proyectos y una serie temporal con los que pretenderemos cumplir los objetivos iniciales de predicción de clientes y del mercado objeto de estudio.

El documento final del proyecto será subido al Repositorio Institucional de Comillas con acceso público. El alumno podrá solicitar un nivel restringido de acceso (incluido el "cerrado" o "confidencial") que podrá concederse, excepcionalmente, si está plenamente justificado.

The final report of the Project will be uploaded to the Comillas Institutional Repository with public access. The student will be able to ask for a restricted access (even "closed" or "confidential") which will be exceptionally accepted if it is fully justified.



Reverso del Anexo A

DATOS RELATIVOS AL PROYECTO FIN DE GRADO

Título del Proyecto Fin de Grado: Blind Aids: La app inteligente de ayuda para personas con deficiencia visual

Director/es del Proyecto Fin de Grado: Fernando Alonso Martín

Curso Académico en el que se realizó: 2019-2020

Universidad (indicarla si no es Comillas): Universidad Carlos III