Proyecto de Estancia de Investigación

Modelos para la Construcción de Portafolios de Productos de Telefonía Celular

Ana Luisa Masetto Herrera - 000183203

Contents

1 Introducción

La industría de las telecomunicaciones ha crecido drásticamente en los últimos años dado a los diversos avances tecnológicos que se han alcanzado; dentro de esta industría se encuentra el sector enfocado a la telefonía celular, sector que también ha presenciado un aumento considerable en su demanda de productos. De acuerdo con información recopilada en conjunto por el Instituto Nacional de Estadística y Geografía (INEGI), la Secretaría de Comunicaciones y Transportes (SCT) y el Instituto Federal de Telecomunicaciones (IFT), el uso de la telefonía celular ha ganado lugar como una de las tecnologías con mayor penetración en la población mexicana, estimando que el año pasado había un total de 69.6 millones de personas que tenían un teléfono inteligente, indicando un incremento de usuarios del 7.57% en comparación con el 2017 (INEGI, 2018).

En la actualidad se sabe que prácticamente todas las personas poseen un celular, no solamente para facilitar la comunicación entre amigos, familiares, compañeros de trabajo, clientes, etc; sino que también se ha convertido en una herramimenta que facilita algunas de las actividades cotidianas de las personas, como: buscar direcciones, pedir comida, solicitar información bancaria, realizar documentos para tareas o trabajos, o simplemente funciona como fuente de entretenimiento gracias a su capacidad para conectarse a redes sociales y para almacenar juegos, videos, fotos y música.

Si bien dijo Oswaldo Contreras Saldívar, presidente del Instituto Federal de Telecomunicaciones (IFT), "No sólo se usa el dispositivo móvil por lo práctico, sino porque lo queremos usar para todo, todo el tiempo: lo queremos al alcance de la mano", sin embargo, es muy importante resaltar que aunque la necesidad creciente de tener un dispositivo movil es de la mayoria de las personas, no todas buscan las mismas caracteristicas en los celulares. Hay personas que se fijan únicamente en el rango de precios (gamma del producto), en la marca, en la apariencia, pero también hay personas que se guían más por la construcción en si del modelo, como son la capacidad de memoria, la vida útil del producto, la definición de la cámara, el software que utiliza, etc.

Es por eso que las compañías enfocadas a la venta de productos de telefonía celular enfrentan el reto de **pronosticar el número unidades a vender de cada producto**, de no hacerse propiamente esto podría generar problemas relacionados con la pérdida de clientes dado la falta de productos en los puntos de venta, o problemas como gastos adicionales en trasporte o almacenamiento. Esto se debe a que las compañias de este sector cuentan con diversos puntos de venta, en distintas ciudades, estados y zonas, y es muy probable que el mercado al que se dirigue cada uno de estos puntos no sea el mismo.

Partiendo de esta situación, el proyecto descrito en este documento busca generar un proyecto de Ciencia de Datos utilizando información real de una empresa de telecomunicaciones en México y cuyo resultado final será la propuesta de modelos para la construcción de portafolios de telefonia celular. El proyecto fue realizado por Ana Luisa Masetto Herrera, alumna de la maestría de Ciencia de Datos en el Instituto Tecnológico Autónomo de México (ITAM); este se llevó acabo en las instalaciones de la empresa IBM México en Santa Fe y fue supervisado por el ingeniero Rubén Pineda Piña. Cabe mencionar que los datos utilizados para este proyecto son de un cliente de IBM, por lo tanto, y dado los contratos de privacidad que se tiene con los clientes de IBM, el nombre real la empresa a la que corresponden los datos no se va a mencionar explícitamente, es por eso que a partir de este momento se le referirá a dicha empresa como **la empresa ABCD**, con el fin de que esta permanezca en anonimato.

2 Descripción del Proyecto

El objetivo general del proyecto es desarrollar un proyecto de ciencia de datos con el que se pueda mejorar la construcción de portafolios de productos de telefonía celular en diversos puntos de venta de la empresa ABCD. Aunado a este objetivo, se tienen diversos objetivos específicos:

1. Familiarice con la problemática a tratar, la empresa y las herramientas que IBM proporciona.
2. Analizar los datos crudos proporcionados por la empresa ABCD para detectar los problemas de calidad que estos presentan con el fin de llevar a cabo una limpieza de datos.
3. Realizar un análisis exploratorio de los datos para obtener información relevante que permita entender de mejor manera la situación actual de la empresa.
4. Proponer modelos de regresión que permitan cumplir con el objetivo principal del proyecto.
5. Realizar las transformaciones necesarias de los datos para facilitar su manejo.
6. Implementar ingeniería de características que permita enriquecer los modelos propuestos.
7. Desarrollar diferentes modelos para comparar su comportamiento.
8. Seleccionar el modelo con el mejor desempeño.
9. Realizar la documentación necesaria para entregar al supervisor del proyecto.
10. Realizar la documentación necesaria para entregar a la maestría de Ciencia de Datos en el ITAM.

3 Herramientas

Antes de indagar más en la problemática a tratar en este proyecto, es importante mencionar las herramientas que se utilizaron para cumplir con cada uno de los objetivos del proyecto.

Lo primero que hay que mencionar es que todos los código se ejecutaron directamente de la plataforma de IBM utilizando **Watson Studio**; una plataforma en la nube que facilita el manejo de grandes volúmenes de datos y que además cuenta con diversas herramientas tales como: **RStudio** y **Jupyter Notebook**.

Una vez que ya se mencionó esa parte, es importante mencionar que para la parte de *limpieza*, *transformación*, *análisis exploratorio*, e *ingeniería de características*, se utilizó **RStudio** como entorno de desarrollo.

Como siguiente punto, se utilizó **Jupyter Notebook** para desarrollar los códigos para los modelos de aprendizaje de máquina.

Finalmente, todos los códigos creados se subieron a la siguiente página de github [1] donde pueden consultarse para futuras referencias.

[1] https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion

4 Descripción de las Bases de Datos

Para este proyecto se tienen datos reales de la empresa **ABCD**, compañía de la industría de las telecomunicaciones enfocada en el sector de telefonía celular. La empresa proporcionó dos bases de datos: la primera corresponde a un catálogo de tiendas y la segunda a un registro de ventas; ambas bases de datos se describen con más detalle a continuación.

4.1 Catálogo de Puntos de Venta

El catálogo fue proporcionado en un archivo con extensión **csv** y tiene un total de 1,911 renglones y 79 columnas. Las 79 variables poseen diferentes propiedades relacionadas con los puntos de venta, sin embargo, de estas sólo se tomarán en cuenta las siguientes:

- **Variables a considerar del catálogo de tiendas:**
  - **Nombre del pdv**: Nombre del punto de venta.
  - **Nuevo nombre del pdv**: Nuevo nombre del punto de venta (no todos los puntos de venta fueron renombrados).
  - **Regiones homologadas**: División por región a la que pertenece cada punto de venta (norte, sur, etc.).
  - **Estado**: Estado donde se encuentra el punto de venta.
  - **Ciudad**: Ciudad donde se encuentra el punto de venta.
  - **Latitud**: Ubicación con coordenadas geográficas del punto de venta.
  - **Longitud**: Ubicación con coordenadas geográficas del punto de venta.

Las razones por las cuales no se consideran las demás variables son: en primer lugar, no es claro a que se refieren algunas variables y no se proporcionó un diccionario con la descripción de ellas, en segundo lugar es porque existen muchos valores faltantes y la información para completarlos no es posible de obtener por cuenta propia y la empresa no accedió a proporcionar más información, por último, hay variables con información estimada (como población en el 2020) de la cuál no se sabe con certeza las unidades ni la forma en la que se calcularon.

4.2 Registro de Ventas

El segundo documento que se proporcionó fue el que contiene los registros de ventas de la empresa, registros que tienen lugar a partir del 1 de junio del 2018 al 31 de marzo del 2019. El archivo con extensión **csv** con dicha información tiene 1,048,575 renglones y 10 columnas.

Las 10 variables que se tienen son las siguientes:

- **Variables del registro de ventas:**
  - **Punto de Venta**: Nombre del punto de venta donde se realizó la compra.
  - **Plan tarifario**: Plan tarifario bajo el cual se vendió la unidad.
  - **Sku**: Código único del producto. Cabe mencionar que son códigos internos de la compañía, por lo tanto, no se sabe el nombre comercial de los productos.
  - **Fecha**: Fecha en la que se registró la venta de la unidad.
  - **Precio**: Precio de la unidad.
  - **Costo**: Costo de la unidad. Estos valores son muy cercanos a los de la variable anterior.
  - **Marca**: Marca de la unidad vendida.
  - **Ventas**: Columna con valores iguales a 1. Es decir, cada registro dentro del documento (cada renglón) corresponde a una venta.
  - **Mth**: Mes en el que se hizo la venta de la unidad.
  - **Yr**: Año en la que se hizo la venta de la unidad.

5 Limpieza y Transformación de los Datos

En esta sección del documento se describe cuáles fueron los problemas de calidad que se detectaron en ambos archivos (catálogo y registro) y cuál fue el proceso de limpieza y transformación que se llevó a cabo para llegar a una base de datos limpia que permita facilitar su manejo e interpretación. Este procesamiento de los datos se llevo a cabo en un script en RStudio que puede consultarse en esta liga [2].

5.1 Problemas de calidad: Catálogo de tiendas

- **Máyusculas y minúsculas**: Había celdas con información en mayúsculas y otras en mínusculas.
- **Caracteres especiales**: Algunos de los registros tenían acentos, guiones y dobles espacios.
- **Valores faltantes**:
  - La columna **Nuevo nombre del pdv** poseía muchos valores faltantes, al igual que las columnas de **longitud** y **latitud**.

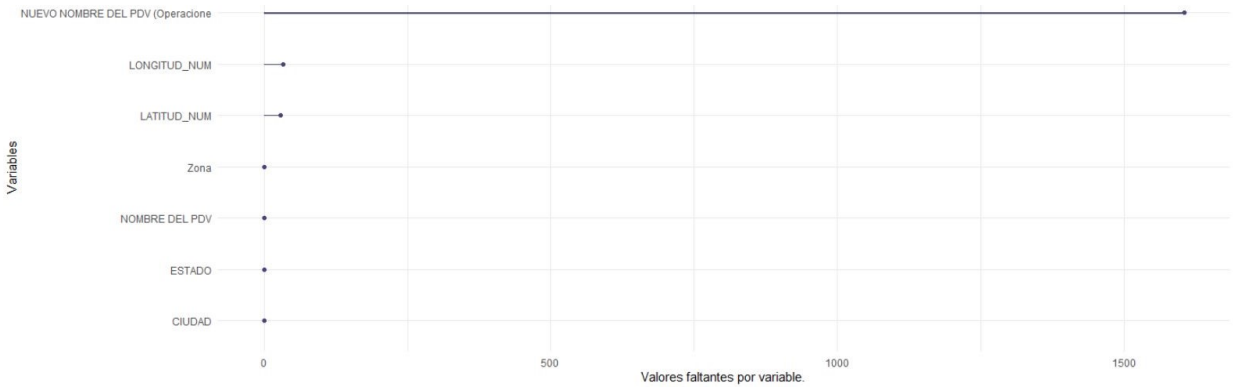


Figure 1: Valores faltantes dentro del cátalogo de tiendas proporcionado por la empresa ABCD.

- **Tiendas faltantes en el catálogo**: Para detectar este problema se intento hacer un **join** del registro con el catálogo y se detectaron algunas tiendas en el registro que no estaban en el catálogo.
- **Tiendas repetidas en el catálogo**: Habían tiendas registradas más de una vez en el catálogo.
- **Registros erroneos**:
  - La columna de estados tenía más de 32 estados registrados y como la información es únicamente de la república mexicana, se sabe que eso no es posible.
  - La columna de las regiones tenía registros que no hacían sentido.
  - Por un lado, la columna de longitud tenía coordenadas registradas **positivas** y eso no es posible dado que los estados de la república mexicana únicamente abarcan coordenadas de longitud entre -86 y -116 aproximadamente. Por el otro, la columna de latitud tenía coordanas registradas fuera de rango, por ejemplo: había un registro cuya latitud era de 20 millones, valores que no es posible.

```
NOMBRE DEL PDV     NUEVO NOMBRE DEL PDV (Operacione    ESTADO           CIUDAD          LATITUD_NUM       LONGITUD_NUM        Zona
Length:2801        Length:2801                     Length:2801       Length:2801       Min.   :       15   Min.   :-117.12   Length:2801
Class :character   Class :character                Class :character  Class :character  1st Qu.:       19   1st Qu.:-102.06   Class :character
Mode  :character   Mode  :character                Mode  :character  Mode  :character  Median :       20   Median : -99.24   Mode  :character
                                                                                       Mean   :    10404   Mean   : -99.44
                                                                                       3rd Qu.:       22   3rd Qu.: -98.75
                                                                                       Max.   :29081531   Max.   : 115.36
```

Figure 2: Resumen donde se notan los valores fuera de rango de las variables de latitud y longitud.

[2] https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion/tree/master/Limpieza_De_Datos

5.2 Problemas de calidad: Registro de ventas

- **Mayúsculas y minúsculas**: No se tenía un formato definido en tanto a la forma en la que se debía de hacer los registros. Había celdas con información en mayúsculas y otras en mínusculas.
- **Valores faltantes**:
  - La columna de **precio** tenía más de 7,000 valores faltantes.
  - La columna de **marca** tenía 9 valores faltantes.
  - La columna de **costo** tenía 7 valores faltantes.

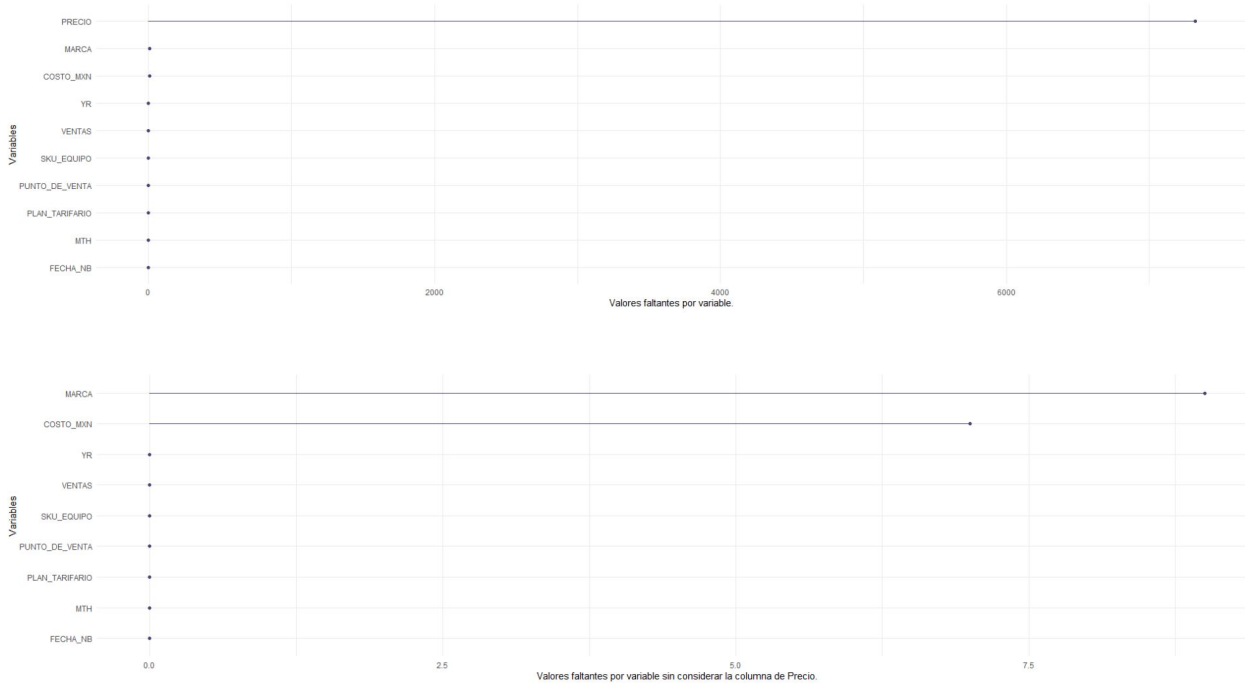


Figure 3: Valores faltantes del Registro de Ventas.

- **Formato no homogeneo en la columna de fecha**: Los registros de la columna de fecha tienen diferentes formatos (01-03-18, 01MAR18, 01-03-2018, entre otros).
- **Caracteres especiales**: Algunos de los registros tenían acentos, guiones y dobles espacios.
- **Errores de registro**: Hay marcas escritas de diversas maneras.
- **Tiendas mal escritas**: Hay tiendas que están registradas con un nombre erróneo.

5.3 Limpieza de Datos: Catálogo de tiendas

Una vez que se detectaron los problemas de calidad en el catálogo se procede con su limpieza. A continuación se presentan las actividades principales que se llevaron a cabo.

- **Homogeneizar todos los registros a minúsculas**: Todos los registros se convierten a minúsculas para facilitar su manejo.
- **Eliminar caracteres especiales**: Todos los acentos se remueven, al igual que la letra **ñ**, los dobles espacios y otros tipos de espacios (Non Breaking Spaces).
- **Imputar valores faltantes**:

  - Los puntos de venta que cambiaron de nombre fueron sustituidos por su nuevo nombre de tal manera que en lugar de trabajar con dos columnas únicamente se quedó una llamada **punto_de_venta_**.
  - Los valores faltantes en las columnas de **longitud** y **latitud** fueron imputados. Tras un proceso sumamente artesanal de buscar cada una de las 39 tiendas con valores faltantes en sus coordenas geograficas en google maps se logró recopilar la información faltante.

- **Correción de registros erroneos**:

  - Se homogenizan los registros de tal manera que solo se tenga información de los 32 estados de la repúbica mexicana. Por ejemplo, en la columna de estado estaba escrito **matamoros** y este se sustituyó por **tamaulipas**, o el **estado de méxico** estaba escrito de diferentes formas y este se homogeneizó.
  - Los registros positivos dentro de la columna de **longitud** se cambiaron a valores negativos y se verificaron de tal manera que estás fueran correctas y efectivamente se refirieran a algún punto de venta de la compañía ABCD.
  - Los registros fuera de rango de la columna de **latitud** se corrigieron a valores dentro de rango.
  - Se buscó información sobre las zonas en las que se divide el territorio mexicano y se encontró en la página de CONABIO [3] que la república se puede dividir en 8 regiones, por lo tanto, se actualizan las regiones originales de la base de datos y se sustituyen por la nueva división.

- **Completar tiendas faltantes en el catálogo**: Se hizo un join con la base de datos de registro de ventas y se detectaron las tiendas que no hicieron match, luego estas se intentaron buscar dentro del catálogo con con otros nombres para ver si no era por un mal registro y las que no se encontraban se buscaron manualmente en google maps para luego agregarlas al catálogo.

- **Detectar tiendas repetidas en el catálogo**: Habían 26 tiendas que estaban registradas más de una vez en el catálogo, estas se analizaron para ver si su repetición era justificada o no. Por ejemplo, la tienda llamada **Chapultepec** tenía 4 registros distintos, los cuales tenían distinta ubicación (zona, ciudad y estado), por ende, lo único que se debía de hacer era cambiar los nombres de los puntos de venta de tal manera que estas tiendas se pudieran identificar por separado, sin embargo, hubo casos en que la tienda se repetía con los mismos valores, por lo tanto, los registros extras se eliminaban.

2. Chapultepec

```r
#cambiar "nuevo nombre" porque son diferentes establecimientos
nuevo_catalogo%>%filter(`NOMBRE DEL PDV`=="chapultepec")
```

NOMBRE DEL PDV <chr>	NUEVO NOMBRE DEL PDV (Operacione <chr>	ESTADO <chr>	CIUDAD <chr>	LATITUD_NUM <dbl>	LONGITUD_NUM <dbl>	Zona <chr>
chapultepec	NA	nuevo leon	monterrey	25.66638	-100.28007	noreste
chapultepec	NA	jalisco	guadalajara	20.67431	-103.36829	centro occidente
chapultepec	NA	morelos	cuernavaca	18.92283	-99.21053	centro sur

3 rows

```r
which(nuevo_catalogo$`NOMBRE DEL PDV`=="chapultepec")
#577 836 1278
```

```
[1]  577  836 1278
```

```r
#Renombrar puntos de venta para distinguirlos dado que corresponden a diferentes estados
nuevo_catalogo[577,2]<-"chapultepec mty"
nuevo_catalogo[836,2]<-"chapultepec gdl"
nuevo_catalogo[1278,2]<-"chapultepec mrls"
```

Figure 4: Ejemplo de tiendas repetidas a ser renombradas.

[3] http://www.conabio.gob.mx/informacion/gis/layouts/recomgw.png

4. Paseo interlomas

```{r}
nuevo_catalogo%>%filter(`NOMBRE DEL PDV`=="paseo interlomas")
```

NOMBRE DEL PDV <chr>	NUEVO NOMBRE DEL PDV (Operacione <chr>	ESTADO <chr>	CIUDAD <chr>	LATITUD_NUM <dbl>	LONGITUD_NUM <dbl>	Zona <chr>
paseo interlomas	NA	estado de mexico	huixquilucan	19.39734	-99.28129	centro sur
paseo interlomas	NA	estado de mexico	huixquilucan	19.40148	-99.27431	centro sur
paseo interlomas	NA	estado de mexico	huixquilucan	19.39742	-99.29236	centro sur

3 rows

```{r}
which(nuevo_catalogo$`NOMBRE DEL PDV`=="paseo interlomas")
#1293 1786 1792 hay que remover los últimos 2 porque son iguales
```

```
[1] 1293 1786 1792
```

Figure 5: Ejemplo de tiendas repetidas donde dos deben de ser eliminadas.

Tras esta limpieza se genera un nuevo catálogo con 2,783 puntos de venta distintos y cada uno de ellos con su información geográfica completa.

punto_de_venta <chr>	estado <chr>	ciudad <chr>	latitud <dbl>	longitud <dbl>	zona <chr>
1 poniente	puebla	tehuacan	18.46210	-97.39496	centro sur
5 de mayo zmm	michoacan	zamora	19.98131	-102.28329	centro occidente
abasolo coahuila	coahuila	saltillo	25.41984	-100.99173	norte
acambaro	guanajuato	acambaro	20.02042	-100.73045	centro occidente
acapulco centro	guerrero	acapulco	16.85070	-99.90670	pacifico sur
acayucan	veracruz	oluta	17.93999	-94.91060	golfo de mexico
acceso norte reducido	san luis potosi	soledad de graciano	22.17022	-100.96278	norte
ace adolfo lopez leon	guanajuato	leon	21.13974	-101.68606	centro occidente
ace aldama centro	guanajuato	leon	21.12395	-101.68078	centro occidente
ace alhondiga	guanajuato	guanajuato	21.02477	-101.25900	centro occidente

1-10 of 2,783 rows Previous 1 2 3 4 5 6 ... 100 Next

Figure 6: Catálogo final.

5.4 Limpieza de Datos: Registro de ventas

Una vez que se detectaron los problemas de calidad en el registro de ventas se procede con su limpieza. A continuación se presentan las actividades principales que se llevaron a cabo.

- **Homogeneizar los registros a minúsculas**: Todos los registros se convierten a minúsculas para facilitar su manejo.
- **Imputar valores faltantes**:
  - La columna de **precio** tenía muchos valores faltantes, sin embargo, se habla con un experto de IBM en la industría de telecomunicaciones y se llega al acuerdo de únicamente utilizar una variable entre precio y costo dado que los valores de estas dos variables eran prácticamente los mismos, por ende, esta columna se descarta.
  - La columna de **costo** tenía 7 valores faltantes, los cuales se imputaron al buscar en los demás registros el sku de los productos con costos faltantes, luego se filtraron los resultados por punto de venta y fecha y al final se obtuvieron los valores a imputar.
  - La columna de **marca** tenía 9 valores faltantes, los cuales se imputaron de manera similar al caso anterior, buscando el sku de los productos con valores faltantes en esta columna en los demás registros, y al ser el sku el código único del producto fue muy sencillo encontrar a cuál marca se referían.
- **Homogeneizar el formato de la fecha**: Todos los registros de fecha se homogeneizar de tal manera que todos fueran con el formato: AAAA-MM-DD
- **Remover caracteres especiales**: Todos los acentos se remueven, al igual que la letra **ñ**, los dobles espacios y otros tipos de espacios (Non Breaking Spaces).

- **Corregir tiendas mal escritas**: Hay tiendas que están registradas con un nombre erróneo, por lo tanto, se detectan estás tiendas y se les cambia el nombre al nombre correcto dentro del catálogo.
- **Eliminar errores de registro**:
  - En un principio se tenían 32 marcas distintas en el registro de ventas, sin embargo, esto se debía a que las marcas estaban registradas de manera erronea ya que en lugar de considerar únicamente la marca, algunos registros tenían incluido el modelo del celular, por lo tanto, se seccionan las marcas únicamente en 12.

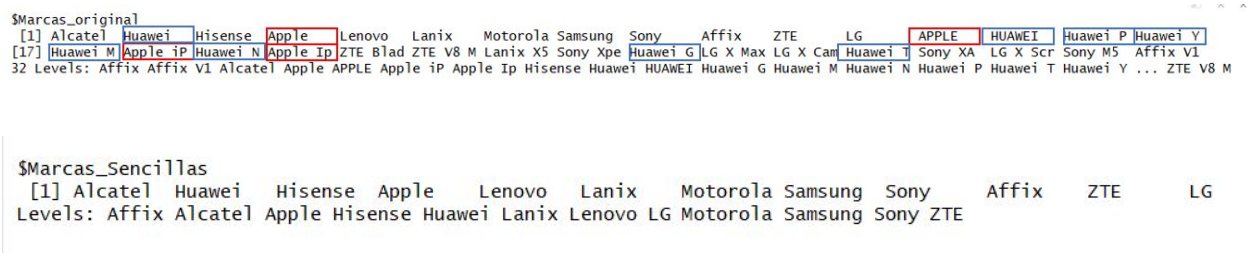


Figure 7: Errores de registro en la columna de Marca.

5.5 Paso final de la limpieza

Como último paso para tener los datos limpios, se hace una fusión de ambas bases de datos para tener un archivo donde se tenga información relacionada con las ventas y con algunas características geográficas de los puntos de venta.

El documento final tras la limpieza tiene 1,048,575 registros de ventas y 15 variables, las cuales son:

- **Variables en el csv limpio:**
  - **Punto de venta**: Nombre del punto de venta.
  - **Plan tarifario**: Plan tarifario al que pertenece la unidad vendida.
  - **Sku por equipo**: Código único del producto vendido. Cabe mencionar que son códigos internos de la compañía ABCD, por lo tanto, no se sabe el nombre comercial de los productos.
  - **Fecha**: Fecha en la que se hizo la venta en formato AAAA-MM-DD.
  - **Costo**: Costo del producto.
  - **Marca**: Marca del producto vendido (32 posibles valores).
  - **Ventas**: Columna con valores iguales a 1. Es decir, cada registro dentro del documento (cada renglón) corresponde a una venta.
  - **Mes**: Mes en el que se realizó la compra del producto.
  - **Anio**: Año en el que se realizó la compra del producto.
  - **Marca modificada**: Marca del producto vendido después de la limpieza (12 valores posibles).
  - **Estado**: Estado en el que se hizo la venta.
  - **Ciudad**: Ciudad en la que se hizo la venta.
  - **Latitud**: Ubicación con coordenadas geográficas del punto de venta.
  - **Longitud**: Ubicación con coordenadas geográficas del punto de venta.
  - **Zona**: Zona en la que se hizo la venta con la división establecida por CONABIO.

5.6 Transformación de los datos

Una vez que ya se tuvieron los datos limpios, hubo dos transformaciones adicionales que se tuvieron que hacer. Este proceso se llevo a cabo en un script en Rstudio que puede encontrarse en la siguiente liga[4] .

[4]https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion/tree/master/Transformacion_De_Datos_E_Ingenieria_De_Caracteristicas

- **Costo promedio por producto:** Tras la limpieza de los datos, se observó que el costo de los productos cambiaba ligeramente en los diferentes periodos de venta (meses), por lo tanto, se optó por obtener el **costo promedio por producto** y utilizar este valor para construir una nueva variable llamada **gamma** con posibles valores: **premium**, **alta**, **media** y **baja**. Esto se hace gracias que se tiene una reunión más con el experto de IBM con respecto a la industría de telecomunicaciones y se discute que para una empresa del sector de telefonía movíl, el tener bien definido a qué gamma pertenece cada producto es muy importante.

Rango de Costo	Gamma correspondiente
Costo $\leq$5,000	Baja
5,000 < Costo $\leq$ 10,000	Media
10,000 < Costo $\leq$ 15,000	Alta
Costo > 15,000	Premium

Figure 8: Rango de valores de la variable de costo con sus respectivas gammas.

- **Agrupación:** Una vez que ya se tuvieron todos los campos limpios y homogeneizados, se procede a hacer una agrupación por **punto de venta, fecha, mes, anio, sku, marca, gamma, zona, estado, ciudad, latitud, longitud**, y después se suman el número de ventas que cumplen con las agrupaciones pasadas. Por ende, se pasa de tener un archivo con 1,048,575 renglones correspondientes a registros individuales, a un archivo con un total de 932,963 renglones correspondientes a registros agrupados.

6 Análisis exploratorio de los datos

A continuación, se procede con el **Análisis Exploratorio de los Datos** que busca extraer información relevante de los datos. A este punto del proyecto, ya se cuenta con un archivo limpio con un total de 932,963 observaciones y 13 variables, las cuales se van a analizar para extraer la mayor cantidad posible de información para determinar el escenario general en el que se encuentra la compañia **ABCD**.

Variables						
1. Punto de venta	2. Fecha	3. Mes	4. Anio	5. Sku	6. Marca	7. Gamma
8. Zona	9. Estado	10. Ciudad	11. Latitud	12. Longitud	12. Ventas Diarias	

Figure 9: Variables de los datos limpios.

Para realizar este análisis fue necesario crear un código en RStudio; las gráficas y resultados que se presentan enseguida fueron los más relevantes del análisis, sin embargo, si se quiere ver todo lo que se hizo, en esta liga[5]se puede consultar el código completo.

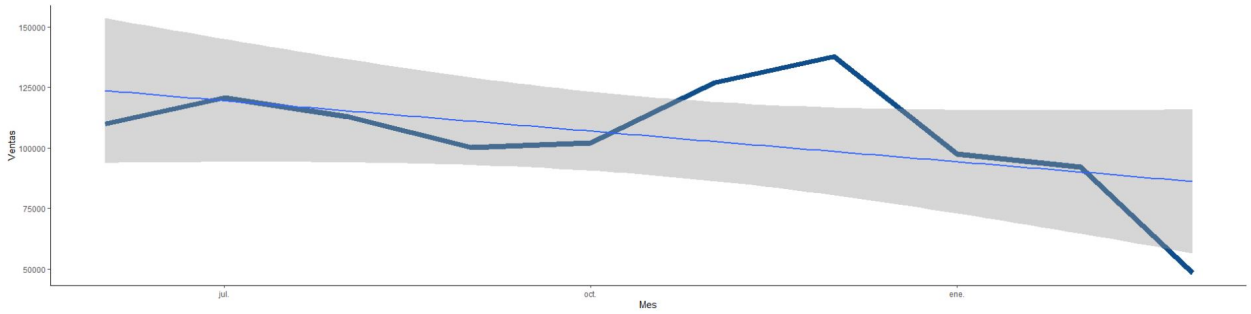
En primer lugar, se presenta una tabla donde se identifican los valores únicos dentro de las variables de: **Punto de venta, sku, marca, gamma, zona, estado y ciudad, es decir, se identifican los distintos puntos de venta, productos, marcas, gammas, zonas, estados y ciudades** en las que se realizan ventas de productos de telefonía celular. Además, se menciona el rango de fecha que abarcan los datos.

Características de los datos	Valores
Puntos de venta:	1,909 tiendas en todo el territorio nacional
Rango de fecha de los datos:	01 de junio de 2018 al 31 de marzo de 2019
	- 10 meses de registro
Productos:	455 productos distintos, cada uno identificado por un código único (SKU)
Marcas:	12 marcas telefónicas que vende la compañía ABCD.
	- Affix, Alcatel, Apple, Hisense, Hauawei, Lannix, Lenovo, LG, Motorola, Samsung, Sony, y ZTE.
Gammas de producto:	4 gammas en la que fueron clasificadas los productos de acuerdo a su costo promedio,
	- Premium, Alta, Media y Baja.
Zonas en las que esta divido el territorio:	8 zonas
	- Centro sur, Centro Occidente, Golfo de México, Norte, Pacífico Sur, Península de Yucatán, Noreste y Noroeste.
Estados :	32 estados de la república mexicana en los que tiene presencia la compañía.
Ciudades :	228 ciudades en las que tiene presencia la compañía.

Figure 10: Valores que adoptan las variables de los datos.

A continuación, se hace un breve análisis sobre el comportamiento de las ventas. El primer análisis que se hace es para observar como se comportan las ventas a nivel general de la compañia ABCD dependiendo del mes; en la gráfica siguiente se puede observarque hay un incremento de ventas en los meses de noviembre y diciembre, seguido por una caida drástica en los 3 meses siguientes. También se puede observar que la linea de regresión que fue incluida en la gráfica tiene pendiente negativa, lo cuál podría indicr que las ventas en general han estado disminuyendo.

[5]https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion/tree/master/Analisis_Exploratorio_De_Datos



El siguiente análisis se hace con respecto a las diferentes divisiones geográficas que se tienen en los datos. Con estas representaciones gráfica se puede observar que las 3 zonas con mayor número de venta en los 10 meses de registro son: **Centro sur con 512,223 ventas (48.85% de las ventas totales), centro occidente con 167,655 ventas (15.99%) y noroeste con 85,779 ventas (8.18%)**; las zonas con menos ventas registradas en los 10 meses de registro son: **Golfo de México (6.31%), Península de Yucatán(4.18%) y Pacífico Sur (2.67%)**. De la misma manera, se puede observar que de los 32 estados de la República Mexicana, **la ciudad de México, el estado de México y Jalisco** son los estados con mayor número de ventas en los 10 meses de reegistro con **207,187**, **174,189** y **71,879** unidades vendidas respectivamente; Y los estados con menor número de ventas son: **Durango, Baja California Sur y Zacatecas**, con **5,213**, **4,781**, y **4,633** unidades vendidas respectivamente.

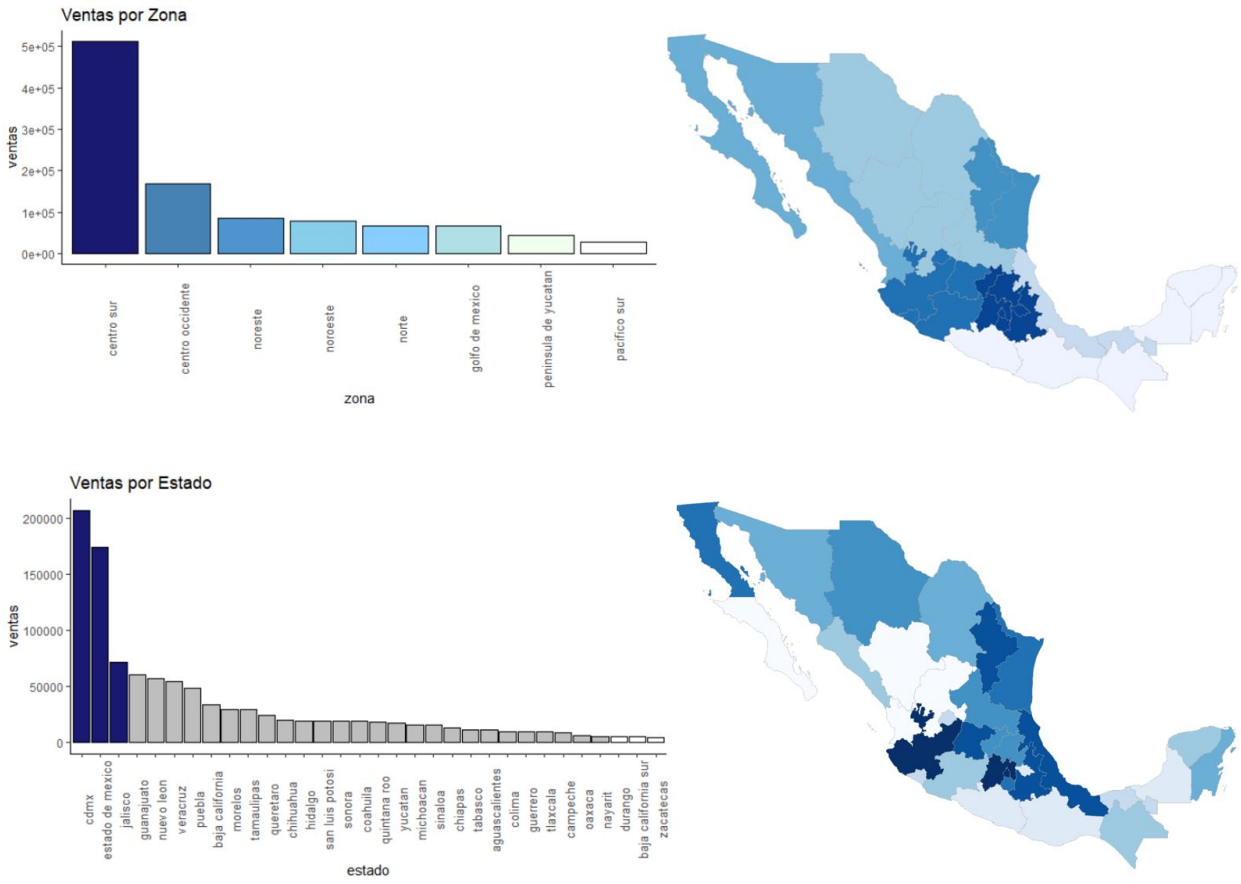


Figure 11: Ventas registradas por zona geográfica y estado en los 10 meses de registro.

Una vez que se tiene esta información a nivel general y geográfico, se puede determinar cuál es contexto de las ventas con respecto a las marcas que maneja la compañía ABCD. Con la siguiente gráfica se puede

observar que la marca que ha tenido mayor número de ventas en los 10 meses de registro fue: **Huawei** con 408,179 unidades vendidas, seguido por Motorola y Samsung con 305,979 y 156,017 unidades vendidas respectivamente. También se puede observar que Sony, Affix y Lenovo son las marcas con menor número de unidades vendidas ya que juntas abarcan menos del 1% de las ventas totales registradas en los 10 meses.

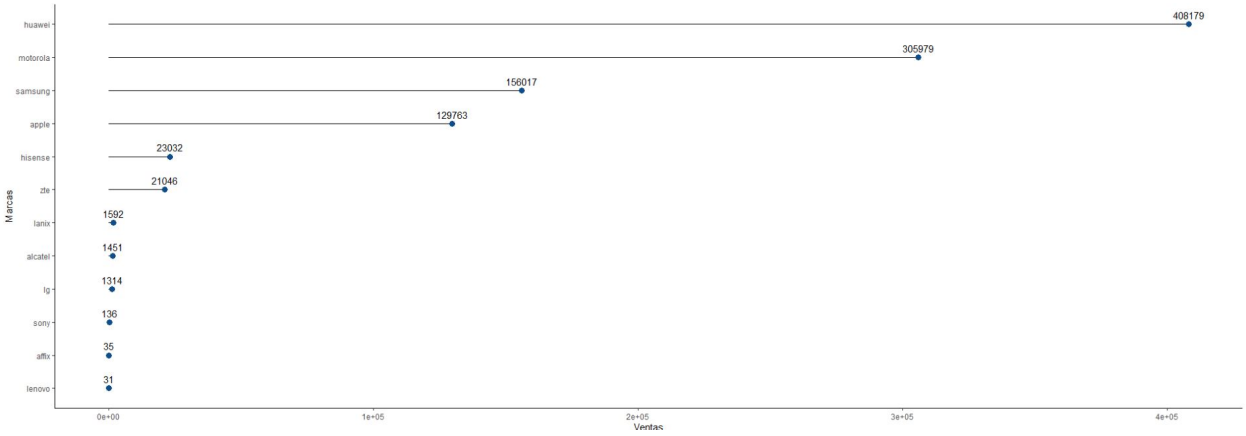


Figure 12: Ventas registradas por marca en los 10 meses de registro.

Como análisis adicional se genera la siguiente gráfica donde se muestra el comportamiento mensual de cada una de las marcas, los cuáles claramente difieren no únicamente en volumen de unidades vendidas, sino en su temporalidad.

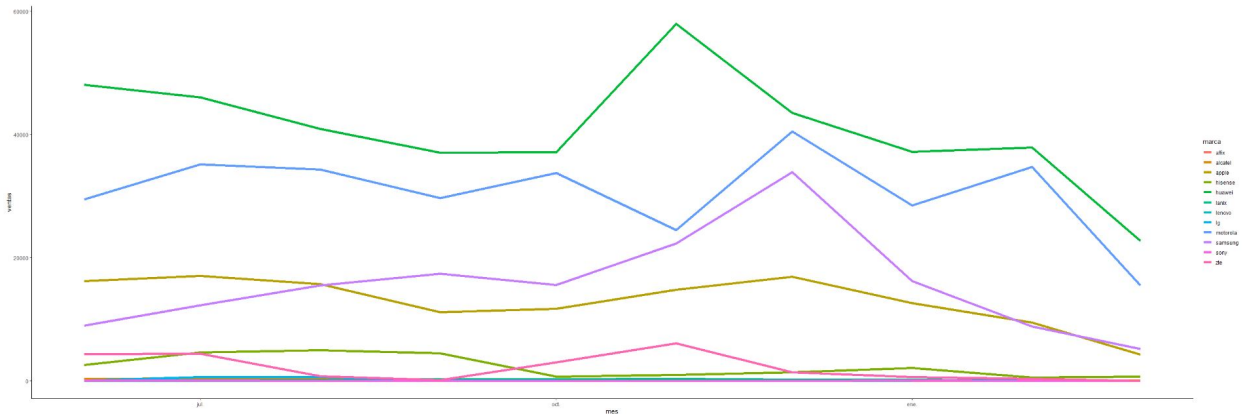


Figure 13: Comportamiento de las ventas por mes dependiendo de cada marca.

Finalmente, se hacen dos gráficas, donde la primera plasma las ventas que se tienen de cada marca en cada estado, con lo cual se puede observar que no todos los estados compran la misma cantidad de los mismos productos. Mientras que lal segunda gráfica plasma que el comportamiento de las ventas por estado y por marca también difieren en el tiempo.

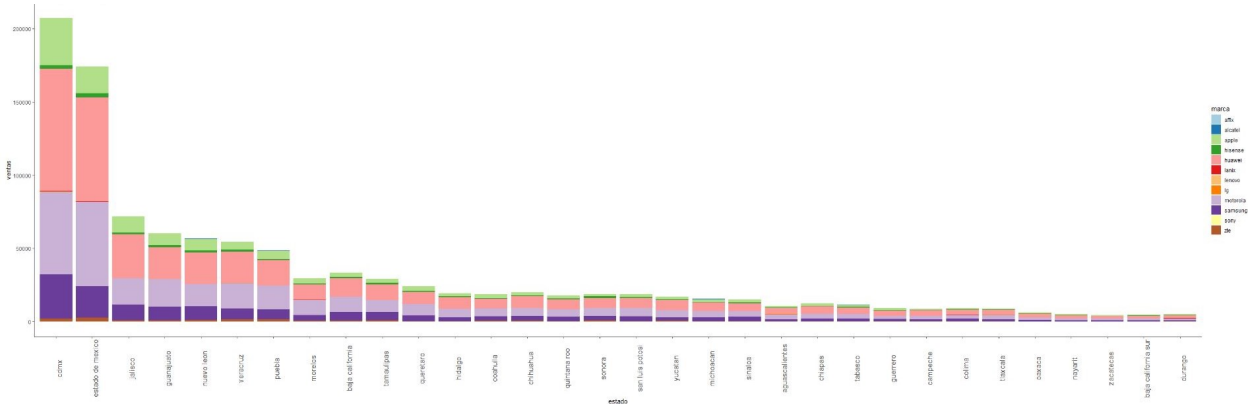


Figure 14: Ventas registradas por marca y estado.

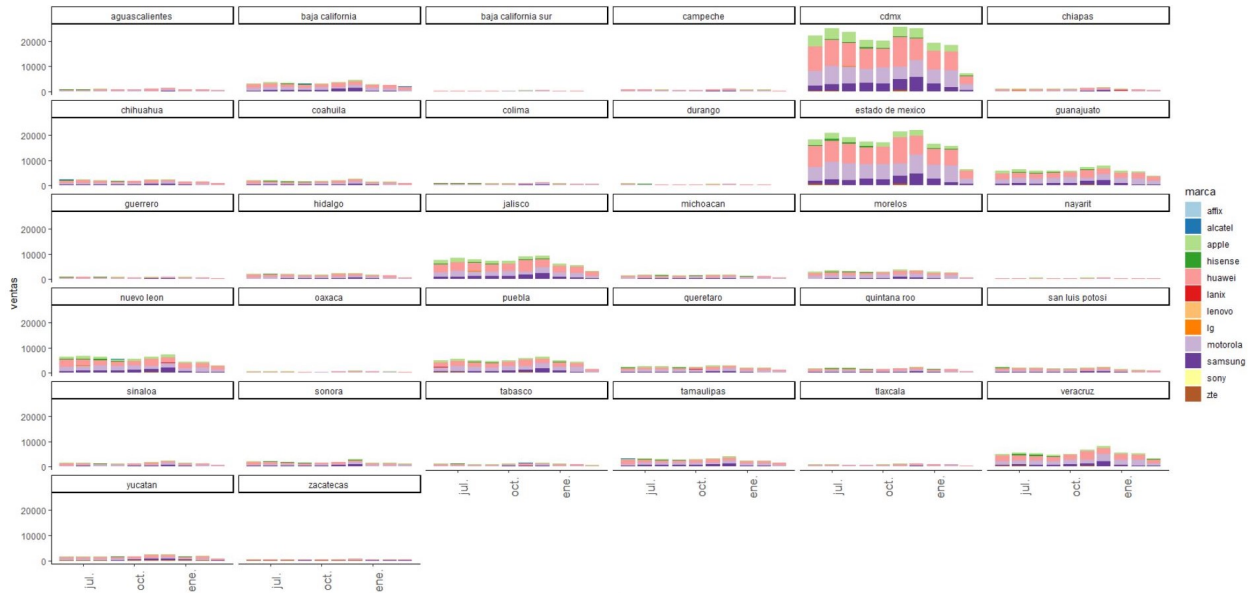


Figure 15: Ventas registradas por marca, estado y mes.

Con este breve análisis lo más notorio que se puede recalcar es que las ventas tienen un comportamiento dinámico, es decir, que cambian respecto al tiempo, lo cual invita a suponer que cada punto de venta requiere de diferentes cantidades y tipos de productos mensualmente. Gracias a este análsis exploratio fue más claro entender el porqué la compañía ABCD tiene la problemática de cómo construir sus portafolios de telefonía celular.

7 Ingeniería de Características

Con una idea bien establecida del escenario general en el que se encuentra la compañía **ABCD**, se puede comenzar a tratar de crear más caracteristicas a partir de las que se tienen con el fin de mejorar la eficacia predictiva de los algoritmos propuestos en la siguiente la sección. Dichos algoritmos buscan cumplir con el objetivo del proyecto que es **mejorar la construcción de portafolios de productos de telefonía celular de la compañía ABCD** donde lo que se busca predecir es **la cantidad de unidades de producto que se van a vender en cada punto de venta al siguiente mes de análisis**. Para ello y dado lo planteado se lleva a cado el siguiente proceso; al igual que en las secciones pasadas, lo plasmado en este documento es la información resumida, si se quiere ver a detalle el código para ejecutar lo siguiente se puede revisar esta liga [6].

- **Construcción de índices:** Para comenzar y para facilitar el uso de variables categóricas dentro de los algoritmos, se crean índices para las variables: **punto de venta**, **sku**, **marca**, **gamma**, y **fecha**.

Característica	Posibles valores
Punto de venta	1 a 1,909 - *Asignados dependiendo del orden descendente de los nombres de los puntos de venta (A-Z).*
Marca	1 a 12 - *Asignados dependiendo del orden descendente de los nombres de las marcas (A-Z).*
Gamma	1 a 4 - *Asignados dependiendo de su valor en tanto a costo promedio.* - *4:Premium, 3:Alto, 2:Medio y 1:Alto*
Bloque de fecha	*0 a 9* - *Asignados dependiendo del mes de registro.* - *El mes más antiguo va a tener el valor más pequeño y el valor más alto va a ser el último mes del que se tiene registro.* - *0:junio 2018, 1:julio 2018, 2:agosto 2018, 3:septiembre 2018, ... , 8:abril 2019, y 9:marzo 2019*
Sku	1 a 455 - *Asignados dependiendo del orden descendente de los códigos únicos de los productos (A-Z).*

Figure 16: Índices para variables categóricas.

- **Agrupación:** Se hace una agrupación por **punto de venta**, **sku**, **marca**, **gamma**, y **fecha**, para calcular el número total de ventas relacionadas con estas caracteristicas, es decir, construir la variable que se quiere predecir más adelante (**número de unidades a vender al siguiente mes**).
- **Completar serie de tiempo:** Con la agrupación anterior se puede observar que la serie de tiempo no esta completa, es decir, hay bloques de meses (0-9) e indices de productos que no aparecen en todos los puntos de venta, por lo tanto, esto se debe de completar.

[6] https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion/tree/master/Transformacion_De_Datos_E_Ingenieria_De_Caracteristicas

- Lo primero que se hace es obtener las combinaciones existentes entre **punto de venta**, **sku** y **bloque de fecha**, únicamente estas tres variables son consideradas para obtener el número total de combinaciones dado que cada sku tiene asociado únicamente una **gamma** y una **marca**. El número total de combinaciones y de registros de la serie de tiempo completa es: 8,685,950.
- A continuación se relacionan los valores obtenidos en la agrupación con la serie de tiempo y los valores nulos significan que no hubo ventas registradas con esas caracteristicas (**punto de venta**, **sku**, **marca**, **gamma**, y **fecha**).

Apartir de este punto, la serie de tiempo ya esta completa, sin embargo, para enriquecer los modelos propuestos se hacen conteos y promedios por duplas de caracteristicas, esto con el fin de recopilar información adicional.

- **Conteos por grupo:** Se sacan conteos y promedios de 4 duplas de caracteristicas:
  - Ventas promedio por tienda por mes.
  - Ventas promedio por marca por mes.
  - Ventas promedio por gamma por mes.
  - Ventas promedio por producto por mes.
  - Ventas totales por tienda por mes.
  - Ventas totales por marca por mes.
  - Ventas totales por gamma por mes.
  - Ventas totales por producto por mes.
- **Rezagos a tres tiempos :** La variables anteriores no pueden emplearse en un modelo dado que al momento de querer hacer la predicción para el mes siguiente (10 en este caso), no se va a tener información sobre cuántas ventas promedio por tienda se tuvieron ese mes ya que es algo que aún no pasa, por lo tanto, se opta por crear rezagos a 3 tiempos (1 mes, 2 meses y 3 meses) para contar con información del pasado para hacer las predicciónes.
  - Por ejemplo, para el mes 10 se van a tener como variables adicionales: Ventas promedio por tienda del mes 9, ventas promedio por tienda del mes 8, ventas promedio por tienda del mes 7, y así sucesivamente con los demás conteos.

Finalmente se tiene el archivo a ocupar con los modelos donde se consideran: 30 Variables y 8,685,950 registros.

Variables	Explicación		
Índices de los puntos de venta:	1,909 índices cada uno relacionado a un punto de venta.		
Bloques de fecha:	10 bloques de fecha (uno por cada mes de registro). *- 0:junio 2018, 1:julio 2018, 2:agosto 2018, 3:septiembre 2018, … , 8:abril 2019, y 9:marzo 2019*		
Índices de los productos:	455 índices cada uno correspondiente a un producto distinto.		
Marcas:	12 índices relacionados con las marcas telefónicas que vende la compañía ABCD. *- Affix, Alcatel, Apple, Hisense, Hauawei, Lannix, Lenovo, LG, Motorola, Samsung, Sony, y ZTE.*		
Gammas de producto:	4 gammas dependiendo del producto y el costo promedio de este. *- Premium, Alta, Media y Baja.*		
24 rezagos	*Mes anterior (Rezago:1)*	*Hace dos meses (Rezago:2)*	*Hace 3 meses (Rezago:3)*
	Ventas promedio por tienda del mes anterior	*Ventas promedio por tienda de hace 2 meses*	*Ventas promedio por tienda de hace 3 meses*
	ventas promedio por marca del mes anterior	*ventas promedio por marca de hace 2 meses*	*ventas promedio por marca de hace 3 meses*
	ventas promedio por gamma del mes anterior	*ventas promedio por gamma de hace 2 meses*	*ventas promedio por gamma de hace 3 meses*
	ventas promedio por producto del mes anterior	*ventas promedio por producto de hace 2 meses*	*ventas promedio por producto de hace 3 meses*
	ventas totales por tienda del mes anterior	*ventas totales por tienda de hace 2 meses*	*ventas totales por tienda de hace 3 meses*
	ventas totales por marca del mes anterior	*ventas totales por marca de hace 2 meses*	*ventas totales por marca de hace 3 meses*
	ventas totales por gamma del mes anterior	*ventas totales por gamma de hace 2 meses*	*ventas totales por gamma de hace 3 meses*
	ventas totales por producto del mes anterior	*ventas totales por producto de hace 2 meses*	*ventas totales por producto de hace 3 meses*
Ventas	Variable a predecir		

Figure 17: Variables a considerar en la sección de modelado.

8 Modelos

Para realizar esta parte del proyecto y cumplir con el objetivo de este, se propusieron 3 modelos de regresión: **árboles de decisión**, **bosques aleatorios** y **XGBoost**. Los tres modelos utilizan el mismo conjunto de datos limpios, transformados y con ingeniería de caractéristicas.

Es importante mencionar que para llevar a cabo esta parte del proyecto, cada modelo fue desarrollado en un jupyter notebook los cuales se pueden consultar en esta liga[7].

8.1 Planteamiento del problema

Dado que los modelos que se proponen son modelos de aprendizaje de máquina, es importante definir brevemente qué es el aprendizaje de máquina. Aprendizaje de máquina son métodos computacionales para aprender de los datos con el fin de mejorar el desempeño de una tarea[8].

En este proyecto se tiene un problema que requiere de modelos de aprendizaje supervisado, es decir, modelos que tienen como tarea predecir o estimar una variable respuesta a partir de ciertos datos de entrada, también se clasifica como un problema de regresión dado que la variable respuesta es de valores continuos y no de asignación de clase (clasificación).

Ya que se definió que el problema al que se enfrenta la compañía **ABCD** es un problema de regresión que requiere de modelos de aprendizaje supervisado, se procede a identificar las partes que se requieren para estructurar un modelo.

En primer lugar, se debe de definir la variable respuesta Y, en el caso de la construcción de portafolios de telefonía celular de la compañia ABCD, la variable respuesta Y es **Ventas totales por tienda, mes y producto**. Seguido, hay que definir cuáles serán los datos de entrada que permitirán que los modelos aprendan, en este caso se cuenta con 29 variables independientes de las cuales 24 son rezagos de conteos y promedios, y las otras 5 son variables categóricas que ayudan a identificar ciertas características de cada producto vendido.

Como siguiente paso al planteamiento del problema, es necesario definir las métricas a las cuáles se van a someter los modelos y así poder ser comparables unos con otros y llegar a un consenso final sobre el modelo con el mejor desempeño.

8.2 Métricas

Antes de comenzar con los modelos, es necesario definir el tipo de métricas que se necesitan para comparar los modelos. Para este proyecto se seleccionan 3 métricas, las cuales son de las más comunes a emplear cuando se trata de medirla precisión de modelos de regresión cuya variable respuesta es continua:

- **Error Absoluto Medio:** Mide el promedio de la magnitud de los errores en un conjunto de predicciones.
- **Raíz del Error Cuadrático Medio:** Muy parecido al error absoluto medio, esta métrica permite medir la magnitud del error con la diferencia de que esta métrica tiene la particularidad de dar mayor peso a errores más grandes.
- **Error Cuadrático Medio:** Al igual que la raíz error cuadrático medio, esta métrica mide la magnitud del error de predicción, sin embargo, las unidades de esta métrica terminan siendo cuadráticas, por ende, su interpretación no es tan fácil.

En el contexto del proyecto, lo que se busca es encontrar el modelo que tenga el mejor desempeño con relación a estas métricas, es decir, el modelo cuyas métricas sean las menores.

[7] https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion/tree/master/Modelado
[8] https://felipegonzalez.github.io/aprendizaje-maquina-mcd-2018/introduccion.html

8.3 Estructura de los modelos

Ya que se definió en su totalidad como va a estar estructurada la problematica, se procede a construir los modelos; y aunque cada modelo tiene sus implicaciones particulares (como formato en el que se ingestan los datos), los tres modelos se dividen de la siguiente manera:

- **Parte 1: Lectura de los Datos:**
  - Los datos se cargan a un proyecto en **Watson Studio** y en esa misma plataforma se crea el jupyter notebook donde se incertan las credenciales apropiadas para extraer los datos del depósito de objetos (object storage).
- **Parte 2: División de los Datos en Entrenamiento y Prueba:**
  - Ya que los datos se hayan leido adecuadamente, se procede a dividir estos en dos conjuntos. El conjunto de entrenamiento que abarca los registros correspondientes a los bloques de fecha: **0,1,2,3,4,5,6,7,y 8** con un total de **7,817,355** observaciones, y el conjunto de prueba abarca únicamente los registros del bloque de fecha: **9** con un total de **868,595** observaciones.
- **Parte 3: construcción del modelo:**
  - Una vez que los datos ya se dividieron en conjunto de entrenamiento y prueba, se continua con la construcción de cada uno de los modelos. Más adelante se especifica lo que fue necesario para ejecutar cada modelo, sin embargo, en este punto se describen las caracteristicas generales que se tuvieron que tomar en cuenta para la construcción de cada modelo.
  - **Aplicar validación cruzada para series de tiempo:**
  - Se sabe que validación cruzada es una técnica popular para evaluar los resultados de un modelo, particionando el conjunto de entrenamiento en: entrenamiento y validación para después calcular la media de las medidas de evaluación sobre las diversas particiones. Una de las principales razones para usar validación cruzada es que no se tienen los suficientes datos para el conjunto de entrenamiento y prueba, y como no se quiere tener sobreajuste, las particiones que hace validación cruzada son muy útiles. Sin embargo, cuando se trata de un problema de series de tiempo donde existe dependencia temporal, la tarea se vuelve un poco más compleja ya que las particiones deben de respetar la cronología de los eventos dentro de las observaciones.
  - **Aplicar un GridSearch:** Uno de los factores más importantes a considerar cuando se están construyendo modelos de aprendizaje de máquina es que el ajuste y la afinación de parámetros. Es por eso que cada uno de los modelos propuestos utilizan diversas combinaciones de parámetros con el fin de encontrar el que mejor se desempeñe.
  - **Entrenamiento y Evaluación del modelo con los diferentes parámetros propuestos:** Como ya se mencionó en el punto anterior, de los tres modelos propuestos cada uno se construyen con diversos parámetros. En total cada modelo tiene 8 combinaciones de parámetros que se utilizan para entrenar 8 modelos y de estos se obtienen las métricas mencionadas con anterioridad.
  - **Selección del mejor conjunto de parámetros:** Ya que se tienen las puntuaciones de los modelos con respecto a sus metricas de error, se selecciona el modelo cuyos errores tienen el menor valor.
  - **Reentrenamiento con los parámetros que obtuvieron el mejor desempeño:** Una vez que el mejor modelo ya fue seleccionado, este se reentrena con los parámetros con el mejor desempeño para después utilizarse con el conjunto de prueba.
  - **Evaluación del mejor modelo con el conjunto de prueba:** Ya que el modelo fue entrenado con los mejores parámetros obtenidos, se utiliza el modelo con el conjunto de datos para ver el resultado que obtiene el modelo con un conjunto de datos nuevo.

Finalmente, se comparan los resultados con el conjunto de prueba de los 3 modelos y se selecciona el mejor como solución para la problemática de la compañía ABCD.

Ya que se definió la estructura que siguen los modelos y las caracteristicas importantes a tomar en cuenta, a continuación se presentan cada uno de los modelos con sus respectivos resultados.

8.4 Árbol de Decisión

El primer modelo que se propuso fue un **árbol de decisión (decision tree)**. Para construir este modelo fue necesario aplicar validación cruzada para series de tiempo con 3 particiones y en seguida se definió un GridSearch donde los parámetros a cambiar fueron:

- **Max Depth:** Profundidad máxima del árbol. Para este problema de consideraron dos valores: $[2, 5]$.
- **Valor mínimo de muestra:** El valor mínimo de muestras requeridas para particionar un nodo interno. Para este problema se probaron también 2 valores distintos: $[2, 10]$.
- **Max leaf node**De la misma manera, se probaron dos valores distintos: $[3, 10]$.

Con estos diferentes parámetros se tiene como resultado un total de 8 modelos a probar para determinar la combinación que de el mejor resultado con respecto a las métricas requeridas (MAE, MSE y RMSE). Los resultados después de entrenar estos 8 modelos fueron:

Conjunto de Entrenamiento: Árbol de Decisión

Modelo	Max_Depth	Max_leaf_nodes	Min_sample_splits	MAE	MSE	RMSE
0	2	3	2	0.180818	0.477934	0.6913277
1	2	3	10	0.180818	0.477934	0.6913277
2	2	10	2	0.157768	0.423271	0.6505928
3	2	10	10	0.157768	0.423271	0.6505928
4	5	3	2	0.180818	0.477934	0.6913277
5	5	3	10	0.180818	0.477934	0.6913277
6	5	10	2	0.156878	0.421043	0.6488783
7	5	10	10	0.156878	0.421043	0.6488783

Figure 18: Resultados del GridSearch con el conjunto de entrenamiento.

Con la tabla anterior se puede observar que el mejor modelo es el número **6** con parámetros **max_depth: 5**, **min_samples_split: 2**, y **max_leaf_nodes: 10**. Estos parámetros se seleccionan y se vuelve a entrenar el modelo ahora con estos valores. Una vez que se ya se reentrenó el modelo con los parámetros que dieron el mejor resultado con el conjunto de entrenamiento, se prueba el modelo ahora con el conjunto de prueba. Los resultados que se obtuvieron fueron:

Conjunto de Prueba: Árbol de Decisión

Max_Depth	Max_leaf_nodes	Min_sample_splits	MAE	MSE	RMSE
5	10	2	0.075631	0.238847	0.488719

Figure 19: Resultados del mejor árbol de decisión con el conjunto de prueba.

Este modelo con sus respectivos resultados es el que se va a comparar con los demás para así determinar cuál fue el que dio mejor resultado.

8.5 Bosque Aleatorio

El siguiente modelo que se propuso fue un **bosque aleatorio (random forest)**. Al igual que el modelo anterior, fue necesario aplicar validación cruzada para series de tiempo con 3 particiones con el fin de evitar sobreajuste. En seguida se definió un GridSearch donde los parámetros a cambiar fueron:

- **Profundidad máxima:**Profundidad máxima del árbol. Para este problema de consideraron dos valores: $[2, 9]$.
- **Valor mínimo de muestra:** El valor mínimo de muestras requeridas para particionar un nodo interno. Para este problema se probaron también 2 valores distintos: $[8, 10]$.
- **n estimator:** Número de árboles en el bosque. De la misma manera, se probaron dos valores distintos: $[2, 3]$\$.

Con estos diferentes parámetros se tiene como resultado un total de 8 modelos a probar para determinar la combinación que de el mejor resultado con respecto a las métricas requeridas (MAE, MSE y RMSE). Los resultados después de entrenar estos 8 modelos fueron:

Conjunto de Entrenamiento: Bosque Aleatorio

Modelo	Max_Depth	N_estimators	Min_sample_splits	MAE	MSE	RMSE
0	2	2	8	0.171287	0.464193	0.6813171
1	2	3	8	0.171460	0.463203	0.6805902
2	2	2	10	0.171025	0.462043	0.6797374
3	2	3	10	0.171204	0.461954	0.679672
4	9	2	8	0.147769	0.459278	0.6777005
5	9	3	8	0.146789	0.439155	0.6626877
6	9	2	10	0.159958	0.528024	0.7266526
7	9	3	10	0.163301	0.468133	0.6842025

Figure 20: Resultados del GridSearch con el conjunto de entrenamiento.

Con la tabla anterior se puede observar que el mejor modelo es el número **5** con parámetros **max_depth: 9**, **min_samples_split: 8**, y **n_estimators: 3**. Estos parámetros se seleccionan y se utilizan para volver a entrenar el modelo (random forest regressor) ahora con estos valores. Una vez que se ya se reentrenó el modelo con los parámetros que dieron el mejor resultado con el conjunto de entrenamiento, se prueba el modelo ahora con el conjunto de prueba. Los resultados que se obtuvieron fueron:

Conjunto de Prueba: Bosque Aleatorio

Max_Depth	N_estimators	Min_sample_splits	MAE	MSE	RMSE
9	3	8	0.088400	0.259616	0.509526

Figure 21: Resultados del mejor bosque aleatorio con el conjunto de prueba.

Este modelo con sus respectivos resultados es el que se va a comparar con los demás para así determinar cuál fue el que dio mejor resultado.

8.6 Extreme Gradient Boosting (XGBoost)

El último modelo que se propuso fue un **Extreme Gradient Boosting (XGBoost)**. Al igual que los modelos anteriores, fue necesario aplicar validación cruzada para series de tiempo con 3 particiones con el fin de evitar sobreajuste. En seguida se definió un GridSearch donde los parámetros a cambiar fueron:

- **Profundidad máxima:**Profundidad máxima del árbol. Para este problema de consideraron dos valores: $[2, 3]$.
- **Tasa de aprendizaje:** $[0.0003, 0.003]$.
- **n estimator:** Número de árboles. De la misma manera, se probaron dos valores distintos: $[2, 3]$.

Con estos diferentes parámetros se tiene como resultado un total de 8 modelos a probar para determinar la combinación que de el mejor resultado con respecto a las métricas requeridas (MAE, MSE y RMSE). Los resultados después de entrenar estos 8 modelos fueron:

Conjunto de Entrenamiento: XGBoost

Modelo	Max_Depth	N_estimators	Learning_rate	MAE	MSE	RMSE
0	2	2	0.0003	0.561577	0.727560	0.8529713
1	2	3	0.0003	0.561457	0.727392	0.8528728
2	3	2	0.0003	0.561566	0.727503	0.8529379
3	3	3	0.0003	0.561439	0.727307	0.852823
4	2	2	0.003	0.559405	0.724562	0.8512121
5	2	3	0.003	0.558203	0.722912	0.8502423
6	3	2	0.003	0.559287	0.723995	0.850879
7	3	3	0.003	0.558026	0.722065	0.8497441

Figure 22: Resultados del GridSearch con el conjunto de entrenamiento.

Con la tabla anterior se puede observar que el mejor modelo es el número **7** con parámetros **max_depth: 3**, **learning rate: 0.003**, y **n_estimators: 3**. Estos parámetros se seleccionan y se utilizan para volver a entrenar el modelo (xgboost regressor) ahora con estos valores. Una vez que se ya se reentrenó el modelo con los parámetros que dieron el mejor resultado con el conjunto de entrenamiento, se prueba el modelo ahora con el conjunto de prueba. Los resultados que se obtuvieron fueron:

Conjunto de Prueba: XGBoost

Max_Depth	N_estimators	Learning_rate	MAE	MSE	RMSE
3	3	0.003	0.061025	0.2278254	0.477310

Figure 23: Resultados del mejor XGBoost con el conjunto de prueba.

Este modelo con sus respectivos resultados es el que se va a comparar con los demás para así determinar cuál fue el que dio mejor resultado.

8.7 Modelo seleccionado

Tras correr los 24 modelos anteriores, se observa en la siguiente tabla el mejor de cada tipo de modelo. De esto se concluye que el mejor modelo que fue construido fue el XGBoos, ya que de todos, este fue el que tuvo mejor desempeño en tanto a las 3 métricas. Este fue el modelo que al final se propuso como solución a la compañía ABCD.

Árbol de Decisión					
Max_Depth	Max_leaf_nodes	Min_sample_splits	MAE	MSE	RMSE
5	10	2	0.075631	0.238847	0.488719

Bosque Aleatorio					
Max_Depth	N_estimators	Min_sample_splits	MAE	MSE	RMSE
9	3	8	0.088400	0.259616	0.509526

XGBoost					
Max_Depth	**N_estimators**	**Learning_rate**	**MAE**	**MSE**	**RMSE**
3	3	0.003	**0.061025**	**0.2278254**	**0.477310**

9 Conclusiones

Este trabajo abarca desde la limpieza de datos hasta el despliegue de resultados de modelos propuestos para ayudar a la compañía ABCD a resolver su problema sobre contrucción de portafoliso de telefonía celular.

Desde un punto de vista productivo dentro de la empresa ABCD, es sorprendente que en estos tiempos, aún existan compañías cuyo sistema de manejo de demanda no haya sido actualizado o siquiera homogeneizado; no hubo mucha información proporcionada por la compañía ABCD, sin embargo, se sabe que esta no poseía un modelo particular para determinar las unidades de producto a vender en cada uno de sus puntos de venta. Dado los avances tecnológicos actuales en áreas como ciencia de datos y aprendizaje de máquina, es de suma importancia que las empresas empiecen a familiarizarse con estos y así seguir manteniendo un nivel competitivo.

Desde un punto de vista académico, es cierto que dentro de un proyecto de ciencia de datos, la mayor parte del tiempo invertido será en la limpieza de datos; el mayor reto en este proyecto con respecto a la limpieza fue que la infromación faltante debía completarse mediante un proceso sumamente artesanal o la infromación extra que fue brindada no tenía un significado preciso dado la falta de un diccionario que explicara lo que abarcaba cada uno de los registros adicionales.

Otro aspecto importante a resaltar, es que herramientas existen para explotar los datos, sin embargo, no todos saben cómo o en qué contexto aplicarlas. Construir los modelos de aprendizaje de máquina fue más fácil desde un código en jupyter notebook, mientras que la limpieza y el procesamiento de los datos se realizó en un código en RStudio.

En forma de resumen, este proyecto presenta la limpieza de datos, su transformación, análIsis exploratorio, ingeniería de caracterÍsticas y modelado. En el proyecto se presentan 3 tipos de modelos de aprendizaje de máquina supervisado para una tarea de regresión. El primero fue un árbol de decisión, seguido por un bosque aleatorio y finalmente un XGBoost. Cada modelo fue provado con diferentes parámetros y, por lo tanto, fueron 24 los modelos que se corrieron en total para llevar a cabo este proyecto.El mejor modelo fue un XGBoost que permite cumplir con el objetivo general de este proyecto que es ayudar a la compañía ABCD a construir sus protafolios de telefonía celular para sus diversos puntos de venta.

Como conclusión, el desarrollo de un proyecto de ciencia de datos requiere de tiempo y suma atención a los detalles. El uso adecuado de estos modelos puede traer consigo un gran número de ventajas; para este caso en particular, el tener un modelo que permita construir los portafolios de la compañia ABCD puede traer consigo un mejor manejo de inventarios, por ende una reducción en costo de almacenamiento y transporte, también puede haber menor deserción de clientes dado que los productos se encuentran donde se requieren, sin embargo, esto no se puede cunatificar en este momento.

Finalmente, el trabajo presente podría mejorarse si se agregan más variables que aporten más información al modelo ya sean proporcionadas por la compañía ABCD, o por variables externas que puedan tener un impacto en las ventas de esta.

10 Código

Si se desean consultar los códigos que se hicieron para llevar acabo este proyecto, estos se encuentran el esta liga[9].

[9] https://github.com/AnaLuisaMasetto/Estancia_de_Investigacion

11 Bibliografía

Aler, R. (2015). DECISION TREE HYPER-PARAMETERS. TUNING DECISION TREES. 26 de septiembre de 2019, de Universidad Carlos III de Madrid Sitio web: http://ocw.uc3m.es/ingenieria-informatica/machine-learning-i/decisiontreeshyperparameters.html

González, L. (2018). ¿Qué es aprendizaje de máquina (machine learning)?. 26 de septiembre de 109, de Github Sitio web: https://felipegonzalez.github.io/aprendizaje-maquina-mcd-2018/introduccion.html

Handika, T. (2017). Practicing Regression Techniques on House Prices Dataset-Part 2. 26 de septiembre de 2019, de Media Sitio web: https://medium.com/@blazetamareborn/practicing-regression-techniques-on-house-prices-dataset-

Kaghazgarian, M. (2018). Decision Tree Regressor on Bike Sharing Dataset. 26 de Septiembre de 2016, de Kaggle Sitio web: https://www.kaggle.com/marklvl/decision-tree-regressor-on-bike-sharing-dataset/comments

Koehrsen, W. (2018). Hyperparameter Tuning the Random Forest in Python. 26 de septiembre 2019, de Medium Sitio web: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-

Lemagnen, K. (2017). Hyperparameter tuning in XGBoost. 26 de septiembre de 2019, de Media Sitio web: https://blog.cambridgespark.com/hyperparameter-tuning-in-xgboost-4ff9100a3b2f

Malik, U. (2018). Cross Validation and Grid Search for Model Selection in Python. 26 de septiembre de 2019, de Stackabuse Sitio web: https://stackabuse.com/cross-validation-and-grid-search-for-model-selection-in-python/

Ruiz, D. (2018). predicting_sales_1c. 26 de septiembre de 2016, de Github Sitio web: https://github.com/Druizm128/predicting_sales_1c

Sammut, C. (2019). Mean Absolute Error. 26 de septiembre de 2019, de Springer Link Sitio web: https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_525

s.a. (2017). XGBRegressor vs. xgboost.train huge speed difference?. 26 de septiembre de 2019, de Stackexchange Sitio web: https://datascience.stackexchange.com/questions/17282/xgbregressor-vs-xgboost-train-huge-speed-differe

s.a. (2016). MAE and RMSE — Which Metric is Better?. 26 de septiembre de 2019, de Media Sitio web: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

s.a. (2019). XGBoost Parameters. 26 de septiembre de 2019, de XGBOOST Sitio web: https://xgboost.readthedocs.io/en/latest/parameter.html

s.a. (2019). COMUNICADO DE PRENSA NÚM. 179/19. 26 de septiembre de 2019, de INEGI Sitio web: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2019/OtrTemEcon/ENDUTIH_2018.pdf

scikit-learn developers. (2019). Decision Tree Regression. 26 de septiembre de 2019, de scikit-learn Sitio web: https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py

scikit-learn developers. (2019). sklearn.tree.DecisionTreeRegressor. 26 de septiembre de 2019, de scikit-learn Sitio web: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html