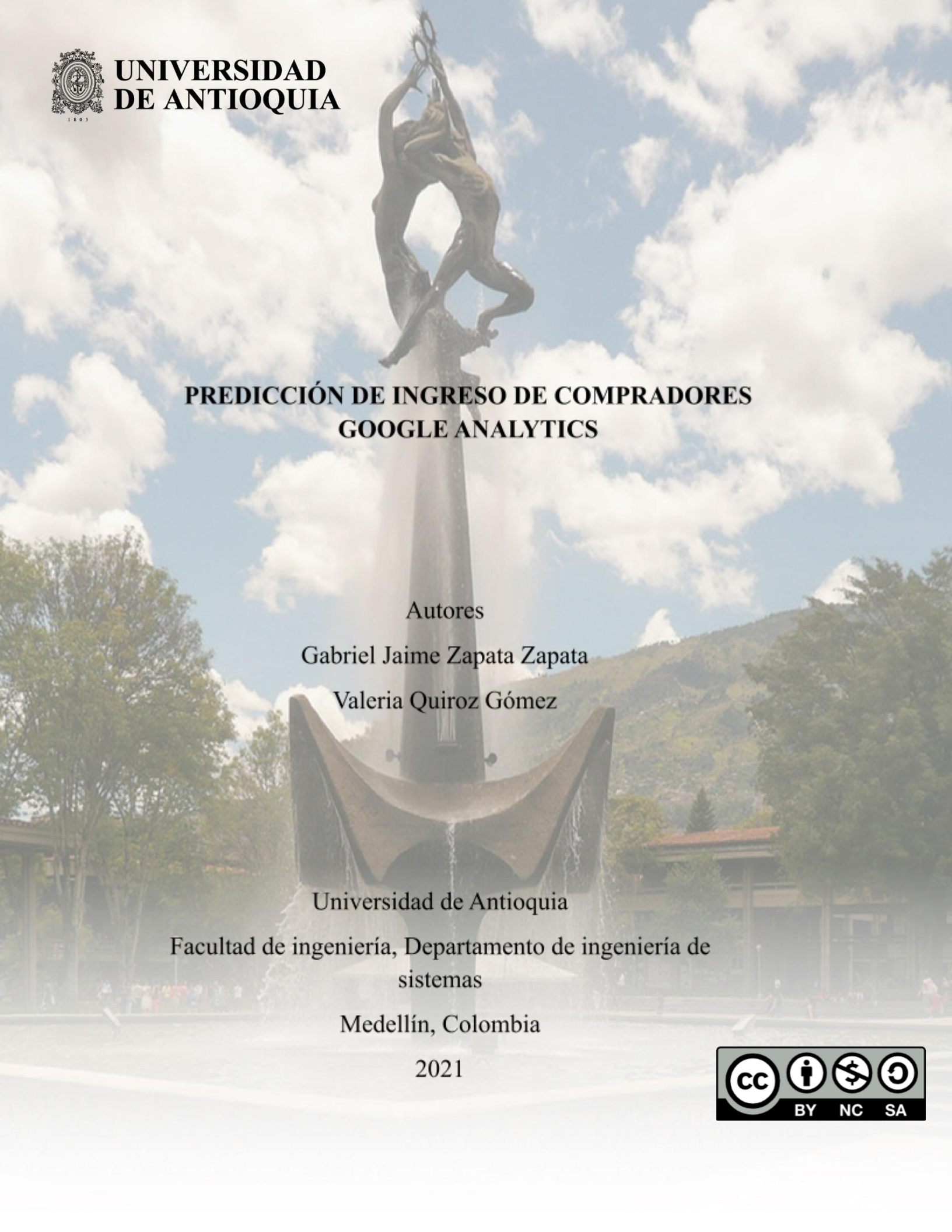


UNIVERSIDAD DE ANTIOQUIA

PREDICCIÓN DE INGRESO DE COMPRADORES GOOGLE ANALYTICS

Autores

Gabriel Jaime Zapata Zapata

Valeria Quiroz Gómez

Universidad de Antioquia

Facultad de ingeniería, Departamento de ingeniería de sistemas

Medellín, Colombia

2021

Predicción de ingreso de compradores Google analytics

Gabriel Jaime Zapata Zapata
Valeria Quiroz Gomez

Tesis o trabajo de investigación presentado como requisito parcial para optar al título de:
Especialista en analítica y ciencia de datos

Asesor:
Julian David Arias Londoño - Doctor en ingeniería

Línea de Investigación:
Diseño de Sistemas basados en Machine Learning

Universidad de Antioquia
Facultad de ingeniería, Departamento de ingeniería de sistemas.
Medellín, Colombia
2021.

1. Descripción del problema

El "Principio de Pareto", o más comúnmente "la regla 80/20" es una relación que describe la causalidad y los resultados ante un suceso. Afirma que aproximadamente el 80% de la producción es un resultado de aproximadamente el 20% de la entrada. Observado por primera vez en 1906 por un economista italiano Vilfredo Pareto *(*Kiremire, 2011).

El mundo digital ha obligado a repensar las ventas y el marketing que se realiza por internet, con el fin de ser eficientes en dichas ventas, por tal motivo se presentó por la subsidiaria Kaggle de Google LLC, el 13 de septiembre del 2018 la competición que hace el uso del principio de Pareto, el cual entregaba un premio de 45 mil USD a quienes presentaran una solución al problema de regresión planteado (Kaggle, 2018). La competición fue superada por el usuario Konstantin Nikolaev quien presentó su solución basada en los algoritmos de árbol de decisiones (Lightgbm. s. f.) En enero del 2020, (Vahid Azizi, 2020), se publican nuevos experimentos tratando de evaluar otros modelos basados en los algoritmos de árboles de decisión incluyendo al usado en la competición, dichos experimentos arrojaron que el modelo más eficiente, es el presentado por el ganador de la competición del concurso de Kaggle. Es importante considerar que el estado del arte no revela que se hayan intentado realizar experimentos con redes neuronales.

Partiendo de la competición realizada por Kaggle (Kaggle, 2020), vamos a considerar información de Google Merchandise Store que es un sitio web de comercio electrónico que vende productos de la marca Google, también conocido como GStore.

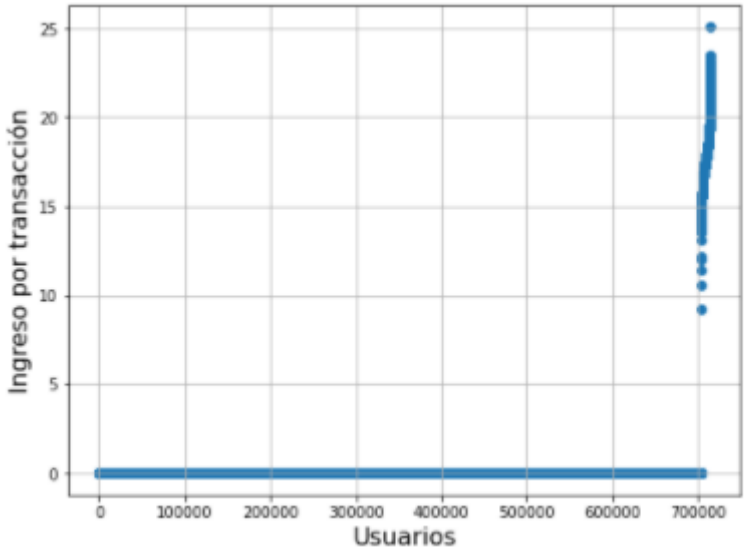
Por lo anterior, el objetivo de este trabajo de investigación es realizar un análisis que permita predecir el registro natural de la suma de todas las transacciones por usuario que ingresa a la tienda de Google.

2. Descripción del dataset

La información disponible para este problema está dada por un conjunto de datos de entrenamiento y un conjunto de test. Cada fila del conjunto de datos es una visita a la tienda por parte de un usuario, es importante mencionar que un registro de visita no necesariamente implica una compra por parte del usuario. La columna TransactionRevenue contiene la información de ingresos de los usuarios que pretendemos predecir, cabe destacar que esta variable existe solo para los datos de entrenamiento.

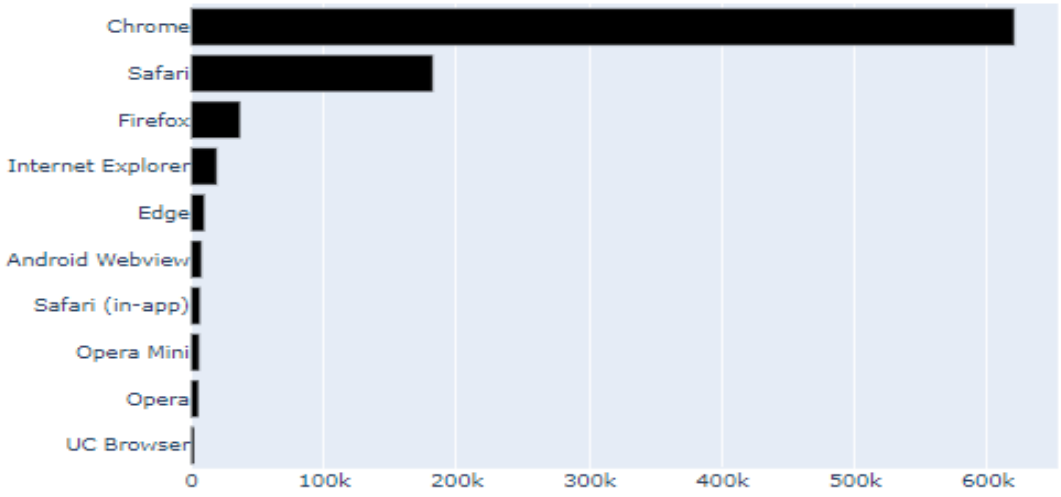
En el conjunto de datos se tienen 55 variables, con 903.653 registros, dentro de las cuales se encuentran 19 variables constantes que no le agregan información relevante al aprendizaje, en los registros se evidencia que 9.996 datos son diferentes de cero, es decir, que generan una compra en la tienda. Dentro de las variables otorgadas en el dataset se considera información tal como: la fecha de visita a la tienda, las marcas de tiempo de la sesión, el identificador único por usuario, el canal a través del cual el usuario ingresa a la tienda, el estado civil del usuario, dispositivo por el cual accede a la tienda, información geográfica del usuario, dominios de red, número de visitas del usuario a la tienda y la variable de salida. Se especifica que las variables dispositivo, posición geográfica, origen del tráfico y totales están en formatos json.

Para cada usuario, debemos predecir la suma de todas las transacciones realizadas. La gráfica 1 representa las transacciones por usuario, se puede evidenciar que las transacciones se concentran en un grupo de personas bastante reducido y que la mayoría de los usuarios que ingresan a la tienda, no generan compras en la misma. Evidenciando el principio de pareto mencionado anteriormente.

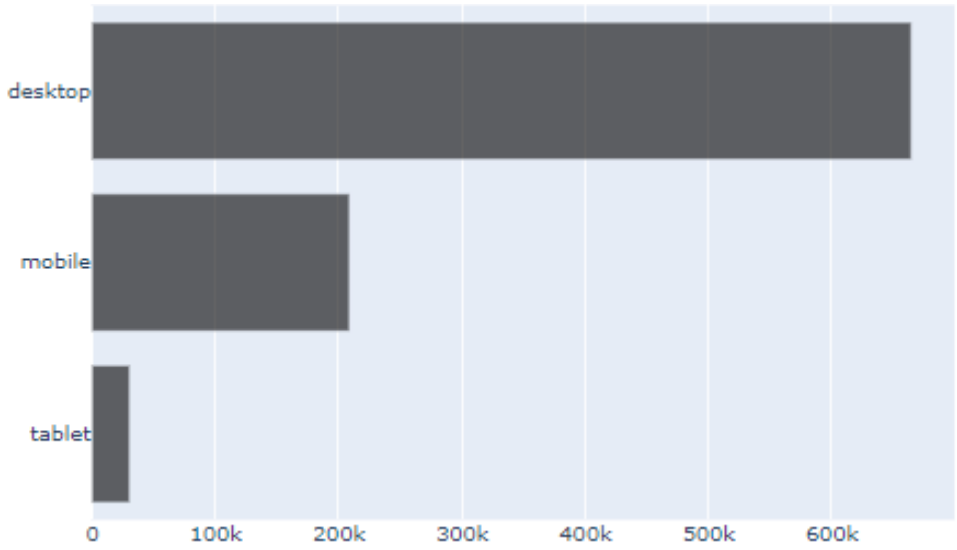


Gráfica 1: Total de transacciones por usuario.

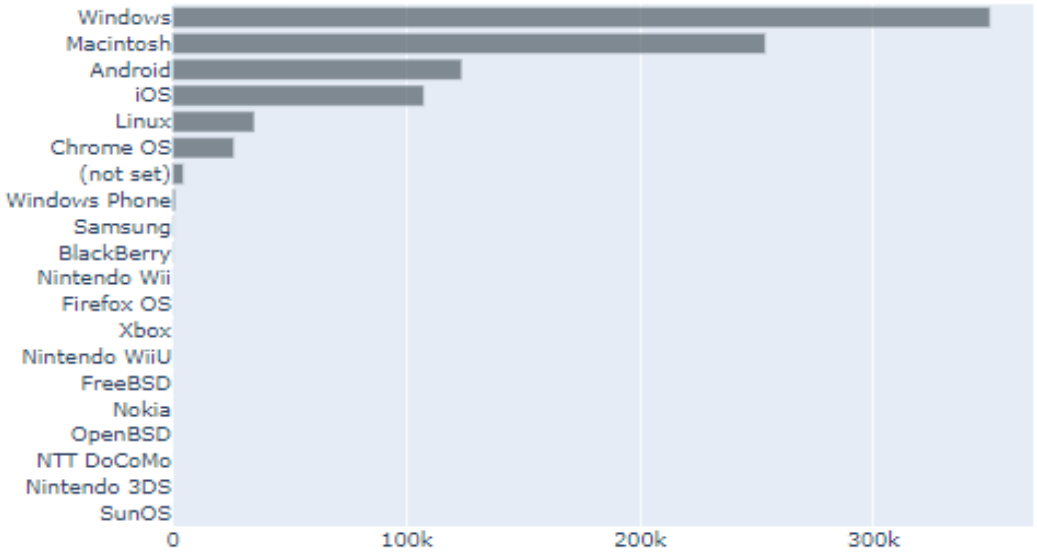
Se realiza un análisis exploratorio con el fin de visualizar el comportamiento de las variables relevantes frente a los compradores de la tienda, en el gráfico 2 se exploran el navegador, el tipo de dispositivo y sistema operativo frente a los compradores.



Gráfica 2: Navegadores



Gráfica 3:Dispositivos

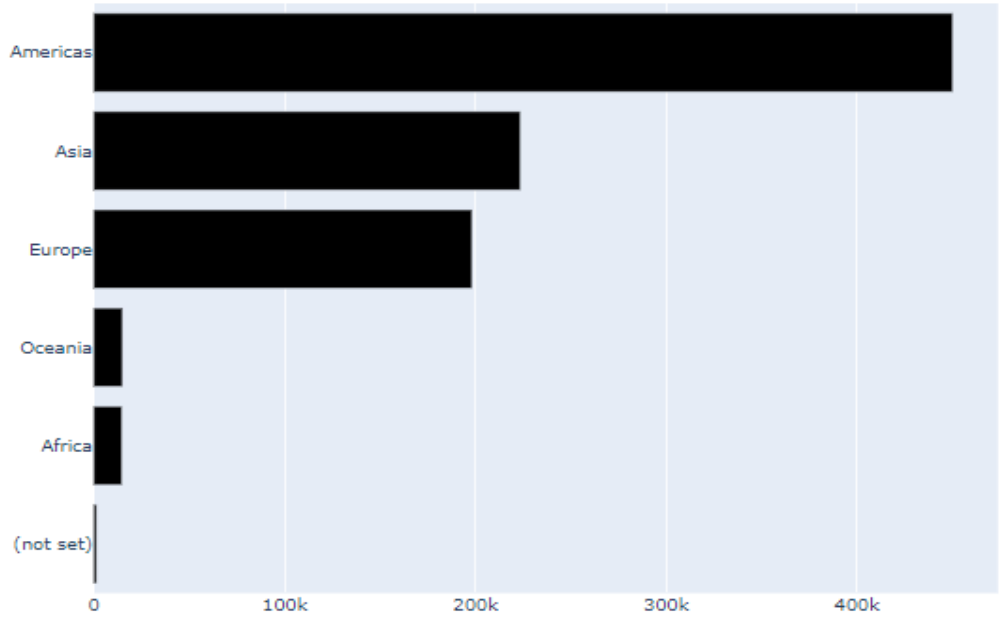


Gráfica 4: Sistemas operativos

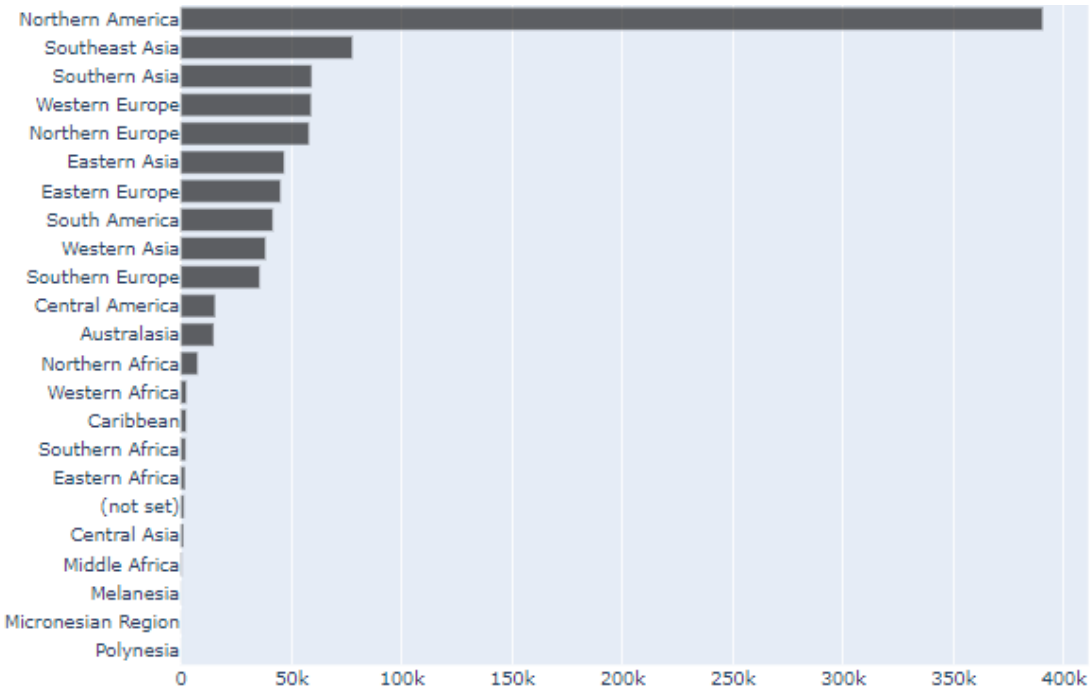
De las gráficas 2, 3 y 4 se infiere lo siguiente:

- La mayoría de las compras se realizan desde un navegador Chrome.
- Los tipos de dispositivos más usados para hacer compras son los dispositivos de escritorio.
- El sistema operativo más utilizado es Windows.

Continuando con la exploración de las variables de entrada, se evalúa la ubicación geográfica de los compradores; en la gráfica 5 es posible inferir que los mayores compradores se localizan en América, especialmente en el norte según la gráfica 6.



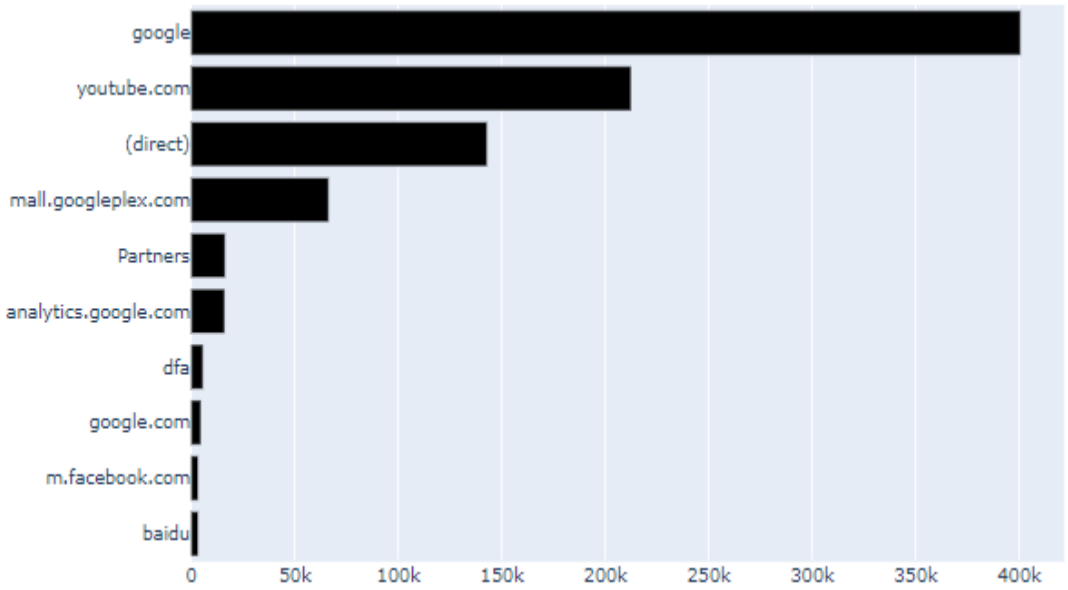
Gráfica 5: continente



Gráfica 6: sub continente

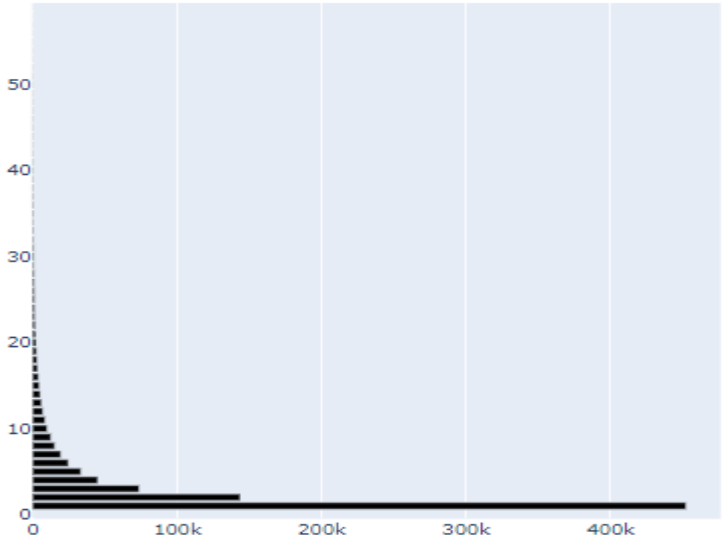
Al indagar de dónde proviene posiblemente la mayor referenciación de los usuarios antes de una compra, se decide explorar la fuente del tráfico antes de la compra e identificar cómo se comporta la variable, esto puede

observarse en el gráfico 7, y se puede inferir que el buscador Google es una buena herramienta de referenciación para la tienda para generar compras.



Gráfica 7: Fuentes de tráfico.

Finalmente, en la gráfica 8 se evidencia que los usuarios realizan búsquedas extensas de los productos en diferentes sitios web, días antes de realizar la compra en la tienda.



Gráfica 8: Perfil de compradores

2.1 Tratamiento del dataset

Al explorar las variables contenidas en el dataset, se evidencian columnas constantes las cuales se eliminan, además, se detectan casillas sin valores (NaN), las cuales se imputaron con cero, adicionalmente se elimina la columna "sessionId" ya que son valores únicos que no aportan valor al entrenamiento del modelo. Con lo anterior nuestro dataset se reduce a 33 variables y la salida, además las variables categóricas son convertidas en variables numéricas.

3. Objetivos

Predecir el logaritmo natural de la suma de todas las transacciones por usuario en la tienda de compras online de Google (Gstore), entre el 01 de agosto de 2016 al 15 de octubre de 2018.

Aplicar conceptos para el diseño de una solución de predicción basada en analítica predictiva, incluyendo fases de análisis exploratorio, entrenamiento

de modelos, validación, procesamiento, elección de mejores hiperparametros, que permitan llevar a cabo la predicción de los ingresos de los usuarios en la tienda.

4. Marco Teórico

El análisis predictivo es uno de los usos más frecuentes del Machine Learning (Spain, 2020), dado que es de los más útiles para las necesidades de la mayoría de las compañías. Este tipo de análisis utiliza modelos matemáticos para pronosticar eventos o resultados futuros. Mediante un proceso iterativo, se desarrolla el modelo predictivo mediante un conjunto de datos de entrenamiento y después se valida y se prueba, para determinar su precisión con el fin de realizar los mejores pronósticos.

En Machine learning existen diversos algoritmos que permiten realizar predicciones sobre datos e información y cada uno de esos algoritmos debe de ser seleccionado según las condiciones de los datos obtenidos, como se detalló anteriormente, para este caso en particular se realiza una predicción de compras de los usuarios, por tanto, se utilizan modelos de regresión dado que estos permiten representar la relación entre las diferentes variables, y a partir de allí se hallan relaciones matemáticas que facilitan entender el modo en que la variable dependiente es afectada por cambios en las demás variables consideradas en el dataset otorgado por Google.

Para abordar el problema planteado, se seleccionaron modelos capaces de brindar soluciones de regresiones no lineales. En la competencia original se propuso como solución el modelo Light GBM (Lightgbm,s. f.) fundamentado en boosting, es decir que ajusta secuencialmente múltiples modelos sencillos, llamados weak learners, de forma que cada modelo aprende de los errores del anterior. Para las iteraciones de los experimentos que más adelante se van a proponer en la monografía, se hace uso de los modelos

de Machine learning de Gradient Boosting Regressor (Scikit Learn, s. f.) y Random Forest Regressor (Scikit-Learn, s. f.). El primer modelo es similar al usado en el notebook ganador (kaggle, 2019), pero no es tan eficiente en temas de tiempos de ejecución, el segundo modelo está basado en bagging, es decir que se ajustan múltiples modelos (Amat, 2020), cada uno con un subconjunto distinto de los datos de entrenamiento. Para predecir, todos los modelos que forman el agregado participan aportando su predicción. En los dos modelos se colocan a variar los parámetros de ramas y profundidad máxima.

También como cuota experimental se hace uso de redes neuronales, con un modelo secuencial, compuesto por 4 capas densas y con una salida lineal, con el fin de no perder la propuesta de los experimentos, se trabajó con el wrappers de Keras que hace uso de de la librería de scikit learn.

Las métricas utilizadas para evaluar los experimentos son el error cuadrático medio, que calcula el valor medio de la diferencia al cuadrado entre el valor real y el predicho para todos los puntos de datos de salida, la raíz cuadrada de este nos servirá como métrica de error - RMSE, al igual que el error absoluto medio - MAE que es el valor medio de la diferencia absoluta entre el valor real y el valor predicho.

Al ser un ejercicio publicado en Kaggle.com hay varios competidores que lo han abordado, especialmente en el lenguaje de programación R y Java, además la competencia ya finalizó, por tanto se puede consultar los resultados obtenidos por parte de los concursantes en la página oficial del concurso. Al consultar el procedimiento de los otros concursantes, se puede concluir que la predicción es realizada por el modelo LightGBM.

Marco experimental

Para lograr los objetivos propuestos se analiza un conjunto de datos otorgado por Kaggle en un periodo de tiempo del 01 de agosto de 2016 al 15 de octubre de 2018 y se busca predecir los ingresos generados entre el 01 de diciembre de 2018 y el 31 de enero de 2019. Con el fin de brindar un mejor uso a los presupuestos de marketing que permita el enfoque en sus campañas y posteriormente mejorar las ventas en la tienda.

Según la fuente de datos presentada, en este trabajo se pretende mediante dos experimentos evaluar el desempeño del entrenamiento de los modelos, esto se realiza mediante la agrupación de los datos de entrenamiento y validación; es importante resaltar que los datos de la competencia “test.csv” no cuentan con la columna “transactionRevenue”, por esto, se decide descartar estos datos que fueron suministrados en la competencia. Esta variable es el objetivo a predecir en la competición, donde, para cada usuario, tenemos que averiguar el total de todas las transacciones durante el período de prueba, tal como se expresa en la ecuación 1.

$$y_{usuario} = \sum_{i=1}^{n} transacción_{usuario_i}$$

Ecuación 1

Donde i es el subíndice que identifica a cada usuario, $transacción_{usuario}$ representa las transacciones realizadas por cada usuario, $y_{usuario}$ es el total de dinero usado por cada usuario para sus compras y está dado en dólares.

Como la salida puede presentar sumas muy elevadas, se solicita en la competición calcular el logaritmo natural. De igual manera pueden existir valores iguales a cero en la salida y el logaritmo natural no está definido en ese valor, por lo que debe sumarse una unidad, tal y como lo ilustra en la ecuación 2.

$$target_{usuario} = ln\left(y_{usuario} + 1\right)$$

Ecuación 2

Con el fin de realizar una buena predicción de los ingresos de los usuarios en la tienda, se plantean dos experimentos, el primero consiste en evaluar todos los registros del dataset, cuyo propósito es medir la respuesta del aprendizaje frente al comportamiento de los usuarios según la secuencia temporal en la cual se presentaron los registros, y lograr predecir futuros comportamientos. La estrategia planteada es similar a la usada en series de tiempo pero con segmentación no acomulada, es decir, se seleccionaron tres grupos temporales independientes uno del otro, de igual manera se separó un cuarto grupo de prueba para la predicción. Es importante aclarar que no existe una librería capaz de realizar este proceso tal y como se describe, asi que se diseña una estrategia manual para la ejecución de este experimento.

El segundo experimento está enfocado en generar un aprendizaje dado el comportamiento de cada usuario, el objetivo es evaluar la respuesta de aprendizaje dada según la huella única digital que genera un usuario en la plataforma. En este se decide tomar la variable “fullVisitorId” , dado que es el registro digital que permite reconocer a cada usuario. En este caso se hace la división con una función llamada *GroupKFold* (Scikit-Learn,s. f.) que permite agrupar los datos según los requerimientos establecidos previamente para este caso.

Para la búsqueda de los mejores hiperparametros del modelo con los algoritmos de árboles de decisión, se opta por variar el número de ramas

entre 20 - 100 con un paso de 10 , y la profundidad entre 3 - 21 con un paso de 3. Mientras que en el aprendizaje con redes neuronales se selecciona como hiperparametros el número de neuronas que varían entre 64 - 1024, con un paso de múltiplos pares, y el parámetro de regularización entre 0,1 - 0,5, con un paso de 0,1.

Resultados

Se presentan los resultados del primer experimento en la tabla 1, la cual está compuesta en la primera columna por el tipo de modelo usado, en la segunda columna contiene el error de entrenamiento subdividido en la evaluación del error realizada por usuario y por registro, la tercera columna ilustra el error de validación subdividido igual a la columna anterior y la cuarta columna contiene el error de prueba, donde se realiza la evaluación con la métrica RMSE subdividida en la evaluación por usuario y por registro, al igual que la evaluación de la métrica MAE que también se subdivide en dos evaluaciones dadas por usuario y por registro.

Es importante resaltar que existen dos filas que dividen la tabla, una es scoring MAE y la otra es scoring RMSE, que fueron importantes en el experimento para determinar los menores errores en la validación, de ahí se encontró que para el modelo Gradient Boosting los mejores hiperparametros para el scoring MAE, fueron con un número de ramas de 80 y con una profundidad máxima de 3, y para el scoring RMSE los mejores hiperparametros fueron 90 ramas, con una profundidad máxima de 3. Para el modelo Random Forest sucede que los mejores hiperparametros para el scoring MAE, fueron con un número de ramas de 20 y con una profundidad máxima de 6, y para el scoring RMSE los mejores hiperparametros fueron 60 ramas, con una profundidad máxima de 6. Por último, se tiene la evaluación de redes neuronales en donde los mejores hiperparametros

encontrados para el scoring MAE, fueron 64 neuronas con un parámetro de regularización de 0,1, y para el scoring RMSE los mejores hiperparametros fueron 128 neuronas con un parámetro de regularización de 0,3.

MODELO	ERROR ENTRENAMIENTO		ERROR VALIDACIÓN		ERROR PRUEBA			
	POR USUARIO	POR REGISTRO	POR USUARIO	POR REGISTRO	RMSE POR USUARIO	RMSE POR REGISTRO	MAE POR USUARIO	MAE POR REGISTRO
SCORING MAE								
GRADIENT BOOSTING	0,3298	0,314	0,3266	0,3026	2,1163	1,7768	0,4235	0,3778
RANDOM FOREST	0,3630	0,3395	0,3489	0,317	2,2359	1,8672	0,4055	0,3509
REDES NEURONALES	0,5213	0,4399	0,5212	0,4362	2,7338	2,1771	0,6075	0,4889
SCORING RMSE								
GRADIENT BOOSTING	1,8998	1,6349	1,8401	1,6098	2,1209	1,777	0,4283	0,3823
RANDOM FOREST	1,9425	1,9033	1,9428	1,6534	2,1337	1,8049	0,4472	0,4014
REDES NEURONALES	2,4037	1,9724	2,3614	1,9497	2,7358	2,1773	0,5969	0,4801

Tabla 1: Resultados experimento 1

La tabla 2 ilustra los resultados del segundo experimento, esta tabla representa la misma estructura que la tabla 1 pero con la diferencia que las columnas de error de entrenamiento por usuario y error de validación por usuario no existen, debido a que se realizó la búsqueda de los mejores hiperparametros con la función GridsearchCV (Scikit-Learn, s. f.).

MODELO	ERROR ENTRENAMIENTO POR REGISTRO	ERROR VALIDACIÓN POR REGISTRO	ERROR PRUEBA			
			RMSE POR USUARIO	RMSE POR REGISTRO	MAE POR USUARIO	MAE POR REGISTRO
SCORING MAE						
GRADIENT BOOSTING	0,2845	0,3046	2,2654	1,6532	0,3278	0,3091
RANDOM FOREST	0,24	0,3041	2,2239	1,6737	0,3285	0,311
REDES NEURONALES	0,2272	0,2272	3,2197	2,0365	0,2939	0,2324
SCORING RMSE						
GRADIENT BOOSTING	1,5156	1,6248	2,2577	1,6526	0,3274	0,3089
RANDOM FOREST	1,6675	1,6446	2,2484	1,6789	0,3325	0,3158
REDES NEURONALES	2,0055	2,0051	3,2081	2,0346	0,3142	0,2494

Tabla 2: Resultados experimento 2

Al analizar los resultados de los dos experimentos, se encuentra que el RMSE por usuario presentado en los modelos de gradient boosting y random forest, en el primer experimento son menores al del experimento 2, lo que demuestra que a los modelos de gradient boosting y random forest

del experimento 2 les cuesta más identificar un usuario nuevo que ingrese a la tienda. Por lo anterior, si se aumenta el número de registros para el primer experimento, la predicción del modelo tendrá un menor error.

En los dos experimentos las redes neuronales obtuvieron errores mayores en el entrenamiento, la validación y las pruebas, a los obtenidos en los modelos Gradient Boosting y Random Forest. Sin embargo, el RMSE por usuario es menor en el primer experimento que en el segundo.

Los resultados obtenidos en las tablas 1 y 2 para los errores de entrenamiento y validación, por registro en los dos experimentos, no se alejan mucho en los valores obtenidos en cada modelo en los diferentes scoring evaluados, exceptuando al modelo de redes neuronales del experimento 2 de scoring MAE, que obtuvo errores menores al del mismo scoring del experimento 1.

El método usado en redes neuronales para construir los modelos está basado en una regresión lineal, lo puede explicar porque los errores hallados son mayores a los registrados por los modelos entrenados basados en árboles de decisión, pues estos son algoritmos de aprendizaje más robustos.

En los modelos de gradient boosting y redes neuronales del experimento 1, fue posible extraer la representación de la función de pérdida, tanto para entrenamiento como para las pruebas. Para el experimento 2 en estos mismos modelos no fue posible realizar esta representación por la forma en que se encontraron los modelos. Los resultados los podemos ver en las gráficas 9 y 10 de los anexos. Para el modelo donde se usó random forest no fue posible hallar esta representación debido a cómo funciona el algoritmo de aprendizaje, descrito en el marco teórico.

5. Conclusiones

Se demostró que el principio de Pareto se cumple según el diagrama de dispersión de los datos de todos los usuarios que ingresan a la tienda, evidenciando que las transacciones se concentran en un grupo de personas bastante reducido y que la mayoría de los usuarios que ingresan a la tienda, no generan compras en la misma.

El análisis de características sirve para identificar y perfilar a los compradores de la tienda para generar posibles campañas de marketing. Además, de aportar a la selección de los modelos que se construyeron en los experimentos presentados.

En los dos experimentos evaluados, con los tres diferentes algoritmos de aprendizaje, se logra predecir el logaritmo natural de las transacciones por usuario y se obtiene un menor error en los resultados del experimento número 1, con la estrategia gradient boosting. También se destaca que en los experimentos que usan algoritmos de árboles de decisión se tuvo un costo computacional muy alto y para obtener los resultados de los experimentos se usaron tecnologías ofrecidas por colab de google tales como GPU, TPU, almacenamiento y memoria RAM, para el entrenamiento de estos modelos (Colab, 2021).

Se logra aplicar los conocimientos adquiridos durante la especialización para la elaboración de los dos experimentos propuestos, en el lenguaje interpretado python. Además, se logra realizar el análisis exploratorio de la base de datos identificando patrones útiles para la identificación de compradores. Se generaron iteraciones en donde los experimentos exigen la correcta interpretación de los modelos y sus validaciones, según la variación de los hiperparametros.

6. Referencias bibliográficas

Amat, J. A. R. (2020, octubre). *Random Forest python*. Random Forest con Python. https://www.cienciadedatos.net/documentos/py08_random_forest_python.html

Colab , Google (2021, Junio) https://colab.research.google.com/notebooks/welcome.ipynb?hl=es

Google Analytics Customer Revenue Prediction | *Kaggle*. (s. f.). Kaggle. Recuperado 2020, de https://www.kaggle.com/c/ga-customer-revenue-prediction/rules

Kaggle. (2019, 25 febrero). *Winning_solution*. Kaggle. https://www.kaggle.com/kostoglot/winning-solution

Kaggle. (2018, septiembre). *Google Analytics Customer Revenue Prediction* | *Kaggle*. https://www.kaggle.com/c/ga-customer-revenue-prediction/overview

Kiremire, A. R. K. (2011, 19 octubre).The application of the pareto principle in software enginieering pdf. http://www2.latech.edu/~box/ase/papers2011/Ankunda_termpaper.PDF

*G*Store. (2018, 5 noviembre). *Google Analytics Customer Revenue Prediction*. Kaggle. https://www.kaggle.com/shrmis/google-analytics-customer-revenue-prediction

sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.24.2 documentation. (s. f.). Sciki Learn. Recuperado 2021, de https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

sklearn.ensemble.RandomForestRegressor — scikit-learn 0.24.2 documentation. (s. f.). Scikit-Learn. Recuperado 2021, de https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

sklearn.model_selection.GridSearchCV — scikit-learn 0.24.2 documentation. (s. f.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

sklearn.model_selection.GroupKFold — scikit-learn 0.24.2 documentation. (s. f.). Scikit-Learn. Recuperado 2021, de https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html

Spain, K. (2020, 16 septiembre). *Las 11 técnicas más utilizadas en el modelado de análisis predictivos*. Keyrus Spain Blog. https://keyrusspainblog.com/2020/05/05/las-11-tecnicas-mas-utilizadas-en-el-modelado-de-analisis-predictivos/

Vahid Azizi, V. A. (2020). Machine Learning Methods for Revenue Prediction in Google Merchandise Store. *Machine Learning Methods for Revenue Prediction in Google Merchandise Store*. Published. https://www.researchgate.net/publication/337545834_Machine_Learning_Methods_for_Revenue_Prediction_in_Google_Merchandise_Store

Welcome to LightGBM's documentation! — LightGBM 3.2.1.99 documentation. (s. f.). Lightgbm. https://lightgbm.readthedocs.io/en/latest/

7. Agradecimientos

Queremos agradecer a todos los profesores de la especialización por la entrega en transmitir sus conocimientos, especialmente al maestro Julian David Arias Londoño, por su constante seguimiento y apoyo en las iteraciones de la monografía, siempre dispuesto aclarar dudas y generar en nosotros la curiosidad para cada vez aprender más en este campo.

8. Anexos

El experimento 1 de Gradient Boosting permite extraer el comportamiento de la función de pérdida de cada uno de los modelos con los mejores hiperparametros:

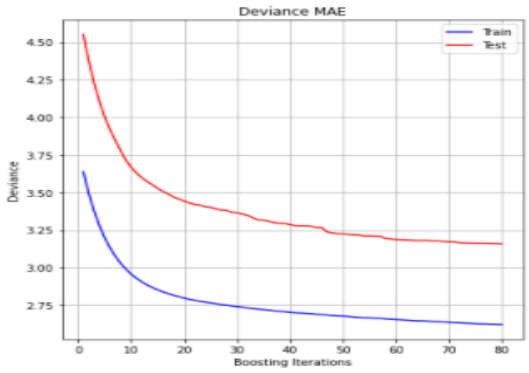


Gráfico 9. Desviación en MAE por usuario - mejor resultado

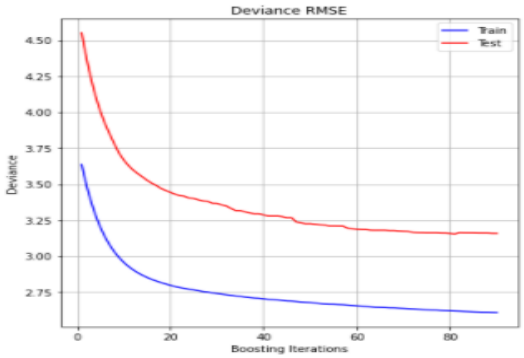
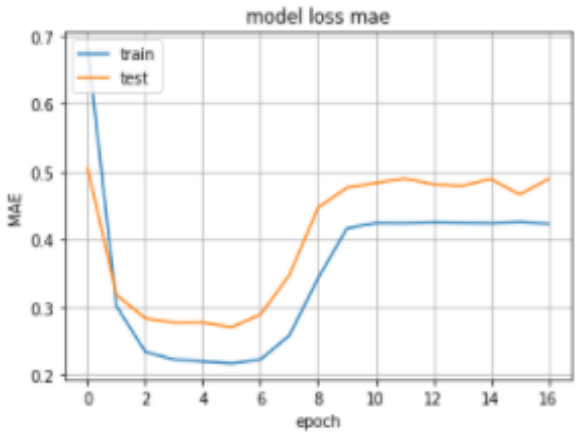
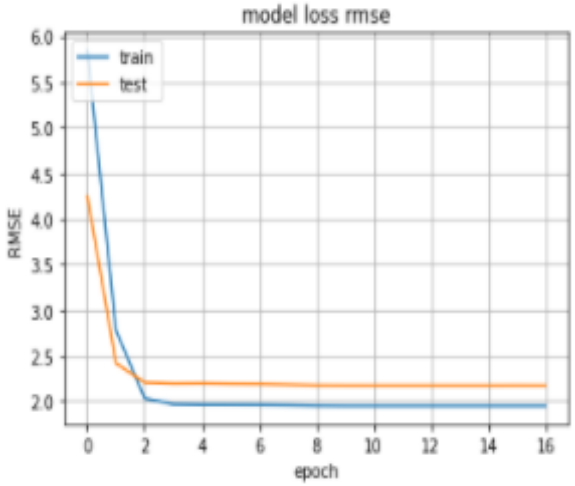


Gráfico 10. Desviación en RMSE por usuario - mejor resultado

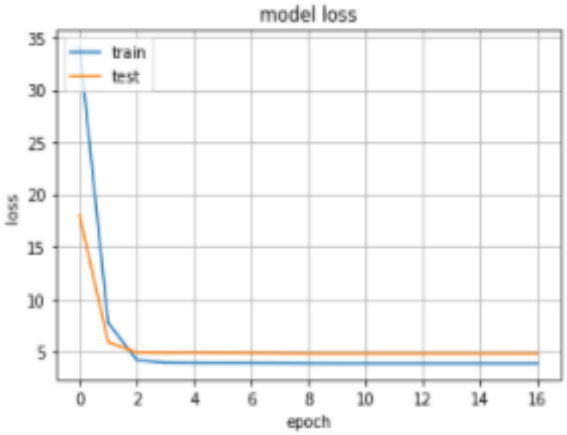
El experimento 1 de redes neuronales permitió extraer según las épocas el comportamiento de las métricas, de igual manera para la función de pérdida.



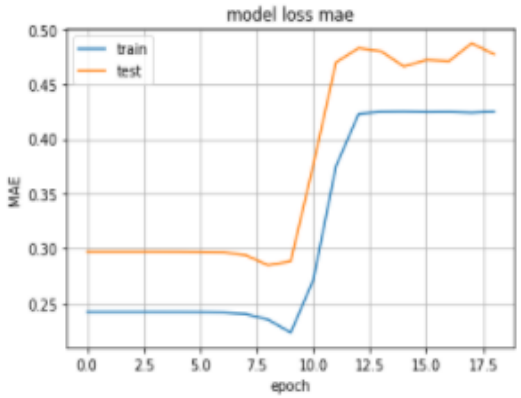
Comportamiento métricas experimento 1 - bajo los mejores hiperparametros MAE



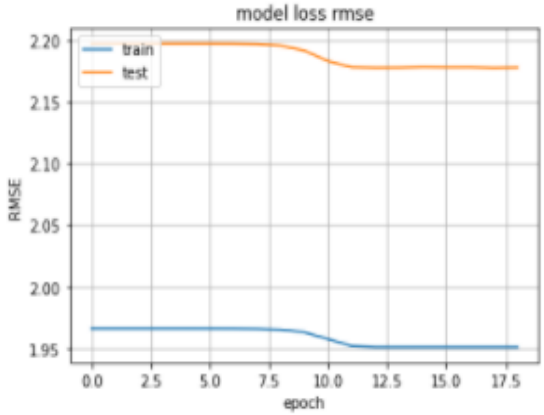
Comportamiento métricas experimento 1 - bajo los mejores hiperparametros MAE



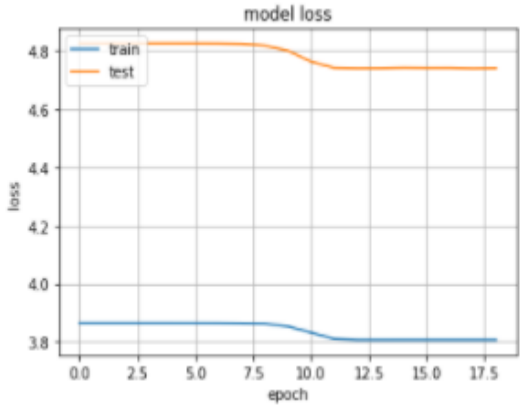
Comportamiento métricas experimento 1 - bajo los mejores hiperparametros MAE



Comportamiento métricas experimento 1 bajo los mejores hiperparametros RMSE



Comportamiento métricas experimento 1 bajo los mejores hiperparametros RMSE



Comportamiento métricas experimento 1 bajo los mejores hiperparametros RMSE

En los experimentos realizados en redes neuronales se obtuvo la siguiente estructura de los modelos creados según los mejores hiperparametros:

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 64)	2112
dense_9 (Dense)	(None, 32)	2080
dense_10 (Dense)	(None, 8)	264
dropout_2 (Dropout)	(None, 8)	0
dense_11 (Dense)	(None, 1)	9

Total params: 4,465
Trainable params: 4,465
Non-trainable params: 0

Mejor modelo métrica MAE experimento 1

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 128)	4224
dense_13 (Dense)	(None, 64)	8256
dense_14 (Dense)	(None, 16)	1040
dropout_3 (Dropout)	(None, 16)	0
dense_15 (Dense)	(None, 1)	17

Total params: 13,537
Trainable params: 13,537
Non-trainable params: 0

Mejor modelo métrica RMSE experimento 1

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 512)	16896
dense_13 (Dense)	(None, 256)	131328
dense_14 (Dense)	(None, 64)	16448
dropout_3 (Dropout)	(None, 64)	0
dense_15 (Dense)	(None, 1)	65

Total params: 164,737
Trainable params: 164,737
Non-trainable params: 0

Mejor modelo métrica MAE experimento 2

Layer (type)	Output Shape	Param #
dense_16 (Dense)	(None, 256)	8448
dense_17 (Dense)	(None, 128)	32896
dense_18 (Dense)	(None, 32)	4128
dropout_4 (Dropout)	(None, 32)	0
dense_19 (Dense)	(None, 1)	33

Total params: 45,505
Trainable params: 45,505
Non-trainable params: 0

Mejor modelo métrica RMSE experimento 2