

**Trabajo Final de Master**

Predicción de Ventas de Comestibles Corporación Favorita

Autor: **Gabriel Kreplak**

Plan: Master en Inteligencia de Negocio y Big Data

Área: Análisis de Datos

Consultora: **Dra. Laia Subirats Maté**

Profesoras responsables de la asignatura: **Dra. Teresa Sancho Vinuesa y Dra. María Pujol Jover**

23 de enero de 2018

  
Esta obra está sujeta a una licencia de Reconocimiento-NoCommercial-SinObraDerivada [3.0 España de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Copyright © 2018 Gabriel Kreplak.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright© 2018 Gabriel Kreplak

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

**FICHA DEL TRABAJO FINAL**

|  |  |
| --- | --- |
| **Título del trabajo:** | Predicción de Ventas de Comestibles Corporación Favorita. |
| **Nombre del autor:** | Gabriel Kreplak |
| **Nombre del consultor/a:** | Dra. Laia Subirats Maté |
| **Nombre del PRA:** | Dra. Teresa Sancho Vinuesa y Dra. María Pujol Jover |
| **Fecha de entrega (mm/aaaa):** | Enero 2018 |
| **Titulación::** | Máster Inteligencia de Negocio y Big Data |
| **Área del Trabajo Final:** | Análisis de Datos |
| **Idioma del trabajo:** | Español – Inglés |
| **Palabras clave** | análisis de datos, ventas, light-GBM, alimentación, predicción, regresión |
| **Resumen del Trabajo (máximo 250 palabras):** Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo. | |
| El trabajo escogido tiene por finalidad conseguir una puntuación relevante en la competición de Kaggle: **Corporación Favorita Grocery Sales Forecasting**. El objetivo de la competición es construir un modelo predictivo para pronosticar ventas futuras.  Los métodos de planificación de ventas de esta compañía actualmente están poco sustentados en datos y como consecuencia no están automatizados. Por este motivo, Corporación Favorita propone a la comunidad Kaggle el desarrollo te técnicas de Machine Learning para conseguir adecuar en lo posible la logística y oferta de productos a la demanda de éstos donde son requeridos.  Como se detallará en el apartado de descripción de los ficheros de datos que se ofrecen para la predicción, éstos incluyen información para entrenamiento y test de los algoritmos supervisados y otros ficheros de datos para poder poner en contexto los datos mencionados. Los ficheros aportados son: train.csv, test.csv, sample\_submission.csv, stores.csv, ítems.csv, transactions.csv, oil.csv y holidays\_events.csv . El trabajo lo estructuraré en 2 partes principales:   1. Análisis Exploratorio de Datos, que incluirá que estudiará detalladamente la información de ventas: en general, por tipo de tienda, geolocalización , ventas por estado, por ciudad, ventas en el tiempo, correlación ventas – precio del petróleo, análisis de los productos y sus familias, transacciones, etc. 2. Estudio Predictivo para lo que utilizaré los algoritmos con los que obtenga un resultado que me permitan obtener una buena clasificación en la tabla y recibir también comentarios positivos. | |
| Abstract (in English, 250 words or less): | |
| The chosen final grade project theme is to earn a relevant score in the Kaggle’s competition: Corporación Favorita Grocery Sales Forecasting by building a state of the art predictive model aimed to forecast future sales.  Corporación Favorita has challenged the Kaggle community to build a model that more accurately forecasts product sales. They currently rely on subjective forecasting methods with very little data to back them up and very little automation to execute plans. They’re excited to see how machine learning could better ensure they please customers by having just enough of the right products at the right time.  As it will be detailed below, in section “Description of the Input Data”, training and testing data files are supplied in order to develop predictive supervised algorithms as well as other data file to be able to put the aforementioned data in context. The files provided are: train.csv, test.csv, simple\_submission, stores.csv, items.csv, transactions.csv, oil.csv and holidays\_events.csv.  The work will be structured in 2 main parts:  a) Exploratory Data Analysis, which will include detailed study of sales information: in general, by type of store, geolocation, sales by state, by city, sales in time, sales correlation - price of oil, analysis of products and their families, transactions, etc.  b) Predictive study for which I will use the state of the art algorithms in order to obtain a relevant classification in the score table and to receive positive comments as well. | |

Índice

1 Introducción 5

1.1 Contexto y justificación del trabajo 5

1.2 Objetivos del trabajo 5

1.3 Enfoque y método seguido 6

1.4 Planificación del trabajo 6

1.5 Descripción de los demás capítulos de la memoria 8

2 Estado del Arte 9

3 Competición Corporación Favorita 10

3.1 Reglas mas relevantes 10

3.2 Descripción de los Datos de Entrada 10

3.3 Descripción de los Datos de Salida 12

3.4 Evaluación 12

4 Análisis Exploratorio de Datos 14

4.1 Estructura de los archivos de datos de entrada 14

4.2 Archivo train.csv 14

4.3 Archivo transactions.csv 15

4.4 Archivo holiday\_events.csv 15

4.5 Archivo oil.csv 16

4.6 Archivo items.csv 16

4.7 Archivo stores.csv 17

4.8 Archivo holiday\_events.csv 20

4.9 Archivo transactions.csv 20

4.10 Archivo test.csv 21

5 Estudio Predictivo 23

5.1 Algoritmo Light Gradient Boosting Machine 23

5.2 Ajuste de parámetros LGBM 24

5.3 Descripción del programa 25

6 Conclusiones 32

7 Glosario 34

8 Bibliografía 35

9 Anexos 36

Lista de Figuras

Figura 2: Precios de petróleo 16

Figura 3: Número de productos por cada categoría 17

Figura 4 : Treemap de tiendas/ciudad 18

Figura 5: Segmentación de las tiendas 19

Figura 6: Clasificación de tiendas 19

Figura 7: Distribución de transacciones por día 21

Figura 8: Crecimiento del árbol por niveles. 23

Figura 9: Crecimiento LGBM por hojas 23

Lista de Tablas

Tabla 1. Planificación 7

Tabla 2: train.csv, 125 millones de líneas x 6 columnas 15

Tabla 3: transactions.csv, 83.500 líneas x 3 columnas 15

Tabla 4: holiday\_events.csv, 350 líneas x 6 columnas 15

Tabla 5: oil.csv, 1.218 x 2 col 16

Tabla 6: items.csv, 4.100 productos x 4 attr 16

Tabla 7: stores.csv, 54 tiendas x 5 atributos 18

Tabla 8: holiday\_events.csv, 350 líneas x 6 columnas 20

Tabla 9: transactions.csv, 83.500 líneas x 3 columnas 20

Tabla 10: test.csv 3.370.464 lineas x 5 columnas 22

Tabla 11: Estructura de X\_train 26

Tabla 12: Atributos dataset entrenamiento 28

Tabla 13: Estructura dataframe Y\_train 28

Tabla 14: Estructura de Predictores de Validación 29

Tabla 15: Estructura de Etiquetas de Validación 29

Tabla 16: Estructura de predictores de test 29

Tabla 17: Resumen de Ajustes 30

Lista de Ecuaciones

Ecuación 1: NWRMSLE 13

# Introducción

## Contexto y justificación del trabajo

Las tiendas de comestibles al detalle siempre se debaten entre la planificación de las ventas y las compras: predecir con exceso de stock puede conducir a problemas de sobre-almacenamiento de productos perecederos; predecir en defecto produce un rápido agotamiento de los productos mas populares con el consiguiente lucro cesante y consumidores insatisfechos.

El problema deviene mas complejo a medida que el minorista agrega nuevas ubicaciones con necesidades propias, productos especiales y gustos estacionales. Corporación Favorita, una gran cadena minorista de comestibles de Ecuador lo sabe muy bien ya que operan en cientos de supermercados con mas de 200.000 diferentes productos en sus estantes.

Corporación Favorita ha lanzado una competición para la comunidad Kaggle para la construcción de un modelo de predicción de las ventas de sus productos. Actualmente la planificación de ventas y compras no está respaldada en datos sino que en pronósticos subjetivos y ello supone poca automatización para ejecutar los planes.

Como patrocinadores de la competición, en Corporación Favorita están muy interesados en comprobar como mediante aprendizaje automático es posible mejorar la aceptación de los clientes simplemente teniendo los productos adecuados en el momento preciso.

## Objetivos del trabajo

Los principales objetivos que se pretende alcanzar son los siguientes:

* Adquirir una metodología profesional en la organización de un proyecto de análisis predictivo, incluyendo la utilización de plataformas standard de discusión y desarrollo como por ejemplo:
  + Kaggle
  + Github
  + Google drive
  + Pycharm
  + Etc.
* Profundizar en el aprendizaje del lenguaje Python, ya que durante el curso del máster he podido conocer más en profundidad el lenguaje R y encuentro en éste trabajo de final de máster la oportunidad ideal para realizar una inmersión en una herramienta con indudable interés laboral.
* Adquirir competencias en la comprensión y la ejecución de análisis predictivos, haciendo uso de las técnicas “state of the art” como los algoritmos lgbm, xgboost, etc.
* Aprender técnicas de ingeniería de datos a efectos de afinar los resultados.
* Obtener de un modelo predictivo de las ventas de Corporación Favorita lo mas preciso posible y con ello obtener una buena calificación en la competición.

## Enfoque y método seguido

Esta competición trata de obtener un pronóstico del número de ítem/tienda que se venderán durante 2 semanas de agosto de 2017, basado en los datos de ventas del año 2016 y de enero a julio de 2017.

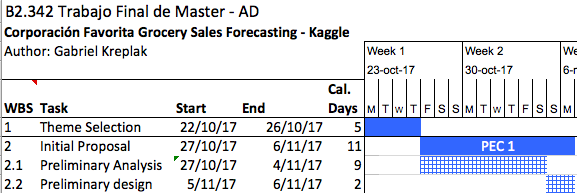
Durante el mismo período de tiempo, se dispone además del numero de transacciones por tienda, clasificación de ítems, precio de petróleo, localización de tiendas y lista de festivos.

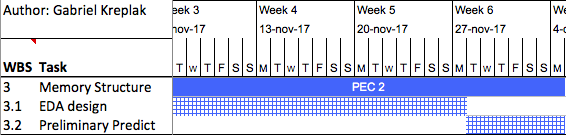
A fin de obtener un modelo predictivo con un nivel mínimo de precisión implementando el algoritmo LGBM, he procedido con los siguientes pasos:

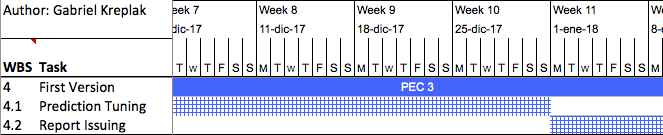
* Carga y limpieza de los datos: La carga de los datos se realiza mediante 6 archivos en formato de valores separados por comas (csv). El dataset de esta competición esta suficientemente depurado, no se han detectado valores atípicos y no requiere modificación destacable de los datos.
* Ingeniería de los atributos: A efectos de conseguir una buena precisión, se realizan cálculos de media de ventas durante diferentes lapsos, como se verá mas adelante y se completará con la información de festivos, precio del petróleo, clase de producto y promociones.
* Análisis exploratorio de datos: Tiene como objeto reflejar las características principales de los datos que se utilizarán en las predicciones.
* Estudio predictivo y envío de resultados: Es la esencia de este trabajo de final de master. Se plica el algoritmo LGBM y se intenta un ajuste de parámetros tendente a conseguir el menor error posible de la predicción aplicada al test set.

## Planificación del trabajo

La planificación del trabajo se divide en la propuesta, análisis y diseño preliminar, estructura de la memoria, tuneado de la predicción y evaluación. El detalle de la planificación se puede visualizar en la Figura 1.







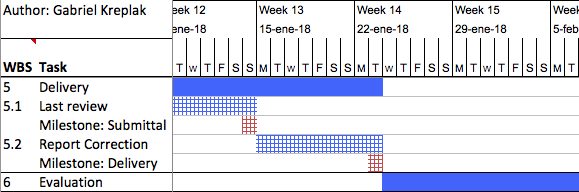


Tabla 1. Planificación

## Descripción de los demás capítulos de la memoria

A continuación se realiza una descripción muy somera del contenido del resto de los capítulos de esta memoria:

* Estado del Arte: esbozo de las tecnologías actuales y referencias a estudios realizados relacionados con el desarrollo de este trabajo final de master.
* Competición Corporación Favorita: exposición de las reglas de la competición, características de los datos de entrada, evaluación y demás detalles de las condiciones iniciales del trabajo.
* Análisis Exploratorio de Datos: Detalle y características de los distintos archivos de datos de la competición.
* Estudio Predictivo: Despliegue del proceso de carga de datos, depuración, ingeniería de atributos, técnica predictiva LGBM y generación de los datos de salida.
* Conclusiones: Lecciones aprendidas, reflexión sobre la consecución de los objetivos planteados inicialmente, análisis crítico del seguimiento de la planificación y líneas de trabajo futuras.
* Glosario.
* Bibliografía.

# Estado del Arte

La Corporación Favorita es uno de los tres supermercados líderes de ventas en Ecuador. Actualmente hay estudios que explican el desarrollo organizacional para su crecimiento [Serrano2016] y analizan el comportamiento del cliente [Baquerizo2017]. También hay estudios estadísticos utilizando análisis multivariante de los factores que intervienen en la compra de marcas propias en el Ecuador, y éstas utilizan el programa R [Santamaría2013].

Actualmente hay diferentes técnicas para problemas de regresión. Una de las más usadas actualmente son los gradient boosting decision trees (GBDT) [Friedman2002]. En [Ponomareva2017], se puede ver una comparativa de las librerías *gradient boosting*. Actualmente, han aparecido algunas técnicas que mejoran su rapidez, como por ejemplo LightGBM [Ke2017, LightGBM2017].

Algunos estudios ya implementan la librería LightGBM, pero en el ámbito de la espectroscopia de suelo [Liu2017], la música [Fonseca2017] y experimentos de física de partículas [Belavin2017]. Esta librería es utilizada en algunos concursos de problemas de regresión. Por ejemplo el ganador de un concurso de e-commerce de 2017 utilizaba la librería LightGBM [Wen2017].

# Competición Corporación Favorita

El objetivo de este trabajo es generar un modelo que permita predecir las ventas de la Corporación Favorita ([http://www.corporacionfavorita.com](http://www.corporacionfavorita.com/)), con el mínimo de error posible.

Para la predicción se ha generado un código disponible en X. Como el código y los datos son abiertos, el resultado es completamente reproducible.

Para la evaluación de la eficacia de la predicción se ha testeado el algoritmo con el 31% de los datos de test. Hay que tener en cuenta no obstante, que para la clasificación final se usa el 69% de los datos de test así que la clasificación puede ser diferente.

## Reglas mas relevantes

* URL: <https://www.kaggle.com/c/favorita-grocery-sales-forecasting/rules>
* Fecha de Inicio: 19 de octubre de 2017
* Fecha final de entrega: 15 de enero de 2017
* Título de la Competición: Corporación Favorita Grocery Sales Forecasting
* Patrocinador: Corporación Favorita C.A.
* Primer premio: $15.000
* Segundo premio: $10.000
* Tercer premio: $5.000
* Datos Externos: Permitidos si no pertenecen a Corporación Favorita ni otra entidad en la misma línea de negocio. La fuente de datos externos se ha de validar previamente con los organizadores de la competencia.
* Reglas generales: <https://www.kaggle.com/c/favorita-grocery-sales-forecasting/rules>

## Descripción de los Datos de Entrada

En esta competición, hay que predecir las ventas de unidades por miles de artículos vendidos en diferentes tiendas Favorita ubicadas en Ecuador. Los datos de capacitación incluyen fechas, información de la tienda y del artículo, ya sea que se promocione ese artículo, así como las ventas de la unidad. Los archivos adicionales incluyen información complementaria que puede ser útil para construir sus modelos.

Descripciones de archivos e información:

**train.csv**

* Datos de training, que incluyen *unit\_sales by date, store\_nbr*, y *item\_nbr* y un identificador *id*.
* El objetivo *unit\_sales* que puede ser un entero (e.g., bolsa de patatas) o número real (e.g., 1.5 kg de queso).
* Números negativos de *unit\_sales* representan las devoluciones de u8n producto.
* La columna *onpromotion* column dice si un *item\_nbr* estaba en promoción en una fecha específica *date* y *store\_nbr*.
* Aproximadamente 16% de los valores *onpromotion* de este fichero son *NaN*.
* NOTA: Los datos de entrenamiento no incluyen filas para los artículos que tenían cero ventas por unidad para una combinación de tienda / fecha. No hay información sobre si el artículo estaba o no disponible para la tienda en la fecha, y los equipos deberán decidir la mejor manera de manejar esa situación. Además, hay una pequeña cantidad de elementos vistos en los datos de capacitación que no se ven en los datos de prueba.

**test.csv**

* Datos de Test, con combinaciones *date, store\_nbr, item\_nbr* que tienen que ser predichas, con la información *onpromotion*.
* NOTA: Los datos de prueba tienen una pequeña cantidad de elementos que no están contenidos en los datos de entrenamiento. Parte del ejercicio será predecir un nuevo artículo de ventas basado en productos similares.
* La división de tabla de clasificación pública / privada se basa en el tiempo. Todos los artículos en la división pública también se incluyen en la división privada.

**sample\_submission.csv**

* Un ejemplo de fichero de envío.

**stores.csv**

* Metadata de la tienda, incluye city, state, type, y cluster.
* cluster es un conjunto de tiendas similares.

**items.csv**

* Metadatos del ítem, incluye family, class y perishable.
* NOTA: los artículos marcados como perecederos tienen un peso de 1,25; de lo contrario, el peso es 1.0.

**transactions.csv**

* El numero de transacciones de ventas por cada combinación *date, store\_nbr*. Sólo incluido para el período de tiempo de los datos de entrenamiento.

**oil.csv**

* Precio diario del petróleo Incluye valores durante el período de tiempo del tren y de la prueba. (Ecuador es un país dependiente del petróleo y su salud económica es altamente vulnerable a los choques en los precios del petróleo).

**holidays\_events.csv**

* Holidays y Events, nacionales, provinciales y locales
* NOTA: prestar especial atención a la columna transferida. Las vacaciones que se transfieren oficialmente caen en ese día calendario, pero el gobierno las cambió a otra fecha. Un día transferido es más como un día normal que un día festivo.
* Los días festivos adicionales son días que se agregan a las vacaciones regulares del calendario, por ejemplo, como suele ocurrir en Navidad (lo que hace que la Navidad sea un feriado).

**Notas adicionales**

* Los salarios en el sector público se pagan cada dos semanas el día 15 y el último día del mes. Las ventas de supermercados podrían verse afectadas por esto.
* Un terremoto de magnitud 7.8 sacudió a Ecuador el 16 de abril de 2016. La gente se unió en esfuerzos de ayuda para donar agua y otros productos de primera necesidad que afectaron en gran medida las ventas de los supermercados durante varias semanas después del terremoto.

## Descripción de los Datos de Salida

El objetivo de este proyecto fin de máster es predecir las ventas de la Corporación Favorita propuesto por Kaggle <https://www.kaggle.com/c/favorita-grocery-sales-forecasting/leaderboard>. Para participar en la competición el usuario utilizado ha sido gkreplak <https://www.kaggle.com/gkreplak>, que ha quedado en la posición X del ranking de un total de Y participantes.

Para cada identificación en el conjunto de prueba, se ha predicho la unidad de ventas. Dado que la métrica utiliza ln (y + 1), las predicciones se validan para garantizar que no haya predicciones negativas. El archivo enviado debe contener un encabezado y tener el siguiente formato:

id, unidad de ventas

125497040,2.5

125497041,0.0

125497042,27.9

etc.

## Evaluación

Las predicciones se evalúan utilizando el Normalized Weighted Root Mean Squared Logarithmic Error (NWRMSLE), calculado de la siguiente manera:



Ecuación 1: NWRMSLE

donde para la fila i, ŷ es la venta de unidad predicha de un artículo e yi es la unidad de ventas real; n es el número total de filas en el conjunto de test.

Los pesos, wi, se pueden encontrar en el archivo items.csv (consulte la página de datos). Los artículos perecederos tienen un peso de 1.25 donde todos los demás artículos tienen un peso de 1.00.

Esta métrica es adecuada para predecir valores en un amplio rango de órdenes de magnitud. Evita penalizar las grandes diferencias en la predicción cuando tanto el número predicho como el verdadero son grandes: predecir 5 cuando el valor verdadero es 50 se penaliza más que la predicción de 500 cuando el valor verdadero es 545.

# Análisis Exploratorio de Datos

## Estructura de los archivos de datos de entrada

Los datos de entrenamiento constan de 6 archivos cuyos atributos se relacionan como se indica en la Figura 1: Estructura de los Datos de Entrada.

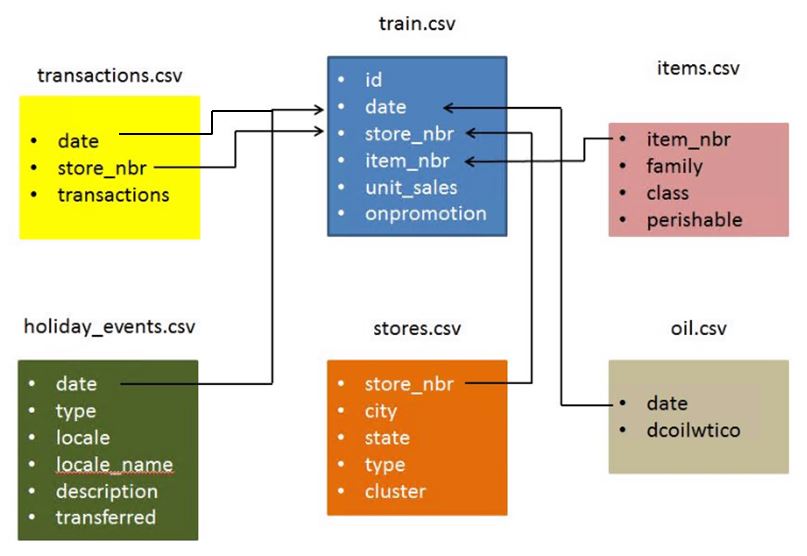
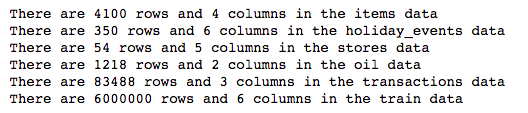


Figura 1: Estructura de los Datos de Entrada.

Fuente: <https://www.kaggle.com/jeru666/all-csv-files-a-glance>

**Recuento:**

****

## Archivo train.csv

Cada línea supone la venta de un único producto. Como ejemplo 5 líneas:

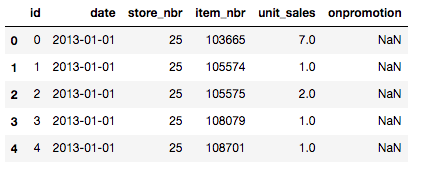


Tabla 2: train.csv, 125 millones de líneas x 6 columnas

## Archivo transactions.csv

Número de transacciones por tienda y día desde el año 2013. 5 líneas de ejemplo:

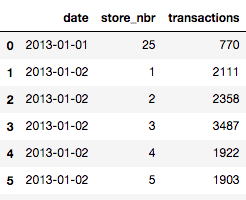


Tabla 3: transactions.csv, 83.500 líneas x 3 columnas

## Archivo holiday\_events.csv

Días festivos desde el año 2012 hasta la actualidad.

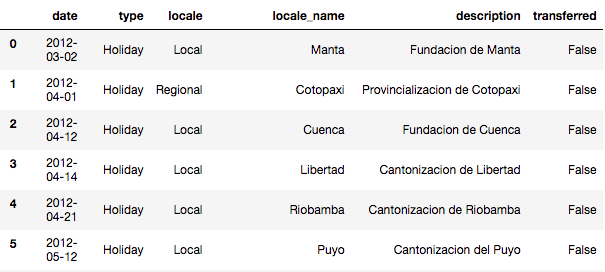


Tabla 4: holiday\_events.csv, 350 líneas x 6 columnas

## Archivo oil.csv

Cotización del petróleo desde el año 2013

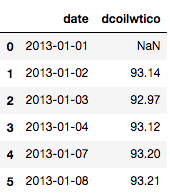


Tabla 5: oil.csv, 1.218 x 2 col

Figura 2: Precios de petróleo

## Archivo items.csv

Catálogo y clasificación de productos a la venta.

La información de los productos se limita a asignar familia a cada producto y como se verá abajo, la inmensa mayoría con comestibles. Indica también si son precederos

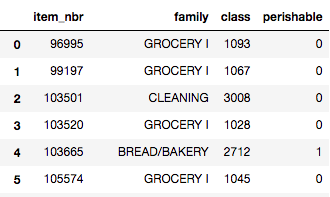


Tabla 6: items.csv, 4.100 productos x 4 attr

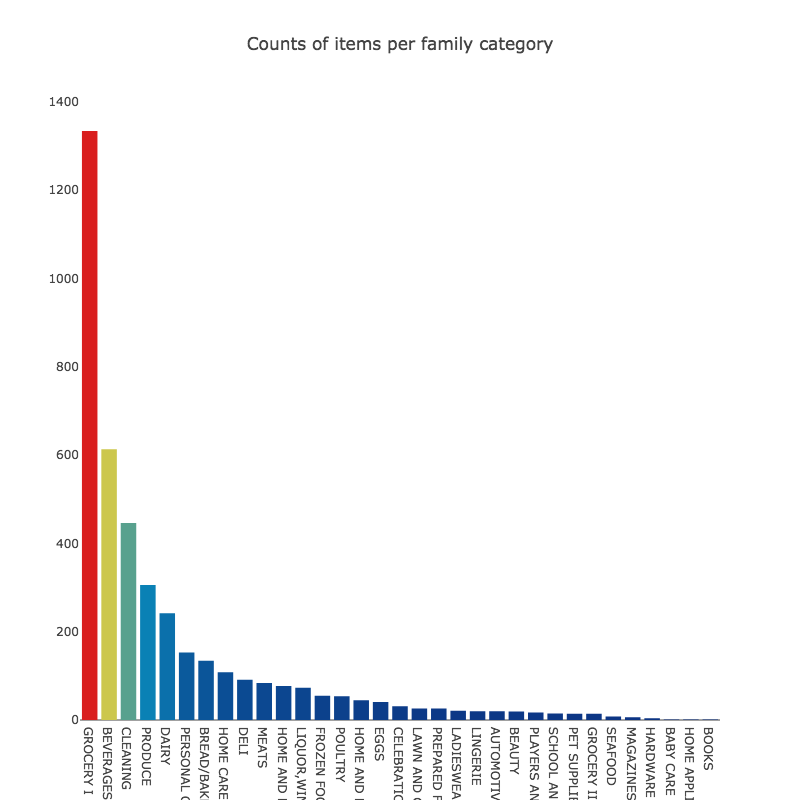


Figura 3: Número de productos por cada categoría

Este gráfico interactivo destaca la proporción logarítmica de numero de productos, siendo los mas numerosos los comestibles, bebidas, limpieza y lácteos.

## Archivo stores.csv

Se trata de 54 tiendas con indicación de la ciudad y el estado. También se incluye una agrupación de tiendas en 17 grupos(clusters) y una clasificación de tienda entre 8 tipos.

18 tiendas están en Quito y 8 en Guayaquil, las 28 restantes están distribuidas entre 20 ciudades distintas.



Tabla 7: stores.csv, 54 tiendas x 5 atributos

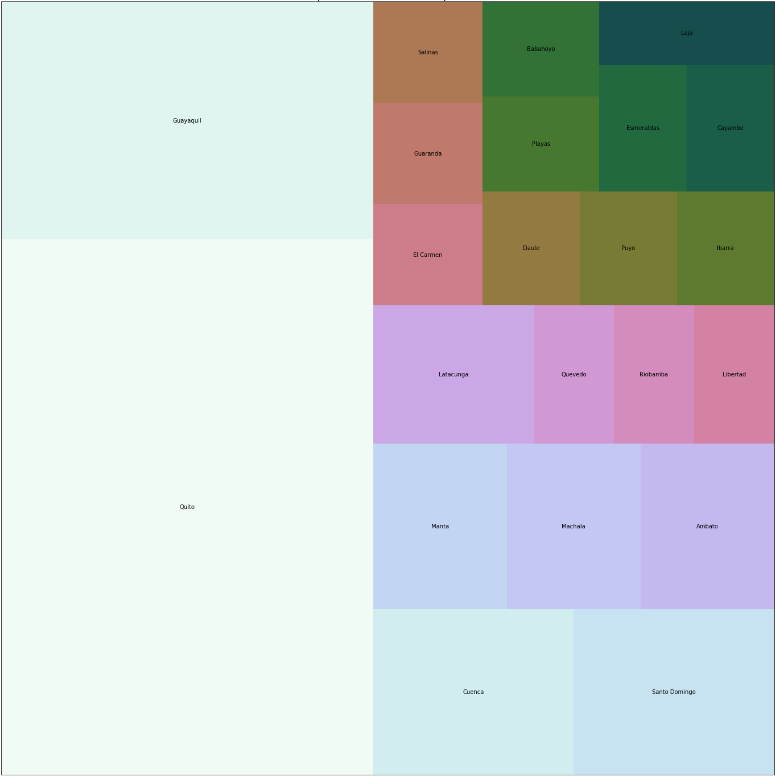


Figura 4 : Treemap de tiendas/ciudad

A continuación se incluye un diagrama indicativo de la distribución del numero de tienda en cada uno de los 16 grupos (clusters) predefinidos.

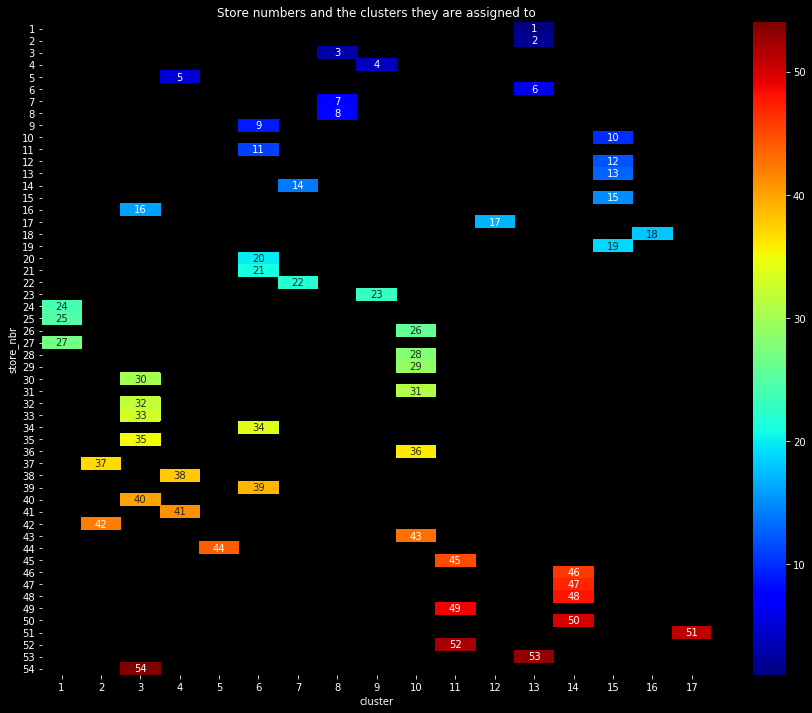


Figura 5: Segmentación de las tiendas

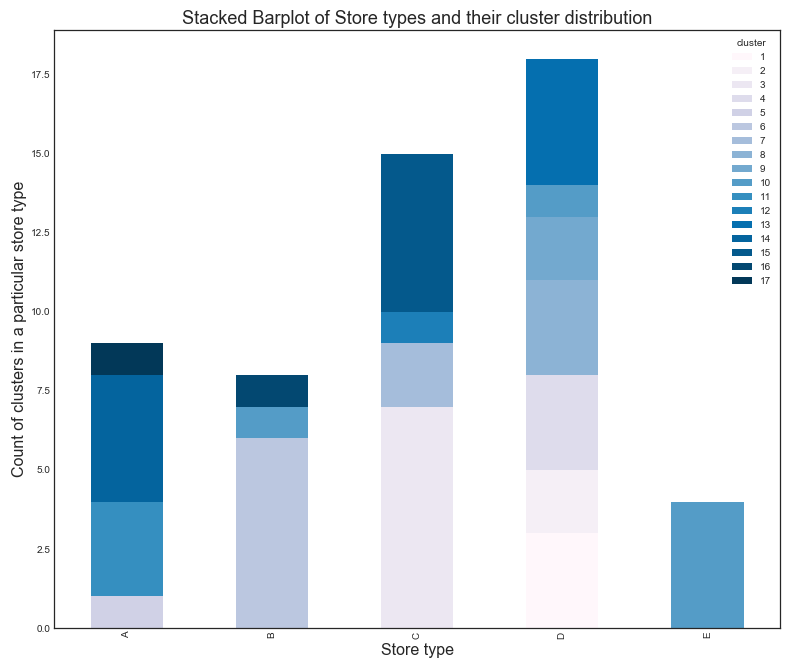


Figura 6: Clasificación de tiendas

## Archivo holiday\_events.csv

Este archivo contiene una línea por cada día festivo en Ecuador, que puede ser nacional, estatal o local. Las ventas son muy sensibles a los días festivos, produciéndose variaciones significativas.

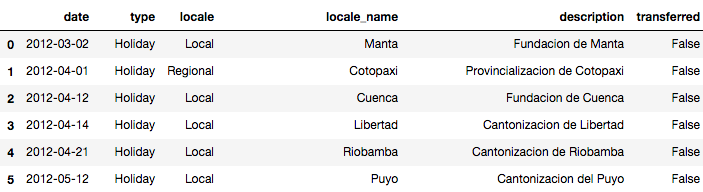


Tabla 8: holiday\_events.csv, 350 líneas x 6 columnas

## Archivo transactions.csv

Un dato importante a considerar es que la ventana de tiempo de las transacciones incluidas en este apartado corresponde al tiempo de los datos de training.

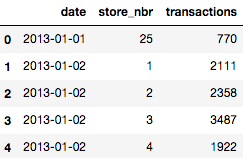


Tabla 9: transactions.csv, 83.500 líneas x 3 columnas

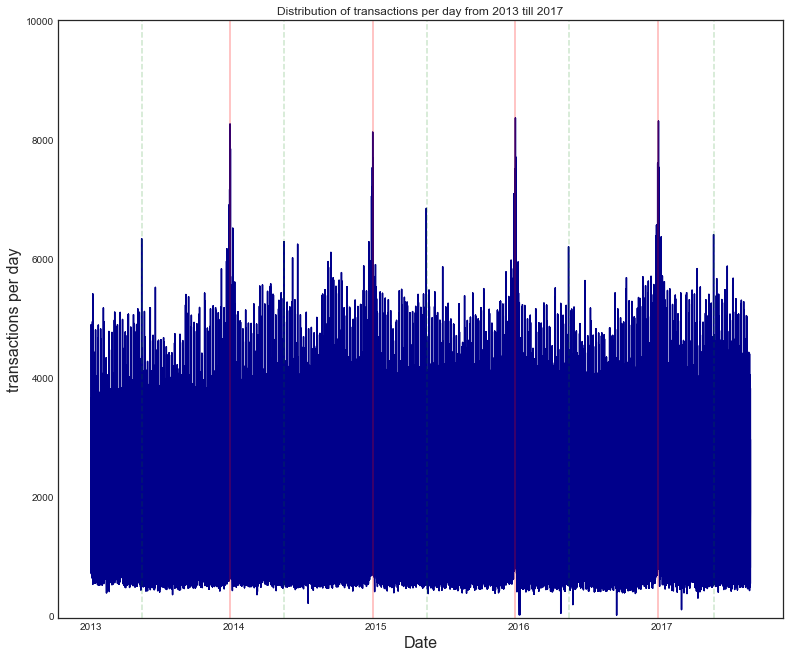


Figura 7: Distribución de transacciones por día

## Archivo test.csv

Este archivo posee la misma estructura que train.csv, excepto la columna unit\_sales, que no existe en este fichero.

Básicamente contiene los datos de numero de ítem, numero de tienda, fecha y en promoción. Con estos identificadores, se procederá a la predicción de las unidades a ser vendidad en esa fecha para cada producto en cada tienda que lo ponga a la venta.

El archivo contiene 3.370.464 de líneas que corresponden a cada uno de los productos existentes en cada tienda durante los 16 días desde 18/8/2017 hasta 31/8/2017.

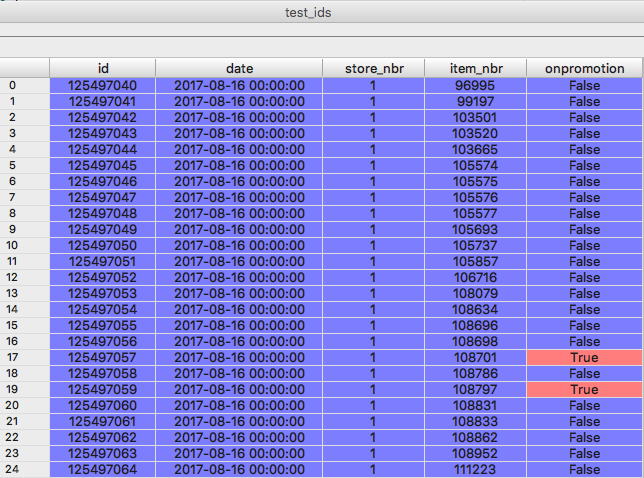


Tabla 10: test.csv 3.370.464 lineas x 5 columnas

# Estudio Predictivo

## Algoritmo Light Gradient Boosting Machine

Como se dice mas arriba, el algoritmo de predicción escogido para el cálculo de las predicciones es: Light Gradient Boosting Machine (LGBM).

Es un algoritmo de tipo Gradient Boosting basado en árboles de decisión desarrollado en el marco del proyecto DMTK de Microsoft.

<http://github.com/microsoft/dmtk>

Las ventajas mas importantes de ésta herramienta son:

* Rapidez en el aprendizaje con alta eficiencia
* Precisión mejorada respecto a otros GBM
* Soporta proceso paralelo y GPU
* Adecuado para datasets de gran tamaño.

La mayoría de algoritmos de árboles crecen por nivel, tal como se indica a continuación:

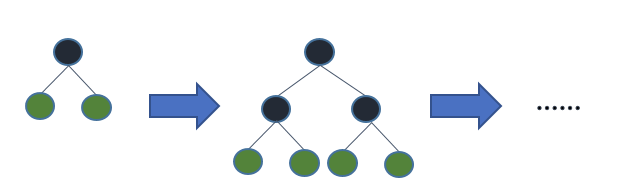


Figura 8: Crecimiento del árbol por niveles.

Fuente: <https://github.com/Microsoft/LightGBM/blob/master/docs/Features.rst>

En cambio, los árboles LightGBM crecen por hojas, según la Figura 9: Crecimiento LGBM por hojas

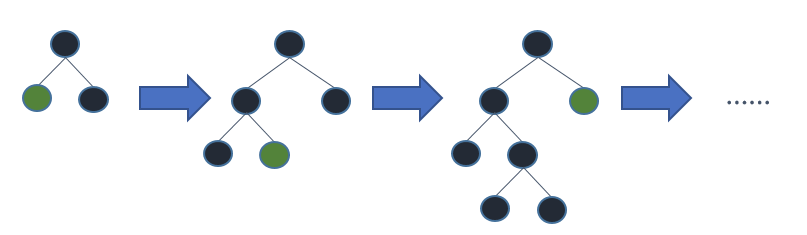


Figura 9: Crecimiento LGBM por hojas

LGBM escogerá para crecer la hoja con mayor delta loss, reduciendo el mismo respecto a un modelo de árbol de crecimiento por niveles.

LGBM no es adecuado cuando el dataset es pequeño porque el crecimiento por hojas conduce al sobre-entrenamiento. En ese caso conviene limitar la profundidad del árbol mediante el parámetro max\_depth.

Una ventaja importante del LGBM es la mayor rapidez de convergencia respecto a arboles de crecimiento por nivel.

## Ajuste de parámetros LGBM

Algunos parámetros importantes:

* **num\_leaves**: Es el principal parámetro para controlar la complejidad del árbol. Tiene que ser menor de 2(max\_depth) para evitar sobre-entrenamiento.
* **min\_data\_in\_leaf**: Importante para controlar el sobre-entrenamiento y depende de num\_leaves. Un número alto evita el crecimiento excesivo del árbol, aunque puede producir sub-entrenamiento.
* **max\_depth**: Límite explícito de la profundidad del árbol.

Para aumentar la velocidad:

* Utilizar bagging activando de **bagging\_fraction** y **bagging\_freq**
* Utilizar feature sub-sampling activando **feature\_fraction**
* Utilizar **max\_bin** pequeño
* Guardar datos binarios usando **save\_binary**

Para aumentar la precisión:

* Utilizar max\_bin grande (puede ralentizar)
* Usar bajo learning\_rate con alto num\_iterations
* Usar alto **num\_leaves** (puede causar sobre-entrenamiento)
* Probar **dart**

Para gestionar el sobre-entrenamiento

* Reducir **max\_bin**
* Reducir num\_leaves
* Usar min\_data\_in\_leaf y min\_sum\_hessian\_in\_leaf
* Utilizar bagging activando de **bagging\_fraction** y **bagging\_freq**
* Utilizar feature sub-sampling activando **feature\_fraction**
* Probar lambda\_l1, lambda\_l2 y min\_gain\_to\_split para regularización
* Probar **max\_depth** para evitar crecimiento de la profundidad del árbol

## Descripción del programa

En el anexo, se adjunta copia del script escrito en python y basado originalmente en una publicación en la plataforma Kaggle de Ceshine Lee, aunque transformado a efectos de mejorar la exactitud de las predicciones.

Es un programa sencillo que realiza movimientos de datos a veces voluminosos y complejos. Los pasos en los que se puede dividir son los siguientes:

1. Lectura
   1. Lectura de datasets de entrada: Los archivos train.csv, test.csv y ítems.csv son leídos y almacenados en los dataframes *df\_train, df\_test* e *items*.
   2. De *df\_train* se descartan las operaciones que no corresponden al año 2017 y quedando los datos útiles en el dataframe *df\_2017*.
   3. Se crea el dataset *promo\_2017* que contiene 167515 líneas que corresponden a las distintas combinaciones tienda/producto que existen en los dataframes *df\_train* y *df\_test*. *promo\_2017* contiene 243 columnas correspondiente a cada uno de los días desde el 1/1/2017 hasta el 31/8/2017. El contenido de *promo\_2017* es binario. Informa sobre los productos/tienda en promoción cada uno de los días.
2. Preparación Dataset

A continuación se detalla la preparación de los datasets necesarios para la ejecución del entrenamiento y predicción del algoritmo LGBM.

Se utiliza respectivamente como predictores y etiquetas X\_train y Y\_train para aprendizaje y X\_val y Y\_val para validación.

Para el cálculo de las predicciones finales para enviar a Kaggle, se dispondrá de X\_test.

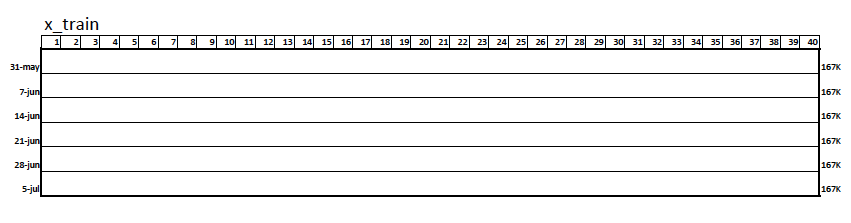
* 1. Predictores Entrenamiento X\_train

El formato del dataset de entrenamiento del algoritmo LGBM sufre una transformación radical respecto a la estructura de atributos original que se puede ver en la Figura 1: Estructura de los Datos de Entrada.

Esta estructura original constaba de unas 125 millones de líneas correspondientes a cada referencia de ítem vendida en cada tienda durante los años 2016 y hasta agosto de 2017.

A efectos de esta predicción de ventas se descarta la información del año 2016 y se trabaja únicamente con los 59 millones de registros correspondientes al año 2017. Las columnas originales de entrenamiento son: *date, store\_nbr y onpromotion.* La etiqueta o target es el atributo: *unit\_sales.*

La transformación aplicada en este dataset original produce un nuevo dataset tal como se describe en la Tabla 11: Estructura de X\_train.



Fechas de referencia

Tabla 11: Estructura de X\_train

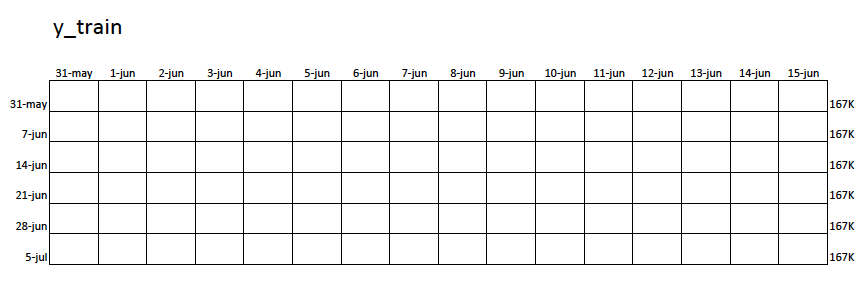
Este nuevo dataset consta, para cada una de las fechas de referencia y cada uno de los 167.515 items/tienda de 40 columnas según se detalla a continuación:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre atributo** | **Atributo original** | **Agregación** | **Días antes de fecha referencia** |
| *day\_1\_2017* | *unit\_sales* |  | 0 |
| *mean\_3\_2017* | *unit\_sales* | Promedio | 3 |
| *mean\_7\_2017* | *unit\_sales* | Promedio | 7 |
| *mean\_14\_2017* | *unit\_sales* | Promedio | 14 |
| *mean\_30\_2017* | *unit\_sales* | Promedio | 30 |
| *mean\_60\_2017* | *unit\_sales* | Promedio | 60 |
| *mean\_140\_2017* | *unit\_sales* | Promedio | 140 |
| *promo\_14\_2017* | *promo* | Suma | 14 |
| *promo\_60\_2017* | *promo* | Suma | 60 |
| *promo\_140\_2017* | *promo* | Suma | 140 |
| *mean\_4\_dow0\_2017* | *unit\_sales* | Promedio ventas de domingos durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow1\_2017* | *unit\_sales* | Promedio ventas de lunes durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow2\_2017* | *unit\_sales* | Promedio ventas de martes durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow3\_2017* | *unit\_sales* | Promedio ventas de miércoles durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow4\_2017* | *unit\_sales* | Promedio ventas de jueves durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow5\_2017* | *unit\_sales* | Promedio ventas de viernes durante 4 sem. Antes fecha ref. | |
| *mean\_4\_dow6\_2017* | *unit\_sales* | Promedio ventas de sábados durante 4 sem. Antes fecha ref. | |
| *mean\_20\_dow0\_2017* | *unit\_sales* | Promedio ventas de domingos durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow1\_2017* | *unit\_sales* | Promedio ventas de lunes durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow2\_2017* | *unit\_sales* | Promedio ventas de martes durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow3\_2017* | *unit\_sales* | Promedio ventas de miércoles durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow4\_2017* | *unit\_sales* | Promedio ventas de jueves durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow5\_2017* | *unit\_sales* | Promedio ventas de viernes durante 20 sem. Antes fecha ref. | |
| *mean\_20\_dow6\_2017* | *unit\_sales* | Promedio ventas de sábados durante 20 sem. Antes fecha ref. | |
| *promo\_1* | *promo* | Estado promo 1 día antes f.r. | |
| *promo\_2* | *promo* | Estado promo 2 días antes f.r. | |
| *promo\_3* | *promo* | Estado promo 3 días antes f.r. | |
| *promo\_4* | *promo* | Estado promo 4 días antes f.r. | |
| *promo\_5* | *promo* | Estado promo 5 días antes f.r. | |
| *promo\_6* | *promo* | Estado promo 6 días antes f.r. | |
| *promo\_7* | *promo* | Estado promo 7 días antes f.r. | |
| *promo\_8* | *promo* | Estado promo 8 días antes f.r. | |
| *promo\_9* | *promo* | Estado promo 9 días antes f.r. | |
| *promo\_10* | *promo* | Estado promo 10 días antes f.r. | |
| *promo\_11* | *promo* | Estado promo 11 días antes f.r. | |
| *promo\_12* | *promo* | Estado promo 12 días antes f.r. | |
| *promo\_13* | *promo* | Estado promo 13 días antes f.r. | |
| *promo\_14* | *promo* | Estado promo 14 días antes f.r. | |
| *promo\_15* | *promo* | Estado promo 15 días antes f.r. | |
| *promo\_16* | *promo* | Estado promo 16 días antes f.r. | |

Tabla 12: Atributos dataset entrenamiento

* 1. Etiquetas Entrenamiento Y\_train

Para las etiquetas o target, se prepara un dataframe correspondiente a las ventas durante los 16 días anteriores a la fecha de referencia, según consta en la Tabla 13: Estructura dataframe Y\_train.



Fechas de referencia

Tabla 13: Estructura dataframe Y\_train

Como se verá mas adelante, para calcular las predicciones de unidades vendidas de cada producto/tienda (atributo: *unit\_sales*) durante el período correspondiente a los datos de test, se utilizan 16 etiquetas por cada producto/tienda correspondientes a los 15 días posteriores a la fecha referencia.

Finalmente, los predictores y etiquetas de entrenamiento son el conjunto de 6 grupos formados cada uno por las 167.515 líneas generadas según se describe en este apartado. Cada uno de estos grupos se genera con fechas de referencia semanal de 6 semanas posteriores a la fecha de referencia inicial.

* 1. Predictores y Etiquetas Validación

Estos datasets tienen un uso interno para el cálculo de las predicciones del algoritmo LGBM, como se verá en el apartado correspondiente.

En cuanto a su estructura de columnas, son idénticos a los datasets de predictores y etiquetas de entrenamiento, descritos en: Predictores Entrenamiento X\_train y Etiquetas Entrenamiento Y\_train.

Estos datasets de validación tienen un tamaño de 167.515 líneas correspondientes a las combinaciones producto/tienda de fecha referencia: 26/7/2017, tal como se refleja en la Tabla 14 y la Tabla 15.

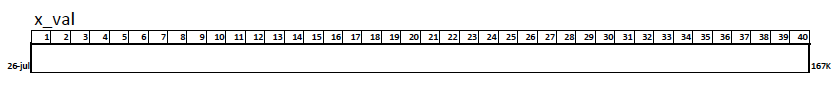


Tabla 14: Estructura de Predictores de Validación

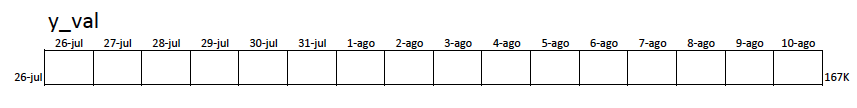


Tabla 15: Estructura de Etiquetas de Validación

* 1. Predictores de Test

El dataframe de predictores de test se genera de forma análoga a lo descrito en los apartados descriptivos de los dataframes de entrenamiento y validación.

La fecha de referencia utilizada es 16/8/2017

En el caso de los datos de test, no existen etiquetas y la operación de medición de precisión de la predicción la realiza Kaggle utilizando según la Ecuación 1: NWRMSLE

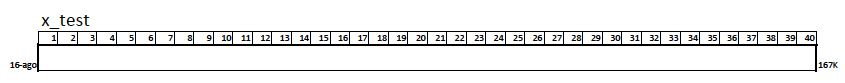


Tabla 16: Estructura de predictores de test

1. Entrenamiento de los modelos

Previo al lanzamiento del aprendizaje del algoritmo LGBM, se definen los parámetros principales tal como se detallan en Ajuste de parámetros LGBM. Estos serán elementos susceptibles de ajuste posterior para obtener una predicción ajustada.

El proceso de entrenamiento de los modelos se ejecuta en un bucle de 16 iteraciones en la que en cada una se utiliza como etiqueta, una de las 16 columnas generadas como se describe en Etiquetas Entrenamiento Y\_train.

Para el entrenamiento se utilizan los predictores y las 16 columnas de etiquetas de entrenamiento así como los predictores y la columna de etiquetas de validación.

ESTUDIAR COMO SE OBTIENE EL MEJOR MODELO A PARTIR DE LAS 16 ITERACIONES

La estrategia de ajuste seguida para la optimización de los resultados del algoritmo incluye los siguientes aspectos y se resumen en la **¡Error! No se encuentra el origen de la referencia.**

* 1. Ajuste de parámetros
  2. Ajuste de fechas de referencia
  3. Ajuste de atributos de entrenamiento
  4. Ajuste de rango de fechas de etiquetas

Timedelta= -8, score 0,633

Timedelta= -4, score 0,668

Timedelta= +4, score 0,570

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Learning rate** | **Fecha Referencia** | **Score** |
|  | 0.03 | 21/6/17 | **0.513** |
|  | 0.03 | 31/5/17 | **0.515** |
|  | 0.025 | 31/5/17 | **0.516** |
|  | 0.03 | 20/7/17 |
|  | 0.01 | 31/5/17 | **0.517** |
|  | 0.03 | 26/6/17 | **0.518** |
|  | 0.03 | 19/6/17 | **0.519** |
|  | 0.03 | 22/6/17 | **0.520** |
|  | 0.03 | 18/6/17 | **0.523** |
|  | 0.03 | 24/6/17 | **0.524** |
|  | 0.1 | 31/5/17 | **0.529** |
|  |  |  |  |
|  |  |  |  |

Tabla 17: Resumen de Ajustes

1. Generación de la Predicciones

Una vez creado el modelo *bst* basado en los predictores *X\_train*, se utiliza generando 2 predicciones:

* 1. *val\_*pred: predicción generada con los predictores *X\_val*, descritos en Predictores y Etiquetas Validación. Esta predicción se utilizará para el cálculo interno de las métricas: mse (mean squared error) para:
* el train dataset sin y con los pesos de productos perecederos (1,25)
* el train dataset considerando las primeras 5 líneas y las líneas siguientes
  1. *test\_pred*: predicción generada con los predictores *X\_test,* descritos en Predictores de Test.

1. Preparación del envío de archivos

En esta última fase, se unen las predicciones generadas basadas en X\_test con el fichero inicial test.csv, obteniéndose un archivo para el envío a Kaggle de 2.680.240 líneas con las predicciones de los 167.515 producto/tienda para cada uno de los 16 días de los datos del test dataset.

# Conclusiones

Lecciones aprendidas:

Para ajustar un modelo predictivo de Kaggle es necesario:

* tiempo para depuración
* hardware suficientemente potente para el tratamiento de datasets de tamaño medio a una velocidad asumible
* equipo humano con formación en ML para brainstorming

Objetivos Planteados:

* Conseguir score Kaggle profesional.
  + Conseguido?
  + Porque?
  + Trabajo Futuro: Participación intensiva en competiciones
* Uso plataformas GitHub, PyCharm
  + Conseguido: No
  + Porque: Falta de tiempo
  + Trabajo Futuro: Desarrollo proyectos
* Profundización lenguaje Python
  + Conseguido: + o –
  + Porque: Falta de tiempo
  + Trabajo Futuro: Asistencia curso avanzado Python
* Aprendizaje y profundización algoritmo Light GBM
  + Conseguido: bastante
* Aprendizaje técnicas de afinación de predicciones
  + Conseguido: No
  + Porque: falta de tiempo
  + Trabajo futuro: ¿?

Seguimiento de la planificación: Correcta, gracias a la ayuda de la tutora.

# Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

# Bibliografía

Lista numerada de las referencias bibliográficas utilizadas dentro de la memoria. En cada lugar donde se utilice una referencia dentro del texto, hay que indicarla citando el número de la referencia, por ejemplo: [7].

Es muy importante incluir **todas** las referencias utilizadas y citarlas apropiadamente, es decir, incluyendo toda la información necesaria para identificar la referencia. La información mínima que hay que incluir según el tipo de referencia es:

* **Libro:** Autores, Título, Edición (si se tercia) Editorial, Ciudad, Año.
* **Artículo de revista:** Autores, Título, Nombre de la Revista, Número de Página inicial y final, Número de la revista / Volumen, Año.
* **Web:** URL y fecha en que se ha visitado.

### Referencias:

* Dataframe with all Date-Store-Item Combinations
* Comprehensive Python and D3.js Favorita analytics. [https://www.kaggle.com/arthurtok/comprehensive-python-and-d3-js-favorita-analytics](https://www.kaggle.com/arthurtok/comprehensive-python-and-d3-js-favorita-analytics" \t "_blank)
* Memory Optimization and EDA on entire dataset. [https://www.kaggle.com/jagangupta/memory-optimization-and-eda-on-entire-dataset](https://www.kaggle.com/jagangupta/memory-optimization-and-eda-on-entire-dataset" \t "_blank)

# Anexos

Listado de apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido (por ejemplo, manuales de usuario, manuales de instalación, etc.)

Dependiente del tipo de trabajo, es posible que no haya que añadir ningún anexo.