AVANCE PROYECTO INTELIGENCIA ARTIFICIAL

Store sales - Time series forecasting

Inteligencia artificial

INTEGRANTES

Miguel Ángel Rivas Acevedo
Santiago Lorduy Gil

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
MEDELLÍN

2022

Store sales - Time series forecasting

Se inicia buscando con mucho detenimiento y descargando uno a uno los datasets de diversas competencias en kaggle, se revisa y analiza el contenido, hasta que finalmente se encuentra un conjunto de datos que cumple con las características pedidas y una competencia agradable y que ve bastante prometedora e interesante para trabajar, esta se llama Store sales - Time series forecasting.

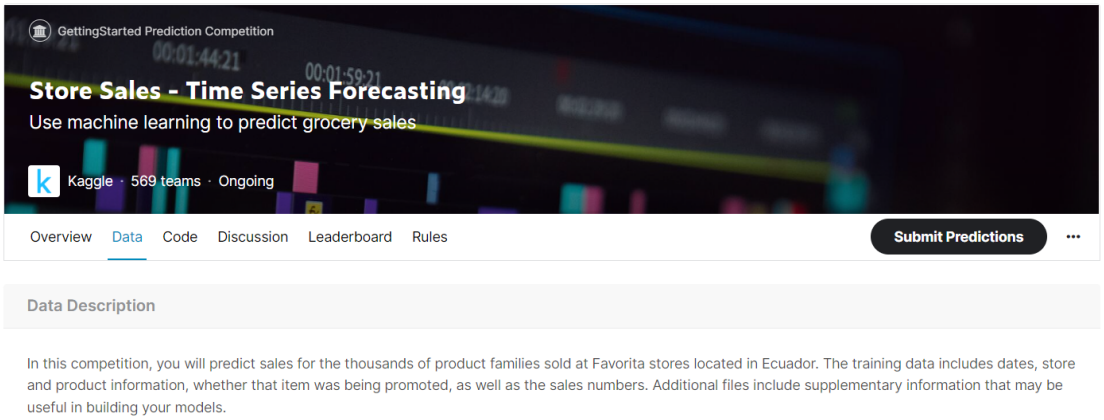

Fig 1. Competencia elegida.

En esta competencia contamos con un dataset de siete archivos, todos son .csv y se tiene uno para entrenar al algoritmo y otro para probarlo.

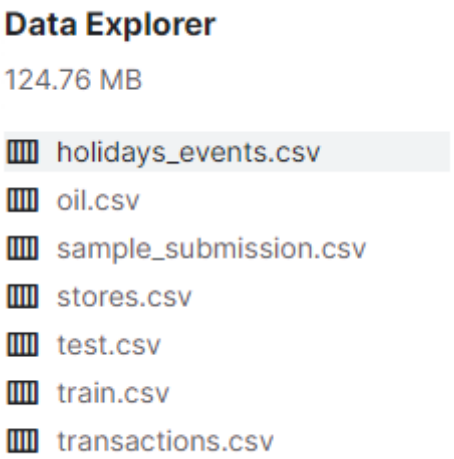

Fig 2. Dataset

En el colab se importan todas las librerías necesarias y se suben todos los archivos, Se descomprimen los datos .zip de la competencia, posteriormente se realiza una lectura de los archivos usando la librería *pandas* y se imprime los datos de estos para ver su dimensión.

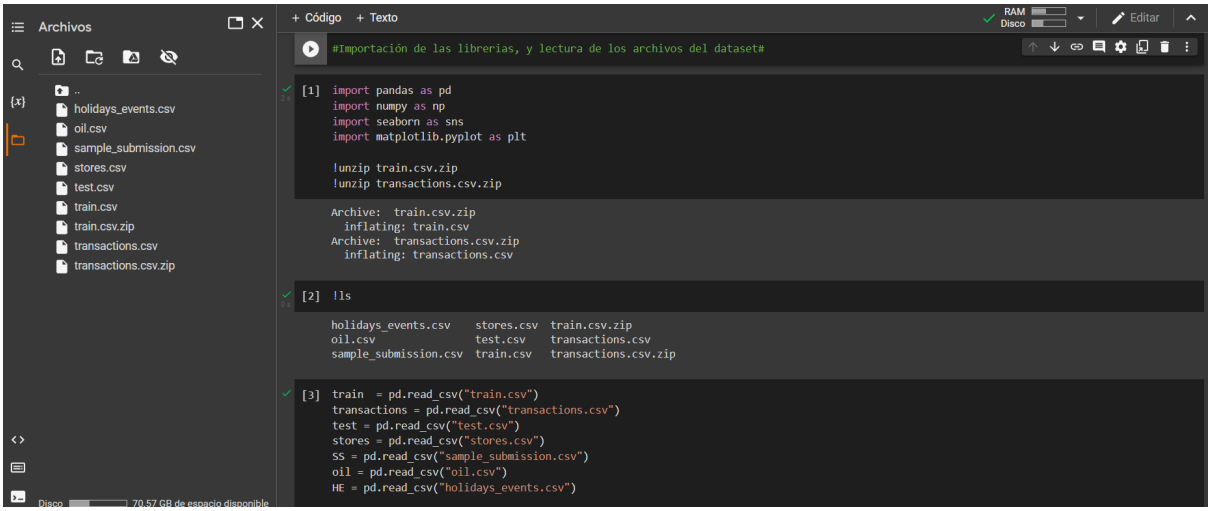

Fig 3. Importación de los modulos y carga del dataset.

Después de observar el contenido de cada archivo, se utiliza la función isnull.num para ver qué columnas tienen valores NaN y así poder reemplazarlos por valores que puedan ser usados posteriormente con mayor eficiencia, para este caso se cambiaron por cero, debido a que tras hacer un análisis se concluyó que era la manera para que la ausencia no afectara el resultado. Continuando se realiza una concatenación de los archivos creando así un train y un test completos que facilitan la manipulación de datos y la facilidad para hacer gráficas con ellos e interpretarlos, todo esto para tener la mejor predicción posible.

```python
#Unión de los archivos para obtener un solo dataset con todas las columnas y filas necesarias
```

```python
[16] train_ = pd.merge(train,oil, how = "left", on = "date")
     train_ = pd.merge(train_, HE, how = "left", on = "date")
     train_ = pd.merge(train_, transactions, how = "left", on =["date","store_nbr"])
     train_ = pd.merge(train_,stores, how = "left", on = "store_nbr")
     train_.rename(columns={"type_x":"holiday_type", "type_y":"store_type"}, inplace = True)
     train_.tail()
```

	id	date	store_nbr	family	sales	onpromotion	dcoilwtico	holiday_type	locale	locale_name	description	transferre
3054343	3000883	2017-08-15	9	POULTRY	438.133	0	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	Fals
3054344	3000884	2017-08-15	9	PREPARED FOODS	154.553	1	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	Fals
3054345	3000885	2017-08-15	9	PRODUCE	2419.729	148	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	Fals
3054346	3000886	2017-08-15	9	SCHOOL AND OFFICE SUPPLIES	121.000	8	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	Fals
3054347	3000887	2017-08-15	9	SEAFOOD	16.000	0	47.57	Holiday	Local	Riobamba	Fundacion de Riobamba	Fals

Fig 4. Train de archivos

Tras esta unión se hace el reemplazo de los valores NaN por cero y también usando la función *bfill* para llenar hacia atrás los NaN

```python
# cambiando los NaN por valores que puedan ser utilizados
```

```python
[20] train_["dcoilwtico"] = train_["dcoilwtico"].fillna(method = "bfill")
     train_["holiday_type"] = train_["holiday_type"].fillna(0)
     train_["locale"] = train_["locale"].fillna(0)
     train_["locale_name"] = train_["locale_name"].fillna(0)
     train_["description"] = train_["locale"].fillna(0)
     train_["transferred"] = train_["transferred"].fillna(0)
     train_["transactions"] = train_["transactions"].fillna(0)
     train_.isnull().sum()
```

Fig 5. Llenado de celdas sin información.

Con lo realizado hasta este punto, se tiene la información lista para ser introducida a un modelo que me asigne una probabilidad de acierto en base a lo buscado, la cual continuará siendo desarrollada en el marco de este proyecto.