MiDS & AI - TECH

Laboratorio de Proyecto

Presentación 30/06/2022, 17H
Presencial
PATIO DE LA INFANTA, SALA ALCARRIA
ZARAGOZA

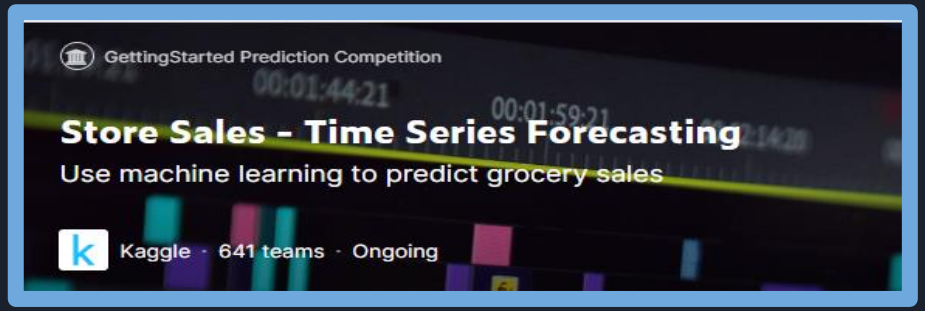
GRUPO R^3:
RAUL PULIDO, RACHELE RENDINA, RODRIGO ANCHELERGUES

MiDS & AI - TECH - Laboratorio de Proyecto

Índice

MiDS & AI - TECH - Laboratorio de Proyecto

A) Selección de competiciones

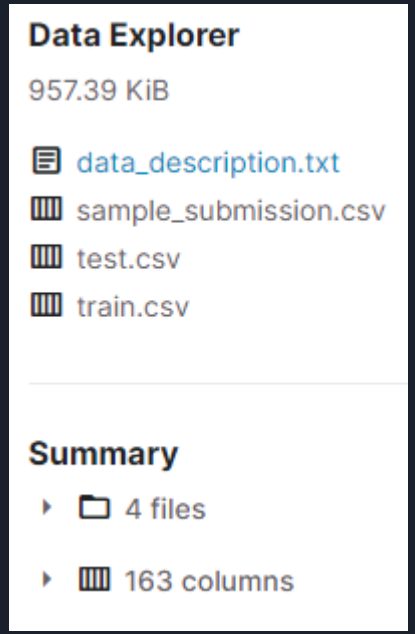


B) KAGGLE: House Prices - Advanced Regression Techniques

1. Descripción

Objetivo: estimación de precios de viviendas

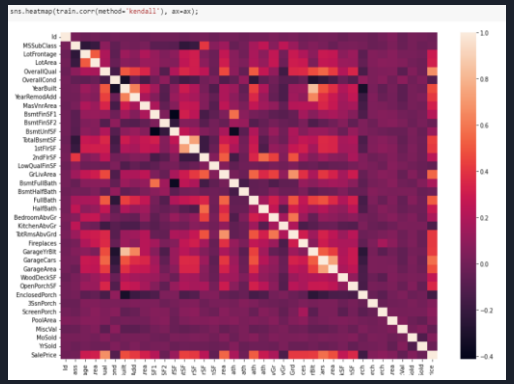
- Datos suministrados: un único dataset con 81 variables relacionadas con localización, antigüedad, estado de las casas, equipamiento.
- Datos train incluyen precios. Relativamente pequeño (1460 filas)
- Fichero test con las mismas variables, pero sin precios.



B) KAGGLE: House Prices - Advanced Regression Techniques

2. Análisis de dataset

- Correlación mediante mapa de calor
- Campos vacíos
- Outliers




```
train.isna().sum().sort_values(asce

PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
FireplaceQu      690
LotFrontage      259
GarageYrBlt       81
GarageCond        81
GarageType        81
GarageFinish      81
GarageQual        81
BsmtFinType2      38
BsmtExposure      38
BsmtQual          37
BsmtCond          37
BsmtFinType1      37
MasVnrArea         8
MasVnrType         8
Electrical         1
Id                 0
dtype: int64
```

B) KAGGLE: House Prices - Advanced Regression Techniques

3. Primeros tanteos

- Pruebas iniciales con un modelo de regresión (XGBoost)
- Dataset inicial
- Primeros posicionamientos en ranking Kaggle

r-3-house-pricing_RA.ipynb	Rac Rac	26 abr 2022 yo
HousePrice_conDepuracion.ipynb	Rac Rac	6 may 2022 Rac Rac
HousePrice_onlyXGBOOST+parameter+depuracion.ipynb	Rac Rac	10 may 2022 Rac Rac
HousePrice_con_varios_modelos.ipynb	Rac Rac	11 may 2022 yo
automl-h2o.ipynb	Rac Rac	12 may 2022 Rac Rac
HousePrice_onlyCatBoostRegressor+parameter+depuracion.ipynb	Rac Rac	12 may 2022 Rac Rac

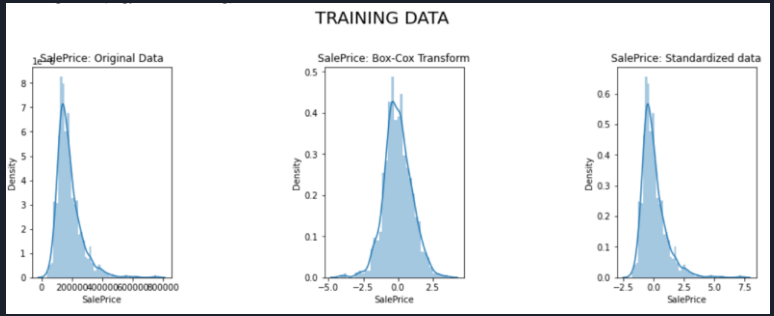
B) KAGGLE: House Prices - Advanced Regression Techniques

4. Transformación de dataset

```
['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']
```

- Eliminación de variables con alto porcentaje de vacíos.
- Completar variables con diferentes estrategias
  - Cualitativos: moda o transformar NA en None
  - Cuantitativos: media o datos de otras variables (fecha)
- Transformación de variables numéricas: min-max scaler
- One-hot encoding variables cualitativas
- Transformación variable objetivo: asemejar a normal
- Mismas transformaciones en test

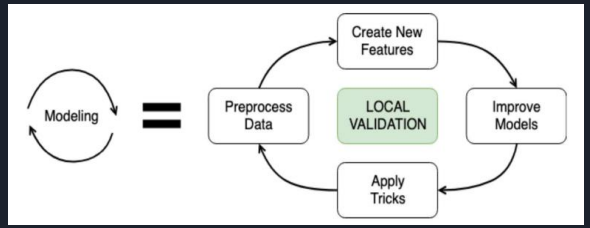
```
# Rellenar campos vacíos (cuantitativos)

#Rellenamos LotFrontage con la media
media= train['LotFrontage'].mean()
train['LotFrontage'].fillna(media, inplace=True)

# Rellenamos GarageYrBlt con YearBuilt
train['GarageYrBlt'].fillna(train['YearBuilt'], inplace=True)
```



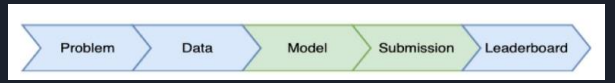
B) KAGGLE: House Prices - Advanced Regression Techniques

5. Modelo

Proceso iterativo para selección del modelo y de los hiper parámetros



- MODELOS INDIVIDUALES
- Optimización de Hiper Parámetros
  - Regularización
  - Búsqueda en el espacio de hyper parámetros (grid search, etc..)
- MODELOS COMBINADOS
  - Combinando la potencia de varios modelos
  - MEZCLADO (BLENDING): la media de las predicciones
  - APILADO (STACKING): utilizar las predicciones de un modelo como nuevas variables para otro modelo



B) KAGGLE: House Prices - Advanced Regression Techniques

5. Modelo

Proceso iterativo para selección del modelo y de los hiper parámetros

- MODELOS INDIVIDUALES
- Optimización de Hiper Parámetros
  - Regularización
  - Búsqueda en el espacio de hyper parámetros (grid search, etc..)

```
Model: XGBRegressor, Score: 0.9154437049345521
Model: LGBMRegressor, Score: 0.9018428526240856
Model: RandomForestRegressor, Score: 0.8759882305439375
Model: ElasticNet, Score: -0.01124048954288659
Model: MLPRegressor, Score: 0.8632540116253808
Model: KNeighborsRegressor, Score: 0.04242045355958435
```

HousePrice_onlyCatBoostRegressor+parameter+depuracion.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Guardado por última vez: 19:51

+ Código + Texto

```
[ ]
```

```
[ ]
# Definimos "unos" hiperparámetros para el modelo

dict_params = { # CatBoostRegressor
                'iterations':[500],
                'depth':[4],
                'learning_rate':[0.03],
                'loss_function':['RMSE'],
                'l2_leaf_reg': [0.2],
                'eval_metric': ['RMSE'],
                'od_type': ['Iter'],
                'od_wait': [1000],
                'rsm': [0.6],
                'random_strength': [2],
                'bagging_temperature': [10]
                }


#{'depth': 6, 'iterations': 200, 'l2_leaf_reg': 0.2, 'learning_rate': 0.05, 'loss_function': 'RMSE'}
#{ # CatBoostRegressor
#                'iterations':[30, 50, 100, 150, 200],
#                'depth':[2, 4, 6],
#                'learning_rate':[0.03, 0.05],
#                'loss_function':['RMSE'],
#                'l2_leaf_reg': [0.2, 0.5,]
#              }
#{'depth': 4, 'iterations': 500, 'l2_leaf_reg': 0.2, 'learning_rate': 0.03, 'loss_function': 'RMSE'}
#{ # CatBoostRegressor
#                'iterations':[200, 300, 500],
#                'depth':[2, 4, 6],
#                'learning_rate':[0.01, 0.03],
#                'loss_function':['RMSE'],
#                'l2_leaf_reg': [0.2, 0.5]
#              }

#{  # XGBRegressor
#              "n_estimators": [200, 250, 300, 400],
#              "max_depth": [6, 15, 20, 25],
#              "learning_rate": [0.03, 0.01, 0.3],
#              "objective": ["reg:squarederror"],
#              "tree_method": ["hist", "approx"],
#              "subsample": [0.6],
#              "colsample_bytree":[0.9],
#              "eta": [0.3],
#              "gamma": [0.01],
#              "grow_policy": ["lossguide"],
#              "max_bin": [1023],
```

B) KAGGLE: House Prices - Advanced Regression Techniques

5. Modelo

Proceso iterativo para selección del modelo y de los hiper parámetros

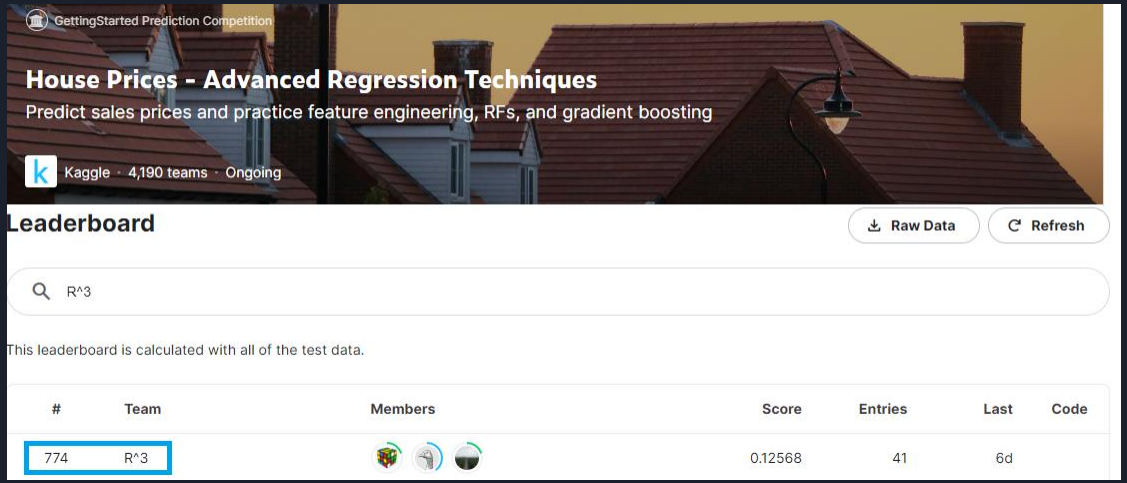
- MODELOS COMBINADOS
  - APILADO (STACKING): utilizar las predicciones de un modelo como nuevas variables para otro modelo

```
from mlxtend.regressor import StackingCVRegressor
from sklearn.linear_model import Ridge, Lasso
from sklearn.svm import SVR

xgb = XGBRegressor()
lgbm = LGBMRegressor()
rf = RandomForestRegressor()
ridge = Ridge()
lasso = Lasso()
svr = SVR(kernel='linear')

stack = StackingCVRegressor(regressors=(ridge, lasso, svr, rf, lgbm, xgb),
                            meta_regressor=xgb, cv=12,
                            use_features_in_secondary=True,
                            store_train_meta_features=True,
                            shuffle=False,
                            random_state=42)

stack.fit(X_train, y_train)

pred = stack.predict(X_test)
score = r2_score(y_test, pred)
print(score)
```

B) KAGGLE: House Prices - Advanced Regression Techniques

6. Predicción y resultado final

- 42 Submissions presentadas
- Mejor modelo CatboostRegressor
- Versión final: posición 774 de 4.190 (<20%)



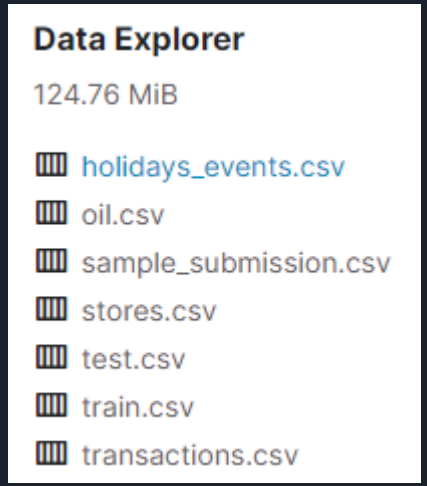
	Id	SalePrice
0	1461	122235.706376
1	1462	158301.267149
2	1463	178810.440249
3	1464	195924.429695
4	1465	188830.422461
...	...	...
1454	2915	82069.611664
1455	2916	84704.788519
1456	2917	161163.791265
1457	2918	112217.560676
1458	2919	232074.137151

C) KAGGLE: Store Sales - Time Series Forecasting

1. Descripción

Objetivo: estimación de ventas en los próximos 15 días.

- **Datos suministrados:**
- Train
  - Ventas por fecha, tienda y familia (3 M de filas)
- Test
  - Fecha, tienda, familia (30.000 filas) para predicción.
- Datos con posible impacto en las ventas: festivos, cotización del petróleo, número de transacciones



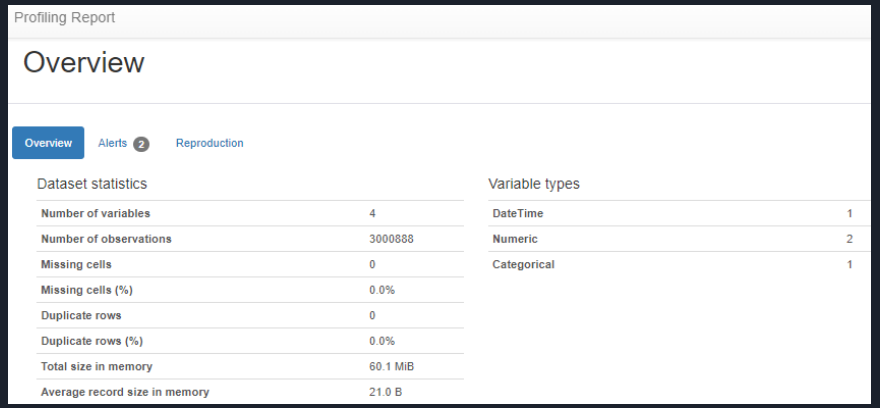
C) KAGGLE: Store Sales - Time Series Forecasting

2. Análisis de datasets

- Revisión del dataset con una herramienta de PROFILING (librería PANDAS_PROFILING)
- Genera un report visual con perfilado de todo el dataset:
  - Análisis de variables: tabla con valores nulos de cada variable, perfilado de cada variable: max, min, promedio, mediana, varianza, desv. estándar…
  - Correlaciones entre variables,, coeficiente Pearson…

```
[ ]  1 from pandas_profiling import ProfileReport
     2 df_profile = ProfileReport(train, title="Profiling Report", progress_bar=True)
     3 profile_html = df_profile.to_html()
```

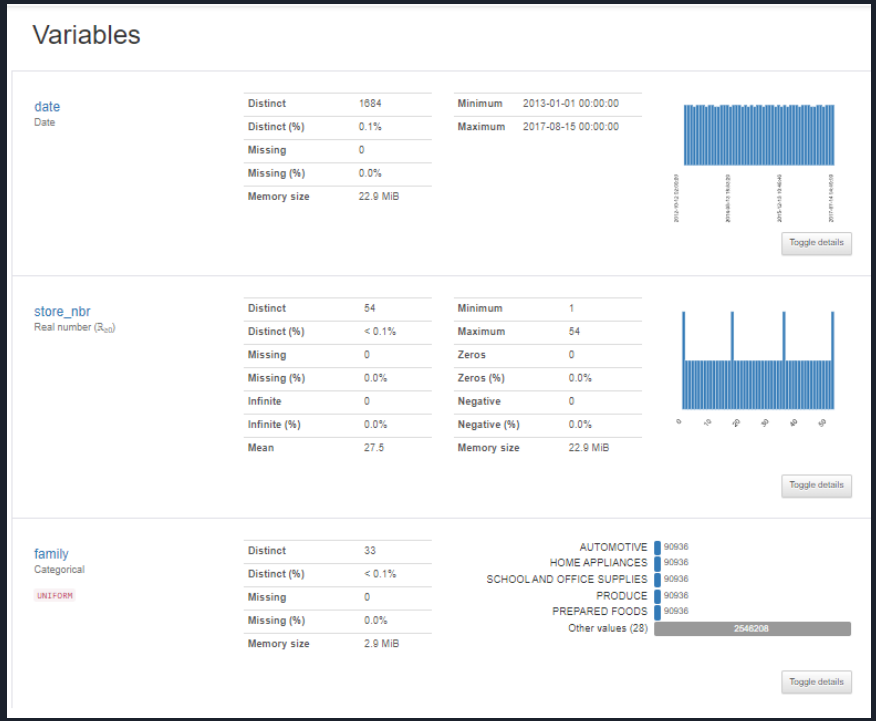
```
[ ]  1 from IPython.display import display, HTML
     2 display(HTML(profile_html))
```



C) KAGGLE: **Store Sales** - Time Series Forecasting

2. Análisis de datasets

- Date: rango de fechas, datos completos, 1684
- Número de tienda
- Categoría familia



C) KAGGLE: Store Sales - Time Series Forecasting

3. Transformación dataset

- Fichero Oil: completar vacíos mediante interpolación
- Agregar cotización Oil en train a través de la fecha
- Determinación festivos por tienda
- Agregar festivos en train para completar dataset

```
[ ] oil.isna().sum().sort_values(ascending=False).head()

    dcoilwtico    43
    dtype: int64

[ ] # Vamos a ver como esos NaN en la primera aproximación los vamos a quitar con una interpolación lineal.
    # Posteriormente se verá si alguno de los datos  por falta de datos reales (cierre de mercados....)
    oil['dcoilwtico'].interpolate(method='linear', inplace=True, limit_direction='both')

[ ] oil.isna().sum().sort_values(ascending=False).head()

    dcoilwtico    0
    dtype: int64
```

```
trainfinal2.head()
```

	date	store_nbr	family	sales	dcoilwtico	holiday
0	2013-01-01	1.0	AUTOMOTIVE	0.0	93.14	True
1	2013-01-01	1.0	BABY CARE	0.0	93.14	True
2	2013-01-01	1.0	BEAUTY	0.0	93.14	True
3	2013-01-01	1.0	BEVERAGES	0.0	93.14	True
4	2013-01-01	1.0	BOOKS	0.0	93.14	True

```
[ ] holidays_by_store.head()
```

	date	store_nbr
0	2012-03-02	1
1	2012-03-02	2
2	2012-03-02	3
3	2012-03-02	4
4	2012-03-02	5

C) KAGGLE: Store Sales - Time Series Forecasting

4. Elección de Modelo

- Los tanteos se traducen a varias versiones en las cuales hemos ido probando distintos modelos según las ayudas que nos dio entre otros nuestro tutor
  1. Regresión lineal - por categoría de familia - negativo: repetirlo por cada familia (33); proceso posible, pero pesado y muy lento

     -> probamos, pero lo descartamos

  1. ARIMA/SARIMA

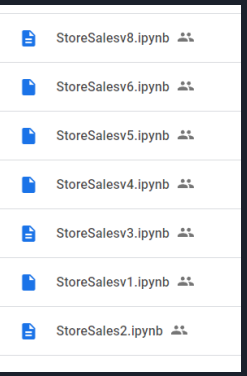
     → llegamos a la conclusión que no aportaba en este caso

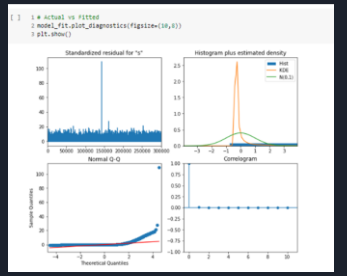
     First kaggle notebook. Following TS tutorial

     Store Sales: Ridge+Voting(Bagging(ET)+Bagging(RF)) | Kaggle

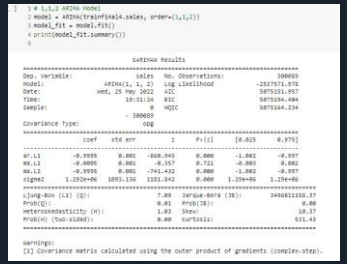
  1. Deterministic - Fourier

     → predicción de los 15 días siguientes






```
from statsmodels.tsa.deterministic import DeterministicProcess

dp = DeterministicProcess(
    index=avg_sales.index,  # dates from the training data
    constant=True,          # dummy feature for the bias (y_intercept)
    order=1,                # the time dummy (trend)
    drop=True,              # drop terms if necessary to avoid collinearity
)
# `in_sample` creates features for the dates given in the `index` argument
X = dp.in_sample()

X.head()
```



ARIMA/SARIMA

C) KAGGLE: Store Sales - Time Series Forecasting
5. Modelo

- Calendar Fourier

  para dar una frecuencia de periodo a nuestro dataset

  -> asemejado a un calendario

- Deterministic Process

  para que un evento del futuro pueda ser calculado sin aleatoriedad

- Modelo Ridge

  es un modelo de regresión con función de pérdida de mínimos cuadrados lineales y con regulación L2 para sacar nuestra predicción

  -> recomendamos esta solución

StoreSales.ipynb

MODELO

```
[ ]  1 #Fechas entre las que hacemos el modelo
     2 sdate = '2017-04-01'
     3 edate = '2017-08-15'
```

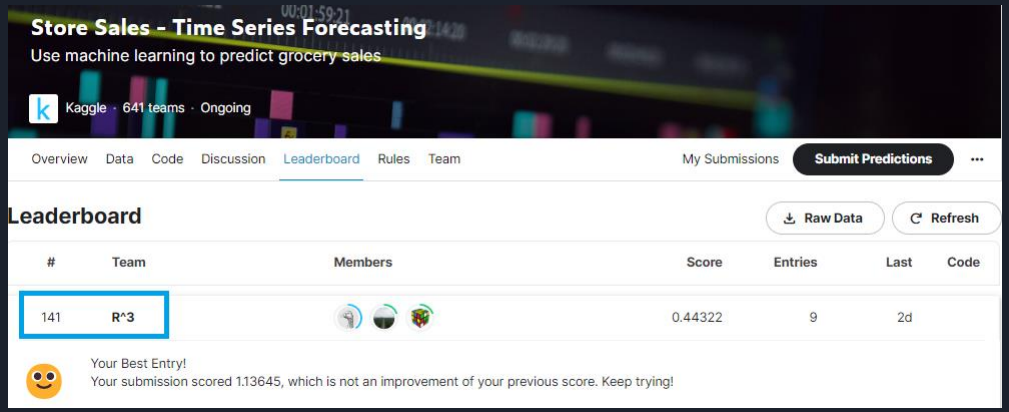
```
[ ]  1 # Cambiamos la forma de organizar nuestra tabla
     2 y = trainfinal2.unstack(['store_nbr','family']).loc[sdate:edate]
```

```
[ ]  1 #Formación de modelo
     2 fourier = CalendarFourier(freq='W', order=4)
     3
     4 dp = DeterministicProcess(index=y.index,
     5                           constant=False,
     6                           order=1,
     7                           seasonal=False,
     8                           additional_terms=[fourier],
     9                           drop=True)
    10
    11 X = dp.in_sample()
    12
    13 # Modelo Ridge
    14 model = Ridge(fit_intercept=True, solver='auto', alpha=0.4, normalize=True, random_state=SEED)
    15 model.fit(X, y)
    16 y_pred = pd.DataFrame(model.predict(X), index=X.index, columns=y.columns)
```

C) KAGGLE: Store Sales - Time Series Forecasting

6. Predicción y resultado final

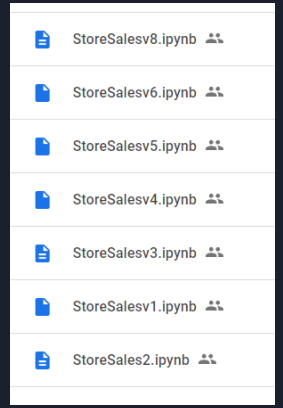
- LEADERBOARD



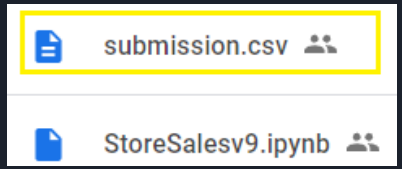
Posición Equipo R^3: **141 de 641 (<25%)** (Fecha: 28/06/2022)

Hemos hecho varias cargas de datasets,
con variaciones de los siguientes notebooks:

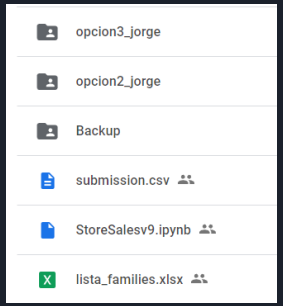
- StoreSalesv1.ipynb
- StoreSalesv2.ipynb
- StoreSalesv3.ipynb
- StoreSalesv4.ipynb
- StoreSalesv5.ipynb
- StoreSalesv6.ipynb
- StoreSalesv8.ipynb



- **StoreSalesv9.ipynb //submission.csv**



D) Dificultades y Conclusiones

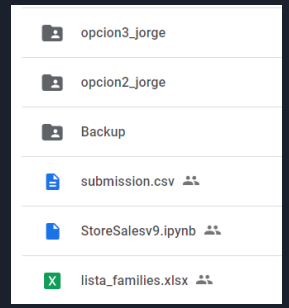


- **DIFICULTADES**

- **HOUSE PRICES:**
- Al principio nuestro posicionamiento no fue tan bueno
- Depurando bien los datos y ajustando los parámetros, conseguimos mejorar nuestro puesto
- Fue un prueba - error constante, hasta mejorar
- Total submissions: 42

- **STORE SALES:**
- Fichero con muchos datos y muchas variables
- Nos resultó complicado encontrar el modelo de predicción de las futuras ventas de las tiendas
- Intentamos las varias opciones mencionadas antes
- Una vez detectado el mejor modelo, la aplicación en sí sencilla
- Total submissions: 8

D) Dificultades y Conclusiones



- **CONCLUSIONES**

- **HOUSE PRICES**
- es importante entrenar el modelo hasta tal momento que se consigue el mejor resultado
- Importante la depuración de los datos y ajustar parámetros
- Kaggle es una plataforma idónea para aprender, ya que mediante el ranking se recibe feedback inmediato de los nuevos datos enviados y motiva para conseguir mejores resultados

- STORE SALES
- En esta competición descubrimos la librería PANDA PROFILING
- Es un método muy útil que recomendamos utilizar, ya que facilita una visión clara del dataset indicando todos los valores más importantes, como máximo y mínimo, promedio, etc.

Para finalizar ...

... Últimas palabras de la presentación:

Nos ha gustado probar la plataforma Kaggle, que es la comunidad Data Science más grande del mundo, con más de 536 mil miembros activos en 194 países. Hay muchas publicaciones, que brindan todas las herramientas, el foro y recursos más importantes para progresar al máximo en data science y nos ha encantado conocerla y poder practicar en este entorno interactivo para aprender a sacarle valor a los datos mediante las nuevas tecnologías y el machine learning. GRACIAS.

¡ MUCHAS GRACIAS POR LA ATENCIÓN !

RAUL PULIDO, RACHELE RENDINA, RODRIGO ANCHELERGUES