Projeto2_ver4

May 20, 2020

1 Data Science Academy

2 Big Data Real-Time Analytics com Python e Spark

3 Capítulo 6

4 Machine Learning em Python - Parte 2 - Regressão

[2]:
```
from IPython.display import Image
Image(url = 'images/processo.png')
```

[2]:
```
<IPython.core.display.Image object>
```

[1]:
```
import sklearn as sl
import warnings
warnings.filterwarnings("ignore")
sl.__version__
```

[1]:
```
'0.21.3'
```

5 XGBRegressor

[70]:
```
# Import dos módulos
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_log_error
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

# Carregando os dados

dados = read_csv("Train_demanda_media")
array = dados.values
```

```
# Separando o array em componentes de input e output
X = array[:,0:5]
Y = array[:,5]

scaler = StandardScaler()
X = scaler.fit_transform(X)

# Divide os dados em treino e teste
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)

# Criando o modelo
modelo = XGBRegressor()

# Treinando o modelo
modelo.fit(X_train, Y_train)

# Fazendo previsões
Y_pred = modelo.predict(X_test)

# Como algumas previsões no Y_pred são negativas e o MSLE não permite valores␣
 ↪negativos no seu cálculo,
# consideramos 0 para toda previsão negativa e arredondamos o valor de saída

Y_pred = np.array(Y_pred)

def zeraneg(i):
    if i < 0:
        return 0
    else:
        return i

Y_pred = list(map(zeraneg, Y_pred))

Y_pred = [round(value) for value in Y_pred]

# Resultado
msle = mean_squared_log_error(Y_test, Y_pred)
print("O MSLE do modelo é:", msle)
```

```
[12:27:14] WARNING: C:\Users\Administrator\workspace\xgboost-
win64_release_1.0.0\src\gbm\gbtree.cc:138: Tree method is automatically selected
to be 'approx' for faster speed. To use old behavior (exact greedy algorithm on
single machine), set tree_method to 'exact'.
O MSLE do modelo é: 0.3982225796934346
```

[72]:
```
# Salvando o modelo
import pickle
```

```
arquivo = 'modelo_regressor_final.sav'
pickle.dump(modelo, open(arquivo, 'wb'))
print("Modelo salvo!")
```

```
Modelo salvo!
```

6 Fazendo as previsões

[78]:
```
dados = read_csv("test.csv")
array = dados.values
X = array[:,2:7]
```

[81]:
```
dados = pd.DataFrame(X)
dados.head()
```

[81]:

	0	1	2	3	4
0	4037	1	2209	4639078	35305
1	2237	1	1226	4705135	1238
2	2045	1	2831	4549769	32940
3	1227	1	4448	4717855	43066
4	1219	1	1130	966351	1277

[82]:
```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

[83]:
```
dados = pd.DataFrame(X)
dados.head()
```

[83]:

	0	1	2	3	4
0	0.382157	-0.265543	0.047312	0.959523	0.702846
1	-0.066695	-0.265543	-0.607851	0.981999	-1.119098
2	-0.114573	-0.265543	0.461870	0.929134	0.576363
3	-0.318551	-0.265543	1.539589	0.986327	1.117914
4	-0.320546	-0.265543	-0.671834	-0.290168	-1.117012

[84]:
```
# Fazendo previsões
Y_pred = modelo.predict(X)
```

[85]:
```
Previsoes = pd.DataFrame(Y_pred)
Previsoes.head()
```

[85]:

	0
0	3.047385
1	8.949769
2	4.514654
3	2.740568

```
4  4.680477
```

[86]:
```
Previsoes.min()
```

[86]:
```
0   -56.099327
dtype: float32
```

[87]:
```
Y_pred = np.array(Y_pred)

def zeraneg(i):
    if i < 0:
        return 0
    else:
        return i

Y_pred = list(map(zeraneg, Y_pred))

Y_pred = [int(round(value)) for value in Y_pred]
```

[94]:
```
Previsoes = pd.DataFrame(Y_pred)
Previsoes.head()
```

[94]:
```
   0
0  3
1  9
2  5
3  3
4  5
```

[100]:
```
Previsoes.to_csv('Sample_submission_PSTT.csv')
```

[103]:
```
Previsoes.min()
Previsoes.max()
```

[103]:
```
0    1671
dtype: int64
```

7 Fim

7.0.1 Obrigado - Data Science Academy - facebook.com/dsacademybr