Прус Ева, ИУ5-64Б

РК №2, Вариант 16

Ввод [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn import preprocessing
from sklearn import svm
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error, mean_squared_error
```

Ввод [5]:
```
data = pd.read_csv('/Users/eva/Documents/Учеба/jupnote/restaurant-scores-lives-standard.csv')
data
```

Out[5]:

...ess_city	business_state	business_postal_code	business_latitude	business_longitude	business_location	business_phone_number	...	inspection_type	
Francisco	CA	NaN	NaN	NaN	NaN	1.415043e+10	...	New Ownership	
Francisco	CA	94118	NaN	NaN	NaN	1.415724e+10	...	Routine - Unscheduled	97975_20190
Francisco	CA	94110	NaN	NaN	NaN	NaN	...	New Ownership	
Francisco	CA	94111	NaN	NaN	NaN	1.415488e+10	...	New Construction	
Francisco	CA	94109	NaN	NaN	NaN	NaN	...	New Ownership	85986_20161
...	...	...	...	...	...	...	...	...	
Francisco	CA	94107	NaN	NaN	NaN	NaN	...	Routine - Unscheduled	89569_20190
Francisco	CA	94132	NaN	NaN	NaN	NaN	...	New Ownership - Followup	
Francisco	CA	94105	NaN	NaN	NaN	NaN	...	Routine - Unscheduled	84541_20190
Francisco	CA	94112	NaN	NaN	NaN	NaN	...	Routine - Unscheduled	91572_20190
Francisco	CA	94107	NaN	NaN	NaN	NaN	...	Routine - Unscheduled	89569_20190

Ввод [6]:
```
data.keys().to_list()
```

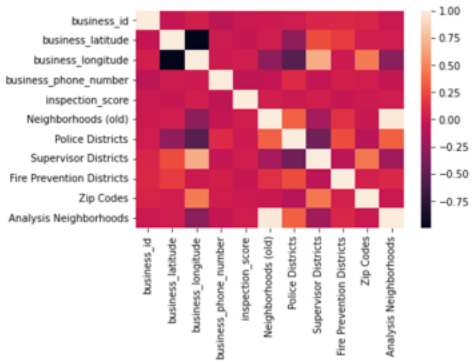
Out[6]:
```
['business_id',
 'business_name',
 'business_address',
 'business_city',
 'business_state',
 'business_postal_code',
 'business_latitude',
 'business_longitude',
 'business_location',
 'business_phone_number',
 'inspection_id',
 'inspection_date',
 'inspection_score',
 'inspection_type',
 'violation_id',
 'violation_description',
 'risk_category',
 'Neighborhoods (old)',
 'Police Districts',
 'Supervisor Districts',
 'Fire Prevention Districts',
 'Zip Codes',
 'Analysis Neighborhoods']
```

Ввод [7]:
```
data.isna().sum()
```

Out[7]:
```
business_id                 0
business_name               0
business_address            0
business_city               0
business_state              0
business_postal_code     1018
business_latitude       19556
business_longitude      19556
business_location       19556
business_phone_number   36938
inspection_id               0
inspection_date             0
inspection_score        13610
inspection_type             0
violation_id            12870
violation_description   12870
risk_category           12870
Neighborhoods (old)     19594
Police Districts        19594
Supervisor Districts    19594
Fire Prevention Districts    19646
Zip Codes               19576
Analysis Neighborhoods  19594
dtype: int64
```

Ввод [11]:
```
corr_data = data.corr()
sns.heatmap(corr_data)
```

Out[11]: <AxesSubplot:>



Ввод [12]:
```
data = data[(data['Police Districts'].isna() == False)]
```

Ввод [13]:
```
data.isna().sum()
```

Out[13]:
```
business_id                   0
business_name                 0
business_address              0
business_city                 0
business_state                0
business_postal_code        392
business_latitude             0
business_longitude            0
business_location             0
business_phone_number     25150
inspection_id                 0
inspection_date               0
inspection_score           7262
inspection_type               0
violation_id               7232
violation_description      7232
risk_category              7232
Neighborhoods (old)           0
Police Districts              0
```

Ввод [18]:
```
data.pop('business_postal_code')
data.pop('business_phone_number')
data.pop('violation_id')
data.pop('violation_description')
data.pop('risk_category')
```

Out[18]:
```
11            Low Risk
16       Moderate Risk
30                NaN
55            Low Risk
64       Moderate Risk
             ...
53850             NaN
53851       High Risk
53852             NaN
53853    Moderate Risk
53854       High Risk
Name: risk_category, Length: 34379, dtype: object
```

Ввод [20]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34379 entries, 11 to 53854
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   business_id                34379 non-null  int64
 1   business_name              34379 non-null  object
 2   business_address           34379 non-null  object
 3   business_city              34379 non-null  object
 4   business_state             34379 non-null  object
 5   business_latitude          34379 non-null  float64
 6   business_longitude         34379 non-null  float64
 7   business_location          34379 non-null  object
 8   inspection_id              34379 non-null  object
 9   inspection_date            34379 non-null  object
 10  inspection_score           34379 non-null  float64
 11  inspection_type            34379 non-null  object
 12  Neighborhoods (old)        34379 non-null  float64
 13  Police Districts           34379 non-null  float64
 14  Supervisor Districts       34379 non-null  float64
 15  Fire Prevention Districts  34327 non-null  float64
 16  Zip Codes                  34379 non-null  float64
 17  Analysis Neighborhoods     34379 non-null  float64
dtypes: float64(9), int64(1), object(8)
memory usage: 5.0+ MB
```

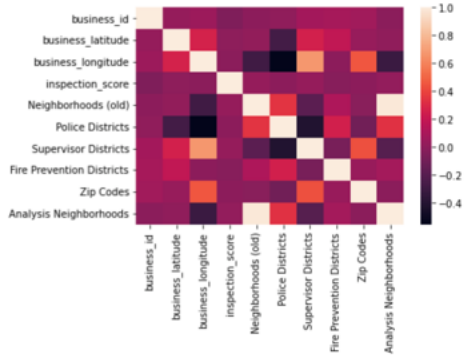
Ввод [22]:
```
data1 = data.dropna(subset=['Fire Prevention Districts'], inplace=True)
```

Ввод [24]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34327 entries, 11 to 53854
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   business_id                34327 non-null  int64  
 1   business_name              34327 non-null  object 
 2   business_address           34327 non-null  object 
 3   business_city              34327 non-null  object 
 4   business_state             34327 non-null  object 
 5   business_latitude          34327 non-null  float64
 6   business_longitude         34327 non-null  float64
 7   business_location          34327 non-null  object 
 8   inspection_id              34327 non-null  object 
 9   inspection_date            34327 non-null  object 
 10  inspection_score           34327 non-null  float64
 11  inspection_type            34327 non-null  object 
 12  Neighborhoods (old)        34327 non-null  float64
 13  Police Districts           34327 non-null  float64
 14  Supervisor Districts       34327 non-null  float64
 15  Fire Prevention Districts  34327 non-null  float64
 16  Zip Codes                  34327 non-null  float64
 17  Analysis Neighborhoods     34327 non-null  float64
dtypes: float64(9), int64(1), object(8)
memory usage: 5.0+ MB
```

Ввод [25]:
```
corr_data = data.corr()
sns.heatmap(corr_data)
```

Out[25]:
```
<AxesSubplot:>
```



Ввод [26]:
```
X_data = data[['business_latitude','business_longitude','Neighborhoods (old)','Supervisor Districts','Fire Prevention D
                ]]
Y_data = data['Police Districts'].to_list()
```

Ввод [27]:
```
X_data=preprocessing.normalize(X_data,axis = 0)
```

Ввод [28]:
```
X_data.shape
```

Out[28]:
```
(34327, 6)
```

Ввод [29]:
```
X_train, X_test, y_train, y_test = train_test_split(X_data,
                                                    Y_data, test_size=0.5,
                                                    random_state=42)
```

Ввод [30]:
```
from sklearn.linear_model import LogisticRegression
```

Ввод [31]:
```
model_logistic = LogisticRegression()
model_logistic.fit(X_train,y_train)
```

Out[31]:
```
LogisticRegression()
```

Ввод [32]:
```
targ_logistic = model_logistic.predict(X_test)
```

Были выбраны такие метрики как MSE,MAPE,MAE, как самые подходящие для логистической регрессии

Ввод [34]:
```
mae = mean_absolute_error(y_test,targ_logistic)
mape = mean_absolute_percentage_error(y_test,targ_logistic)
mse = mean_squared_error(y_test,targ_logistic)
print('MAE '+str(round(mae,3)) + ' MAPE ' + str(round(mape,3)) + ' MSE ' + str(round(mse,3)) )
```

```
MAE 3.824 MAPE 0.624 MSE 23.39
```

Ввод [36]:
```
import sys
!{sys.executable} -m pip install xgboost
from xgboost import XGBRegressor
```

```
Collecting xgboost
  Downloading xgboost-1.6.1-py3-none-macosx_10_15_x86_64.macosx_11_0_x86_64.macosx_12_0_x86_64.whl (1.7 MB)
     |████████████████████████████████| 1.7 MB 179 kB/s eta 0:00:01
Requirement already satisfied: scipy in /Users/eva/opt/anaconda3/lib/python3.9/site-packages (from xgboost) (1.7.1)
Requirement already satisfied: numpy in /Users/eva/opt/anaconda3/lib/python3.9/site-packages (from xgboost) (1.20.3)
Installing collected packages: xgboost
Successfully installed xgboost-1.6.1
```

Ввод [37]:
```
XGB_model = XGBRegressor()
mape = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_absolute_percentage_error').mean()
mae = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_absolute_error').mean()
mse = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_squared_error').mean()
print('MAE '+str(round(mae,3)) + ' MAPE ' + str(round(mape,3)) + ' MSE ' + str(round(mse,3)) )
```

Ввод [37]:
```
XGB_model = XGBRegressor()
mape = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_absolute_percentage_error').mean()
mae = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_absolute_error').mean()
mse = -cross_val_score(XGB_model,X_train,y_train,cv=4,scoring = 'neg_mean_squared_error').mean()
print('MAE '+str(round(mae,3)) + ' MAPE ' + str(round(mape,3)) + ' MSE ' + str(round(mse,3)) )
```

```
MAE 0.009 MAPE 0.002 MSE 0.007
```

Ввод [38]:
```
XGB_model.fit(X_train,y_train)
mae = mean_absolute_error(y_test,XGB_model.predict(X_test))
mape = mean_absolute_percentage_error(y_test,XGB_model.predict(X_test))
mse = mean_squared_error(y_test,XGB_model.predict(X_test))
print('MAE '+str(round(mae,3)) + ' MAPE ' + str(round(mape,3)) + ' MSE ' + str(round(mse,3)) )
```

```
MAE 0.008 MAPE 0.003 MSE 0.011
```

Ввод [ ]:
На основе трех метрик можно сказать, что либо я где-то по пути ошиблась, либо Градиетный брустинг очень хорошо справился с задачей