Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

Отчет по рубежному контролю №2

Вариант 19

По дисциплине:
«Технологии машинного обучения»

Выполнил:

Студент группы ИУ5 _____ Тарновский Д.Р.

(Подпись, дата) (Фамилия И.О.)

Проверил:

_____ **Гапанюк Ю. Е.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2022

Задача

Для заданного набора данных (по Вашему варианту) постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы 1 и 2 (по варианту для Вашей группы). Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик). Какие метрики качества Вы использовали и почему? Какие выводы Вы можете сделать о качестве построенных моделей? Для построения моделей необходимо выполнить требуемую предобработку данных: заполнение пропусков, кодирование категориальных признаков, и т.д.

ИУ5-63Б, ИУ5Ц-83Б	Дерево решений	Случайный лес

Набор данных

https://www.kaggle.com/rhuebner/human-resources-data-set

Тарновский Д.Р. ИУ5-63Б РК №2

Импорт библиотек

Ввод [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
warnings.filterwarnings('ignore')
sns.set(style="ticks")
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error,  median_absolute_error, r2_score
```

Ввод [2]:
```
data = pd.read_csv('HRDataset_v14.csv', sep = ',' )
data = data.fillna(0)
```

Ввод [3]:
```
data.head()
```

Out[3]:

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	Salary	...	ManagerName	Manag
0	Adinolfi, Wilson K	10026	0	0	1	1	5	4	0	62506	...	Michael Albert	
1	Ait Sidi, Karthikeyan	10084	1	1	1	5	3	3	0	104437	...	Simon Roup	
2	Akinkuolie, Sarah	10196	1	1	0	5	5	3	0	64955	...	Kissy Sullivan	
3	Alagbe,Trina	10088	1	1	0	1	5	3	0	64991	...	Elijiah Gray	
4	Anderson, Carol	10069	0	2	0	5	5	3	0	50825	...	Webster Butler	

5 rows × 36 columns

Ввод [4]:
```
data.dtypes
```

Out[4]:
```
Employee_Name                object
EmpID                         int64
MarriedID                     int64
MaritalStatusID               int64
GenderID                      int64
EmpStatusID                   int64
DeptID                        int64
PerfScoreID                   int64
FromDiversityJobFairID        int64
Salary                        int64
Termd                         int64
PositionID                    int64
Position                     object
State                        object
Zip                           int64
DOB                          object
Sex                          object
MaritalDesc                  object
CitizenDesc                  object
HispanicLatino               object
RaceDesc                     object
DateofHire                   object
DateofTermination            object
TermReason                   object
EmploymentStatus             object
Department                   object
ManagerName                  object
ManagerID                   float64
RecruitmentSource            object
PerformanceScore             object
EngagementSurvey            float64
EmpSatisfaction               int64
SpecialProjectsCount          int64
LastPerformanceReview_Date   object
DaysLateLast30                int64
Absences                      int64
dtype: object
```

Ввод [5]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[5]:
```
Employee_Name                0
EmpID                        0
MarriedID                    0
MaritalStatusID              0
GenderID                     0
EmpStatusID                  0
DeptID                       0
PerfScoreID                  0
FromDiversityJobFairID       0
Salary                       0
Termd                        0
PositionID                   0
Position                     0
State                        0
Zip                          0
DOB                          0
Sex                          0
MaritalDesc                  0
CitizenDesc                  0
HispanicLatino               0
RaceDesc                     0
DateofHire                   0
DateofTermination            0
TermReason                   0
EmploymentStatus             0
Department                   0
ManagerName                  0
ManagerID                    0
RecruitmentSource            0
PerformanceScore             0
EngagementSurvey             0
EmpSatisfaction              0
SpecialProjectsCount         0
LastPerformanceReview_Date   0
DaysLateLast30               0
Absences                     0
dtype: int64
```

Ввод [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 311 entries, 0 to 310
Data columns (total 36 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Employee_Name               311 non-null    object
 1   EmpID                       311 non-null    int64
 2   MarriedID                   311 non-null    int64
 3   MaritalStatusID             311 non-null    int64
 4   GenderID                    311 non-null    int64
 5   EmpStatusID                 311 non-null    int64
 6   DeptID                      311 non-null    int64
 7   PerfScoreID                 311 non-null    int64
 8   FromDiversityJobFairID      311 non-null    int64
 9   Salary                      311 non-null    int64
 10  Termd                       311 non-null    int64
 11  PositionID                  311 non-null    int64
 12  Position                    311 non-null    object
 13  State                       311 non-null    object
 14  Zip                         311 non-null    int64
 15  DOB                         311 non-null    object
 16  Sex                         311 non-null    object
 17  MaritalDesc                 311 non-null    object
 18  CitizenDesc                 311 non-null    object
 19  HispanicLatino              311 non-null    object
 20  RaceDesc                    311 non-null    object
 21  DateofHire                  311 non-null    object
 22  DateofTermination           311 non-null    object
 23  TermReason                  311 non-null    object
 24  EmploymentStatus            311 non-null    object
 25  Department                  311 non-null    object
 26  ManagerName                 311 non-null    object
 27  ManagerID                   311 non-null    float64
 28  RecruitmentSource           311 non-null    object
 29  PerformanceScore            311 non-null    object
 30  EngagementSurvey            311 non-null    float64
 31  EmpSatisfaction             311 non-null    int64
 32  SpecialProjectsCount        311 non-null    int64
 33  LastPerformanceReview_Date  311 non-null    object
 34  DaysLateLast30              311 non-null    int64
 35  Absences                    311 non-null    int64
dtypes: float64(2), int64(16), object(18)
memory usage: 87.6+ KB
```

Ввод [7]:
```
data.head()
```

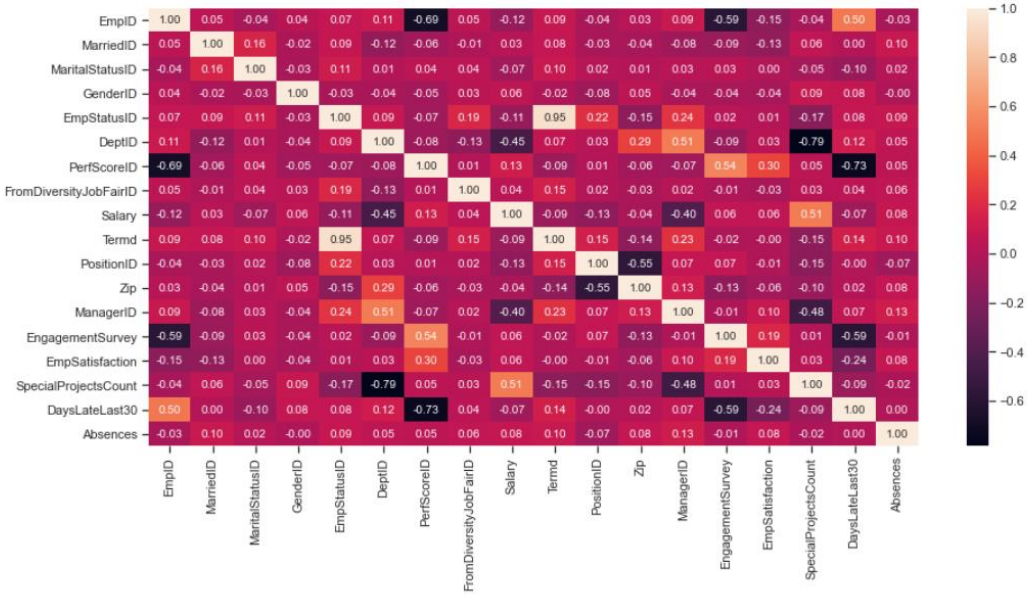
Out[7]:

	Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairID	Salary	...	ManagerName	Manag
0	Adinolfi, Wilson K	10026	0	0	1	1	5	4	0	62506	...	Michael Albert	
1	Ait Sidi, Karthikeyan	10084	1	1	1	5	3	3	0	104437	...	Simon Roup	
2	Akinkuolie, Sarah	10196	1	1	0	5	5	3	0	64955	...	Kissy Sullivan	
3	Alagbe,Trina	10088	1	1	0	1	5	3	0	64991	...	Elijiah Gray	
4	Anderson, Carol	10069	0	2	0	5	5	3	0	50825	...	Webster Butler	

5 rows × 36 columns

Ввод [8]:
```
#Построим корреляционную матрицу
fig, ax = plt.subplots(figsize=(15,7))
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, fmt='.2f')
```

Out[8]: <AxesSubplot:>



Ввод [9]:
```
X = data[["PerfScoreID","EngagementSurvey"]]
Y = data.EmpID
print('Входные данные:\n\n', X.head(), '\n\nВыходные данные:\n\n', Y.head())
```

```
Входные данные:

    PerfScoreID  EngagementSurvey
0            4              4.60
1            3              4.96
2            3              3.02
3            3              4.84
4            3              5.00

Выходные данные:

 0    10026
1    10084
2    10196
3    10088
4    10069
Name: EmpID, dtype: int64
```

Ввод [10]:
```
X_train,  X_test,  Y_train,  Y_test = train_test_split(X,  Y, random_state = 0, test_size = 0.1)
print('Входные параметры обучающей выборки:\n\n',X_train.head(), \
      '\n\nВходные параметры тестовой выборки:\n\n', X_test.head(), \
      '\n\nВыходные параметры обучающей выборки:\n\n', Y_train.head(), \
      '\n\nВыходные параметры тестовой выборки:\n\n', Y_test.head())
```

```
Входные параметры обучающей выборки:

      PerfScoreID  EngagementSurvey
167            4              4.10
291            3              4.50
189            3              3.70
259            3              3.80
254            3              3.32

Входные параметры тестовой выборки:

      PerfScoreID  EngagementSurvey
212            4              5.00
146            3              4.00
225            3              4.80
129            3              3.03
89             3              3.13

Выходные параметры обучающей выборки:

 167    10017
291    10253
189    10213
259    10153
254    10178
Name: EmpID, dtype: int64

Выходные параметры тестовой выборки:

 212    10005
146    10138
225    10225
129    10195
89     10189
Name: EmpID, dtype: int64
```

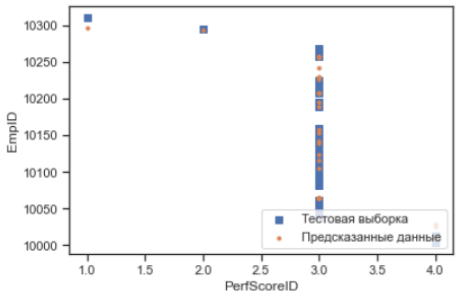
Ввод [11]:
```
from sklearn.ensemble import RandomForestRegressor
```

Ввод [12]:
```
forest_1 = RandomForestRegressor(n_estimators=5, oob_score=True, random_state=10)
forest_1.fit(X_train,Y_train)
```

Out[12]:
```
RandomForestRegressor(n_estimators=5, oob_score=True, random_state=10)
```

Ввод [13]:
```
pred_y = forest_1.predict(X_test)
```

Ввод [14]:
```
plt.scatter(X_test.PerfScoreID, Y_test,    marker = 's', label = 'Тестовая выборка')
plt.scatter(X_test.PerfScoreID, pred_y, marker = '.', label = 'Предсказанные данные')
plt.legend (loc = 'lower right')
plt.xlabel ('PerfScoreID')
plt.ylabel ('EmpID')
plt.show()
```



Ввод [15]:
```
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, export_graphviz
from sklearn.tree import export_graphviz
from sklearn import tree
import re
```

Ввод [16]:
```
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, Y)
lr_y_pred = clf.predict(X_test)
```

Ввод [17]:
```
plt.scatter(X_test.PerfScoreID, Y_test,    marker = 's', label = 'Тестовая выборка')
plt.scatter(X_test.PerfScoreID, lr_y_pred, marker = 'o', label = 'Предсказанные данные')
plt.legend (loc = 'lower right')
plt.xlabel ('PerfScoreID')
plt.ylabel ('EmpID')
plt.show()
```