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)

Рубежный контроль №2

по курсу «Технологии машинного обучения»

Вариант 19

Выполнил

студент группы ИУ5-64Б

Шпак И.Д.

Москва, 2021

1 Исходное задание

Задание.

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.

Группа	Метод 1	Метод 2
ИУ5-64Б	Линейная/логистическая регрессия	Градиентный бустинг

2 Исходный код

[1]:
```
from IPython.display import Image
import numpy as np
import pandas as pd
from sklearn import svm, datasets
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, cross_validate
from sklearn.model_selection import KFold, RepeatedKFold
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
import seaborn as sns
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.linear_model import RidgeClassifier, SGDClassifier
from sklearn.linear_model import Ridge,ElasticNet
from sklearn.preprocessing import OrdinalEncoder, StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingClassifier
```

```
from sklearn.datasets import load_wine
from sklearn.metrics import precision_score, recall_score, f1_score,␣
 ↪classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from io import StringIO
%matplotlib inline
pd.set_option("display.max_rows", None, "display.max_columns", None)
sns.set(style="ticks")
```

2.0.1 Загрузка данных

[2]:
```
data = pd.read_csv("/home/igor/     /mllabs/rk2/HRDataset_v14.
 ↪csv",sep=',')
data.shape
data["RaceDesc"].unique()
```

[2]:
```
array(['White', 'Black or African American', 'Two or more races',␣
 ↪'Asian',
       'American Indian or Alaska Native', 'Hispanic'], dtype=object)
```

2.0.2 Предварительная обработка данных

[3]:
```
data.
 ↪drop(["Employee_Name","EmpID","Position","LastPerformanceReview_Date","CitizenDesc
data = data.dropna(axis = 1 , how = "any")
```

2.0.3 Анализ полученного датасета

[15]:
```
#     5
data.head()
```

[15]:

	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	\
0	0	1	1	5	4	
1	1	1	5	3	3	
2	1	0	5	5	3	

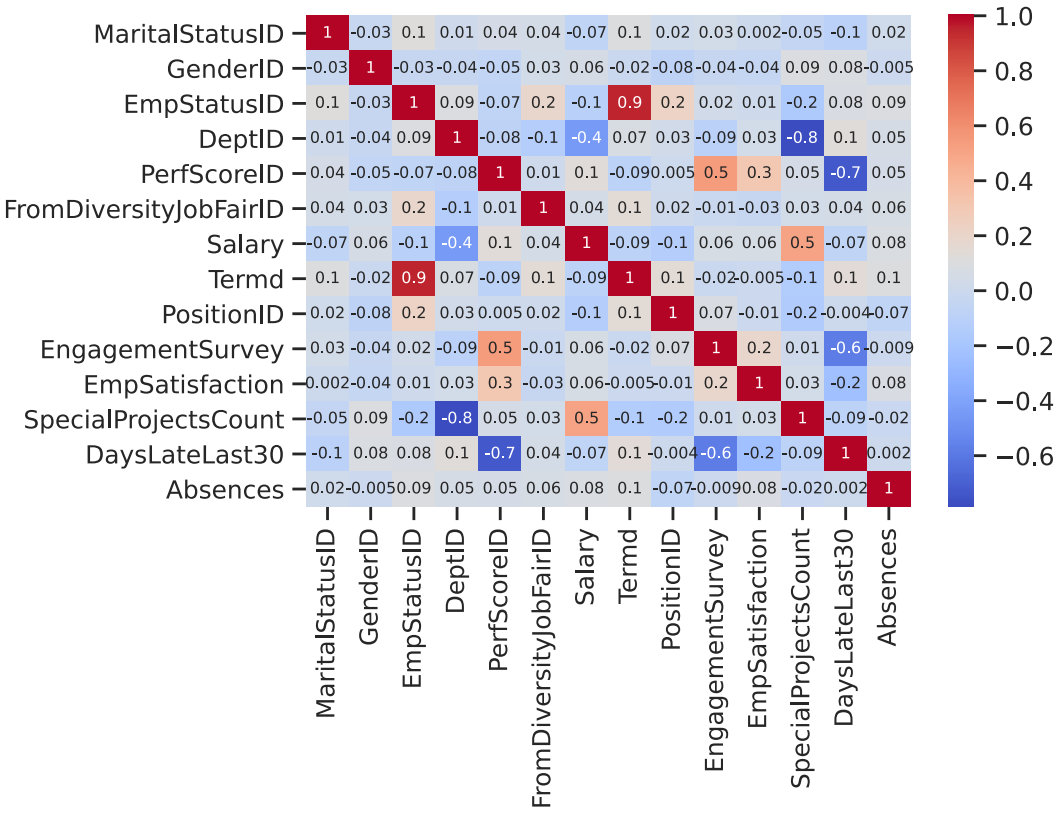
```
3                  1          0             1        5             3
4                  2          0             5        5             3

   FromDiversityJobFairID  Salary  Termd  PositionID  EngagementSurvey ␣
↪\
0                       0   62506      0          19              4.60
1                       0  104437      1          27              4.96
2                       0   64955      1          20              3.02
3                       0   64991      0          19              4.84
4                       0   50825      1          19              5.00

   EmpSatisfaction  SpecialProjectsCount  DaysLateLast30  Absences
0                5                     0               0         1
1                3                     6               0        17
2                3                     0               0         3
3                5                     0               0        15
4                4                     0               0         2
```

[4]:
```
#
sns.heatmap(data.corr(),annot = True,fmt='.1g', cmap=␣
↪'coolwarm',annot_kws={"size":7})
```

[4]:
```
<AxesSubplot:>
```



2.0.4 Разбиение датасета на тестовую и тренировочную выборки

[5]:
```
data_Y = data.loc[:, 'PerfScoreID']
data_X = data.drop(["PerfScoreID"],axis=1,inplace=False)
data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(
    data_X, data_Y,test_size=0.35, random_state=360)
```

2.0.5 Обучение моделей

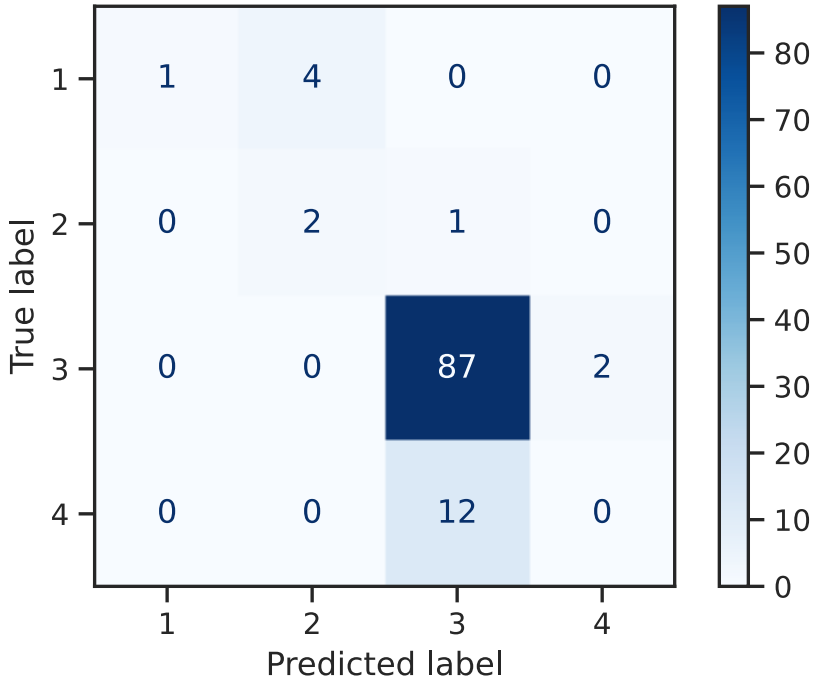
В качестве метрик качества рассматриваемых моделей будем использовать accuracy (как наглядную численную оценку) и матрицу ошибок, чтобы понять распределение ошибок по классам.

[18]:
```
cl1 = RidgeClassifier()
cl1.fit(data_X_train,data_y_train)
target1_0 = cl1.predict(data_X_train)
target1_1 = cl1.predict(data_X_test)
accuracy_score(data_y_train, target1_0), accuracy_score(data_y_test,␣
 ↪target1_1)
```

[18]: (0.8316831683168316, 0.8256880733944955)

[65]:
```
confusion_matrix(data_y_test, target1_1, labels=[1, 2,3 ,4])
plot_confusion_matrix(cl1, data_X_test, data_y_test,
                      display_labels=[1, 2,3 ,4], cmap=plt.cm.Blues)
```

[65]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f2333e144c0>



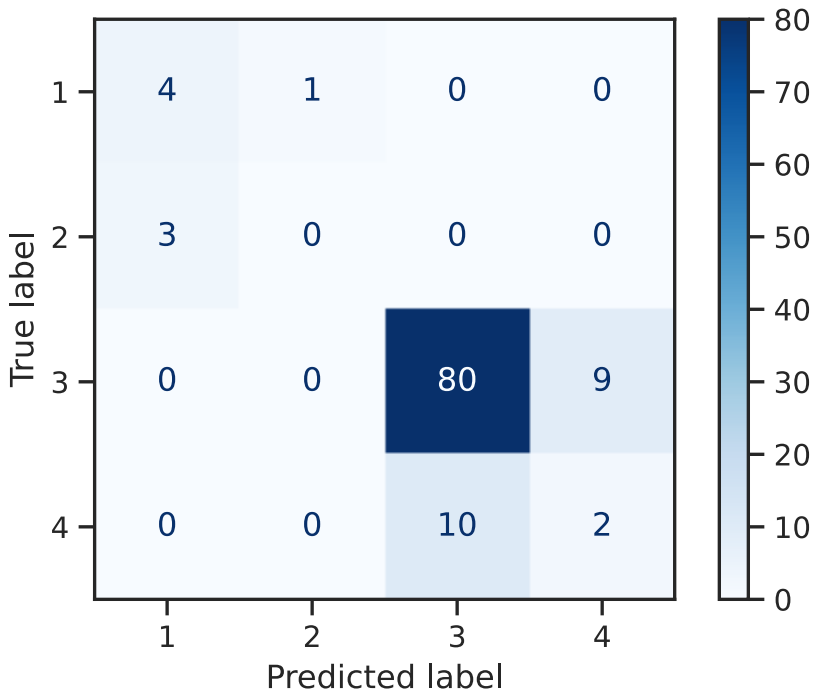
[23]:
```
cl2 =  GradientBoostingClassifier()
cl2.fit(data_X_train,data_y_train)
target2_0 = cl2.predict(data_X_train)
target2_1 = cl2.predict(data_X_test)
accuracy_score(data_y_train, target2_0), accuracy_score(data_y_test,␣
↪target2_1)
```

[23]: (1.0, 0.7889908256880734)

[67]:
```
confusion_matrix(data_y_test, target1_1, labels=[1, 2,3 ,4])
plot_confusion_matrix(cl2, data_X_test, data_y_test,
```

```
                    display_labels=[1, 2,3 ,4], cmap=plt.cm.Blues)
```

[67]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f23338dbb50>



В качестве вывода об оценке качества моделей можно указать, что обе модели показали схожие результаты, а ошибки вызванные неверным определением некоторых классов могут быть вызваны неравномерностью распределения классов в датасете или при разбиении датасета на тренировочную и тестовую выборки