

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2022 г.

"__" _____ 2022 г.

**Отчет по лабораторной работе № 4 по курсу
Технологии машинного обучения
ГУИМЦ**

Тема работы: " Линейные модели, SVM и деревья решений. "

11

(количество листов)

Вариант № 16

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-63Б

Наказной Н.А.

(подпись)

"__" _____ 2022 г.

Москва, МГТУ - 2022

Цель лабораторной работы:

изучение линейных моделей, SVM и деревьев решений.

Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую.
4. Обучите следующие модели:
 - одну из линейных моделей (линейную или полиномиальную регрессию при решении задачи регрессии, логистическую регрессию при решении задачи классификации);
 - SVM;
 - дерево решений.
5. Оцените качество моделей с помощью двух подходящих для задачи метрик. Сравните качество полученных моделей.
6. Постройте график, показывающий важность признаков в дереве решений.
7. Визуализируйте дерево решений или выведите правила дерева решений в текстовом виде.

Ход выполнения работы

В качестве набора данных используется dataset - Результаты студентов на экзаменах (Оценки, полученные учащимися по различным предметам).

Датасет доступен по адресу:

<https://www.kaggle.com/datasets/spscientist/students-performance-in-exams>

Из набора данных будет рассматриваться только файл « StudentsPerformance.csv»

Лабораторная работа №4

```
In [1]: import numpy as np
import pandas as pd
from typing import Dict, Tuple
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.impute import SimpleImputer
import warnings
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
from sklearn.linear_model import LinearRegression
warnings.simplefilter("ignore")
```

```
In [2]: # чтение обучающей выборки
data = pd.read_csv('StudentsPerformance.csv')
data.head()
```

```
Out[2]:
```

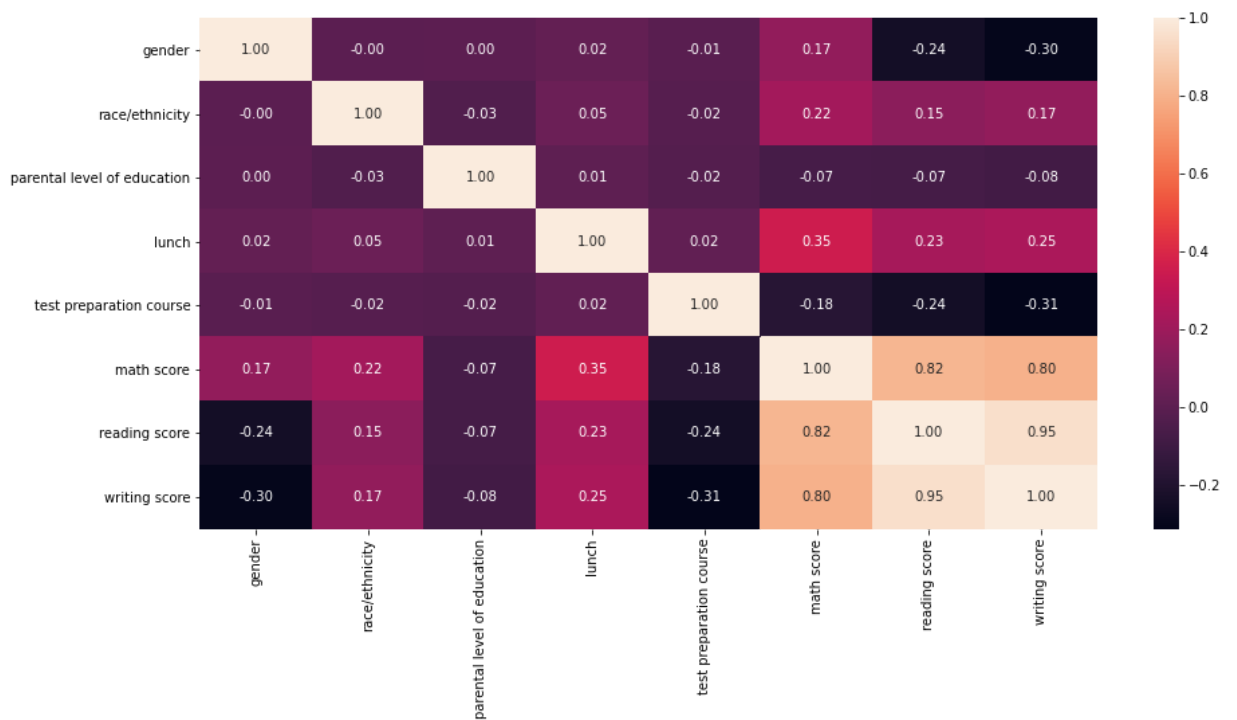
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

```
In [3]: from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

```
In [4]: le = LabelEncoder()
# "gender" - пол
le.fit(data.gender.drop_duplicates())
data.gender = le.transform(data.gender)
# "race/ethnicity" - раса
le.fit(data["race/ethnicity"].drop_duplicates())
data["race/ethnicity"] = le.transform(data["race/ethnicity"])
# "Lunch" - обед
le.fit(data.lunch.drop_duplicates())
data.lunch = le.transform(data.lunch)
# "parental level of education" - образование родителей
le.fit(data["parental level of education"].drop_duplicates())
data["parental level of education"] = le.transform(data["parental level of education"])
# "test preparation course" - подготовительный курс
le.fit(data["test preparation course"].drop_duplicates())
data["test preparation course"] = le.transform(data["test preparation course"])
```

```
In [5]: #Построим корреляционную матрицу
fig, ax = plt.subplots(figsize=(15,7))
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, fmt='.2f')
```

```
Out[5]: <AxesSubplot:>
```



Предскажем значения поля Writing score по Math score и Reading score, так как значение корреляции ближе всего к 1.

```
In [6]: X = data[["math score", "reading score"]]
Y = data["writing score"]
print('Входные данные:\n\n', X.head(), '\n\nВыходные данные:\n\n', Y.head())
```

Входные данные:

	math score	reading score
0	72	72
1	69	90
2	90	95
3	47	57
4	76	78

Выходные данные:

0	74
1	88
2	93
3	44
4	75

Name: writing score, dtype: int64

```
In [7]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state = 0, test
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:

	math score	reading score
785	32	51
873	90	90
65	67	64
902	34	48
317	83	72

Входные параметры тестовой выборки:

math score	reading score
------------	---------------

993	62	72
859	87	73
298	40	46
553	77	62
672	69	78

Выходные параметры обучающей выборки:

785	44
873	82
65	61
902	41
317	78

Name: writing score, dtype: int64

Выходные параметры тестовой выборки:

993	74
859	72
298	50
553	64
672	76

Name: writing score, dtype: int64

Построение линейной регрессии

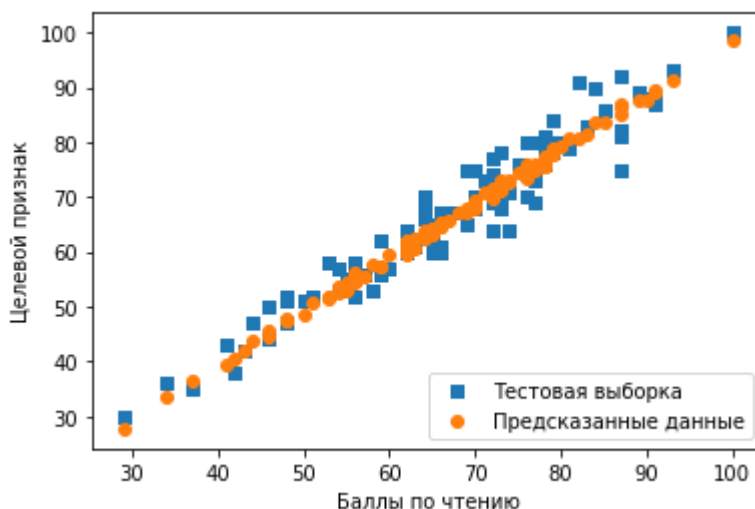
```
In [8]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error
```

```
In [9]: Lin_Reg = LinearRegression().fit(X_train, Y_train)

lr_y_pred = Lin_Reg.predict(X_test)
```

Возьмем тот параметр, чья корреляция ближе всего к единице, т.е. Reading score

```
In [10]: plt.scatter(X_test["reading score"], Y_test, marker = 's', label = 'Тестовая выбо
plt.scatter(X_test["reading score"], lr_y_pred, marker = 'o', label = 'Предсказанные
plt.legend (loc = 'lower right')
plt.xlabel ('Баллы по чтению')
plt.ylabel ('Целевой признак')
plt.show()
```



SVM

```
In [11]: from sklearn.svm import SVC , LinearSVC
```

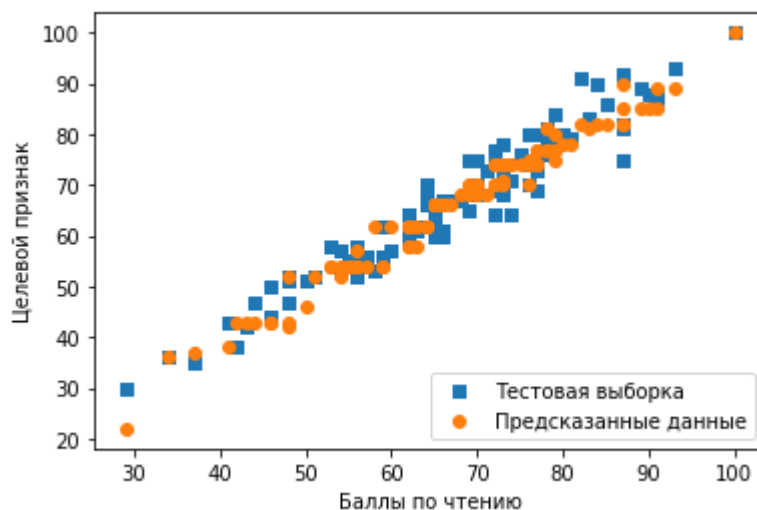
```
from sklearn.datasets.samples_generator import make_blobs
from matplotlib import pyplot as plt
```

```
In [12]: svc = SVC(kernel='linear')
svc.fit(X_train, Y_train)
```

```
Out[12]: SVC(kernel='linear')
```

```
In [13]: pred_y = svc.predict(X_test)
```

```
In [14]: plt.scatter(X_test["reading score"], Y_test, marker = 's', label = 'Тестовая выбо
plt.scatter(X_test["reading score"], pred_y, marker = 'o', label = 'Предсказанные да
plt.legend (loc = 'lower right')
plt.xlabel ('Баллы по чтению')
plt.ylabel ('Целевой признак')
plt.show()
```



Tree

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

```
In [16]: # Обучим дерево на всех признаках iris
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_test, Y_test)
```

```
In [17]: from IPython.core.display import HTML
from sklearn.tree.export import export_text
tree_rules = export_text(clf, feature_names=list(X.columns))
HTML('<pre>' + tree_rules + '</pre>')
```

```
Out[17]: |--- reading score <= 61.00
|       |--- math score <= 64.50
|       |   |--- math score <= 53.00
|       |   |   |--- reading score <= 56.50
|       |   |   |   |--- math score <= 46.50
|       |   |   |   |   |--- reading score <= 31.50
|       |   |   |   |   |   |--- class: 30
|       |   |   |   |   |   |--- reading score > 31.50
|       |   |   |   |   |   |--- math score <= 36.00
|       |   |   |   |   |   |   |--- class: 43
|       |   |   |   |   |   |   |--- math score > 36.00
```

```

|--- reading score <= 38.00
|   |--- class: 36
|   |--- reading score > 38.00
|       |--- reading score <= 42.50
|           |--- class: 38
|           |--- reading score > 42.50
|               |--- math score <= 41.00
|                   |--- class: 50
|                   |--- math score > 41.00
|                       |--- math score <= 43.00
|                           |--- class: 54
|                           |--- math score > 43.00
|                               |--- truncated branch of depth 2
|--- math score > 46.50
|   |--- reading score <= 45.00
|       |--- class: 35
|       |--- reading score > 45.00
|           |--- reading score <= 53.50
|               |--- class: 58
|               |--- reading score > 53.50
|                   |--- math score <= 47.50
|                       |--- class: 53
|                       |--- math score > 47.50
|                           |--- class: 58
|--- reading score > 56.50
|   |--- reading score <= 58.00
|       |--- class: 56
|       |--- reading score > 58.00
|           |--- class: 56
|--- math score > 53.00
|   |--- reading score <= 49.50
|       |--- math score <= 58.50
|           |--- math score <= 54.50
|               |--- class: 52
|               |--- math score > 54.50
|                   |--- reading score <= 47.00
|                       |--- class: 44
|                       |--- reading score > 47.00
|                           |--- class: 51
|                   |--- math score > 58.50
|                       |--- class: 47
|--- reading score > 49.50
|   |--- math score <= 63.00
|       |--- reading score <= 53.00
|           |--- class: 52
|           |--- reading score > 53.00
|               |--- class: 55
|       |--- math score > 63.00
|           |--- class: 52
|--- math score > 64.50
|   |--- reading score <= 57.00
|       |--- class: 57
|       |--- reading score > 57.00
|           |--- reading score <= 59.00
|               |--- class: 53
|               |--- reading score > 59.00
|                   |--- class: 57
|--- reading score > 61.00

```

```

|--- reading score <= 68.50
|   |--- reading score <= 66.50
|       |--- math score <= 69.50
|           |--- math score <= 60.00
|               |--- reading score <= 62.50
|                   |--- math score <= 45.50
|                       |--- class: 61
|                           |--- math score > 45.50
|                               |--- math score <= 53.50
|                                   |--- class: 60
|                                       |--- math score > 53.50
|                                           |--- class: 64
|--- reading score > 62.50
|   |--- reading score <= 63.50
|       |--- class: 62
|   |--- reading score > 63.50
|       |--- math score <= 55.00
|           |--- reading score <= 64.50
|               |--- class: 68
|                   |--- reading score > 64.50
|                       |--- class: 65
|                           |--- math score > 55.00
|                               |--- class: 63
|--- math score > 60.00
|   |--- reading score <= 62.50
|       |--- class: 60
|   |--- reading score > 62.50
|       |--- reading score <= 65.50
|           |--- class: 61
|       |--- reading score > 65.50
|           |--- math score <= 62.00
|               |--- class: 61
|               |--- math score > 62.00
|                   |--- class: 67
|--- math score > 69.50
|   |--- reading score <= 64.50
|       |--- reading score <= 63.00
|           |--- class: 64
|       |--- reading score > 63.00
|           |--- math score <= 73.00
|               |--- class: 70
|               |--- math score > 73.00
|                   |--- class: 66
|--- reading score > 64.50
|   |--- class: 60
|--- reading score > 66.50
|   |--- class: 67
|--- reading score > 68.50
|   |--- reading score <= 70.50
|       |--- reading score <= 69.50
|           |--- math score <= 63.00
|               |--- class: 65
|           |--- math score > 63.00
|               |--- class: 75
|--- reading score > 69.50
|   |--- math score <= 71.50
|       |--- math score <= 59.00
|           |--- math score <= 55.50

```



```

|--- class: 70
|--- math score > 55.50
|--- class: 68
|--- math score > 59.00
|--- class: 70
|--- math score > 71.50
|--- class: 75
|--- reading score > 70.50
|--- reading score <= 81.50
|--- reading score <= 77.50
|--- math score <= 76.50
|--- math score <= 74.50
|--- reading score <= 75.00
|--- math score <= 57.50
|--- class: 64
|--- math score > 57.50
|--- math score <= 62.50
|--- class: 74
|--- math score > 62.50
|--- reading score <= 72.50
|--- class: 77
|--- reading score > 72.50
|--- truncated branch of depth 3
|--- reading score > 75.00
|--- math score <= 54.50
|--- class: 70
|--- math score > 54.50
|--- reading score <= 76.50
|--- class: 80
|--- reading score > 76.50
|--- math score <= 61.00
|--- class: 80
|--- math score > 61.00
|--- truncated branch of depth 2
|--- math score > 74.50
|--- class: 68
|--- math score > 76.50
|--- reading score <= 74.00
|--- reading score <= 72.50
|--- math score <= 79.50
|--- class: 69
|--- math score > 79.50
|--- class: 73
|--- reading score > 72.50
|--- class: 72
|--- reading score > 74.00
|--- reading score <= 75.50
|--- class: 76
|--- reading score > 75.50
|--- reading score <= 76.50
|--- class: 74
|--- reading score > 76.50
|--- class: 73
|--- reading score > 77.50
|--- math score <= 73.50
|--- math score <= 58.00
|--- class: 79
|--- math score > 58.00

```

```

|--- reading score <= 78.50
|   |--- class: 76
|--- reading score > 78.50
|   |--- class: 79
|--- math score > 73.50
|   |--- math score <= 87.50
|       |--- reading score <= 78.50
|           |--- class: 81
|       |--- reading score > 78.50
|           |--- reading score <= 79.50
|               |--- math score <= 76.50
|                   |--- class: 80
|                   |--- math score > 76.50
|                       |--- class: 78
|                   |--- reading score > 79.50
|                       |--- class: 80
|       |--- math score > 87.50
|           |--- math score <= 91.50
|               |--- class: 79
|           |--- math score > 91.50
|               |--- class: 84
|--- reading score > 81.50
|   |--- reading score <= 89.50
|       |--- math score <= 74.00
|           |--- reading score <= 85.00
|               |--- class: 83
|           |--- reading score > 85.00
|               |--- class: 82
|       |--- math score > 74.00
|           |--- math score <= 78.00
|               |--- class: 91
|           |--- math score > 78.00
|               |--- math score <= 84.50
|                   |--- reading score <= 87.00
|                       |--- class: 86
|                   |--- reading score > 87.00
|                       |--- class: 89
|               |--- math score > 84.50
|                   |--- math score <= 91.50
|                       |--- class: 75
|                   |--- math score > 91.50
|                       |--- reading score <= 85.50
|                           |--- class: 90
|                       |--- reading score > 85.50
|                           |--- math score <= 96.50
|                               |--- class: 92
|                           |--- math score > 96.50
|                               |--- class: 81
|--- reading score > 89.50
|   |--- math score <= 79.00
|       |--- class: 88
|   |--- math score > 79.00
|       |--- math score <= 81.50
|           |--- class: 87
|       |--- math score > 81.50
|           |--- math score <= 87.00
|               |--- class: 93

```

```
| | | | | | | | |--- math score > 87.00  
| | | | | | | | |--- class: 100
```

```
In [18]: pred_y = clf.predict(X_test)  
plt.scatter(X_test["reading score"], Y_test, marker = 's', label = 'Тестовая выбо  
plt.scatter(X_test["reading score"], pred_y, marker = 'o', label = 'Предсказанные да  
plt.legend (loc = 'lower right')  
plt.xlabel ('Баллы по чтению')  
plt.ylabel ('Целевой признак')  
plt.show()
```

