

Рубежный контроль №2

Запруднов М.С. Группа ИУ5-62Б

Вариант 9

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.

Методы для ИУ5-62Б. Метод №1: "Метод опорных векторов". Метод №2: "Случайный лес".

Набор данных: [Houses to rent](#).

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('houses_to_rent.csv')
```

Ввод [3]:

```
data.head()
```

Out[3]:

	Unnamed: 0	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa	rent amount	property tax	fire insurance	total
0	0	1	240	3	3	4	-	accept	furnished	R\$0	R\$8,000	R\$1,000	R\$121	R\$9,121
1	1	0	64	2	1	1	10	accept	not furnished	R\$540	R\$820	R\$122	R\$11	R\$1,493
2	2	1	443	5	5	4	3	accept	furnished	R\$4,172	R\$7,000	R\$1,417	R\$89	R\$12,680
3	3	1	73	2	2	1	12	accept	not furnished	R\$700	R\$1,250	R\$150	R\$16	R\$2,116
4	4	1	19	1	1	0	-	not accept	not furnished	R\$0	R\$1,200	R\$41	R\$16	R\$1,257

Ввод [4]:

```
data.dtypes
```

Out[4]:

```
Unnamed: 0      int64
city            int64
area            int64
rooms           int64
bathroom        int64
parking spaces  int64
floor           object
animal          object
furniture       object
hoa             object
rent amount     object
property tax    object
fire insurance  object
total           object
dtype: object
```

Ввод [5]:

```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[5]:

```
Unnamed: 0      0
city            0
area            0
rooms           0
bathroom        0
parking spaces  0
floor           0
animal          0
furniture       0
hoa             0
rent amount     0
property tax    0
fire insurance  0
total           0
dtype: int64
```

Ввод [6]: data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6080 entries, 0 to 6079
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0             6080 non-null   int64
1   city                   6080 non-null   int64
2   area                   6080 non-null   int64
3   rooms                  6080 non-null   int64
4   bathroom               6080 non-null   int64
5   parking spaces         6080 non-null   int64
6   floor                  6080 non-null   object
7   animal                 6080 non-null   object
8   furniture               6080 non-null   object
9   hoa                    6080 non-null   object
10  rent amount            6080 non-null   object
11  property tax           6080 non-null   object
12  fire insurance         6080 non-null   object
13  total                  6080 non-null   object
dtypes: int64(6), object(8)
memory usage: 665.1+ KB
```

Ввод [7]: data.head()

Out[7]:

	Unnamed: 0	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa	rent amount	property tax	fire insurance	total
0	0	1	240	3	3	4	-	accept	furnished	R\$0	R\$8,000	R\$1,000	R\$121	R\$9,121
1	1	0	64	2	1	1	10	accept	not furnished	R\$540	R\$820	R\$122	R\$11	R\$1,493
2	2	1	443	5	5	4	3	accept	furnished	R\$4,172	R\$7,000	R\$1,417	R\$89	R\$12,680
3	3	1	73	2	2	1	12	accept	not furnished	R\$700	R\$1,250	R\$150	R\$16	R\$2,116
4	4	1	19	1	1	0	-	not accept	not furnished	R\$0	R\$1,200	R\$41	R\$16	R\$1,257

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.drop(['Unnamed: 0', 'city', 'area', 'floor', 'animal', 'furniture', 'hoa', 'rent amount', 'property tax', 'fire insurance', 'total'])`
`Y = data.bathroom`
`print('Входные данные:\n', X.head(), '\n\nВыходные данные:\n\n', Y.head())`

Входные данные:

	rooms	bathroom	parking spaces
0	3	3	4
1	2	1	1
2	5	5	4
3	2	2	1
4	1	1	0

Выходные данные:

0	3
1	1
2	5
3	2
4	1

Name: bathroom, 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:

	rooms	bathroom	parking	spaces
6065	2	2		1
2207	1	1		1
1277	2	1		0
3847	3	2		2
1299	2	1		1

Входные параметры тестовой выборки:

	rooms	bathroom	parking	spaces
1430	1	1		0
3875	2	1		1
5399	3	1		0
394	2	2		1
2179	2	2		1

Выходные параметры обучающей выборки:

```
6065    2
2207    1
1277    1
3847    2
1299    1
Name: bathroom, dtype: int64
```

Выходные параметры тестовой выборки:

```
1430    1
3875    1
5399    1
394     2
2179    2
Name: bathroom, dtype: int64
```

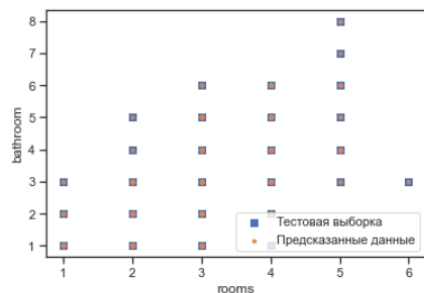
```
Ввод [11]: from sklearn.svm import SVC, LinearSVC
from sklearn.datasets.samples_generator import make_blobs
from matplotlib import pyplot as plt
```

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

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

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

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



```
Ввод [15]: from sklearn.ensemble import RandomForestRegressor
```

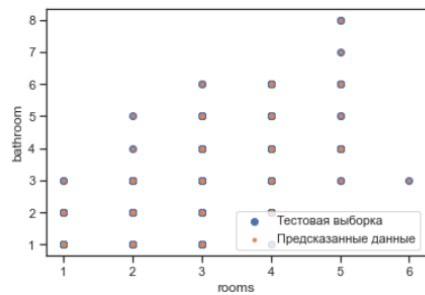
```
Ввод [16]: forest_1 = RandomForestRegressor(n_estimators=5, oob_score=True, random_state=10)
forest_1.fit(X, Y)
```

```
Out[16]: RandomForestRegressor(n_estimators=5, oob_score=True, random_state=10)
```

```
Ввод [17]: Y_predict = forest_1.predict(X_test)
print('Средняя абсолютная ошибка:', mean_absolute_error(Y_test, Y_predict))
print('Средняя квадратичная ошибка:', mean_squared_error(Y_test, Y_predict))
print('Median absolute error:', median_absolute_error(Y_test, Y_predict))
print('Коэффициент детерминации:', r2_score(Y_test, Y_predict))
```

Средняя абсолютная ошибка: 0.0
Средняя квадратичная ошибка: 0.0
Median absolute error: 0.0
Коэффициент детерминации: 1.0

```
Ввод [18]: plt.scatter(X_test.rooms, Y_test, marker = 'o', label = 'Тестовая выборка')
plt.scatter(X_test.rooms, Y_predict, marker = '.', label = 'Предсказанные данные')
plt.legend(loc = 'lower right')
plt.xlabel('rooms')
plt.ylabel('bathroom')
plt.show()
```



Вывод. Обе модели обучились хорошо, и предсказали значения с точностью 100%.