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 ИУ5 «Системы обработки информации и управления»

Отчёт по рубежному контролю №2

«Технологии машинного обучения»

Вариант 9

Выполнила:

студентка группы ИУ5-63Б

Пересыпкина М.А.

Преподаватель:

Гапанюк Ю. Е.

2023 г.

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

https://www.kaggle.com/datasets/rubenssjr/brasilian-houses-to-rent

Решение:

Загружаем датасет и подключаем необходимые библиотеки:

Ввод [23]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
```

Ввод [24]:
```
#Загрузка датасета
data = pd.read_csv("houses_to_rent_v2.csv")
```

Ввод [25]:
```
data.head()
```

Out[25]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836

Проверка типов данных:

Ввод [26]:
```
data.dtypes
```

Out[26]:
```
city                  object
area                   int64
rooms                  int64
bathroom               int64
parking spaces         int64
floor                 object
animal                object
furniture             object
hoa (R$)               int64
rent amount (R$)       int64
property tax (R$)      int64
fire insurance (R$)    int64
total (R$)             int64
dtype: object
```

Посчитаем количество пустых значений:

Ввод [28]:
```
data.isnull().sum()
```

Out[28]:
```
city                  0
area                  0
rooms                 0
bathroom              0
parking spaces        0
floor                 0
animal                0
furniture             0
hoa (R$)              0
rent amount (R$)      0
property tax (R$)     0
fire insurance (R$)   0
total (R$)            0
dtype: int64
```

Ограничим данные до 500 строк:

Ввод [29]:
```
data = data.head(500)
```

Ввод [30]:
```
data.head()
```

Out[30]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836

Удалим некоторые столбцы со стоимостями, оставим только total.

Ввод [33]:
```
cols_drop = ['hoa (R$)', 'rent amount (R$)','property tax (R$)', 'fire insurance (R$)']
data = data.drop(cols_drop, axis = 1)
```

Кодирование категориальных признаков:

Ввод [34]:
```
#Кодирование категориальных признаков
LE = LabelEncoder()
for col in data.columns:
    if data[col].dtype == "object":
        data[col] = LE.fit_transform(data[col])
```

Ввод [35]:
```
data.dtypes
```

Out[35]:
```
city              int32
area              int64
rooms             int64
bathroom          int64
parking spaces    int64
floor             int32
animal            int32
furniture         int32
total (R$)        int64
dtype: object
```

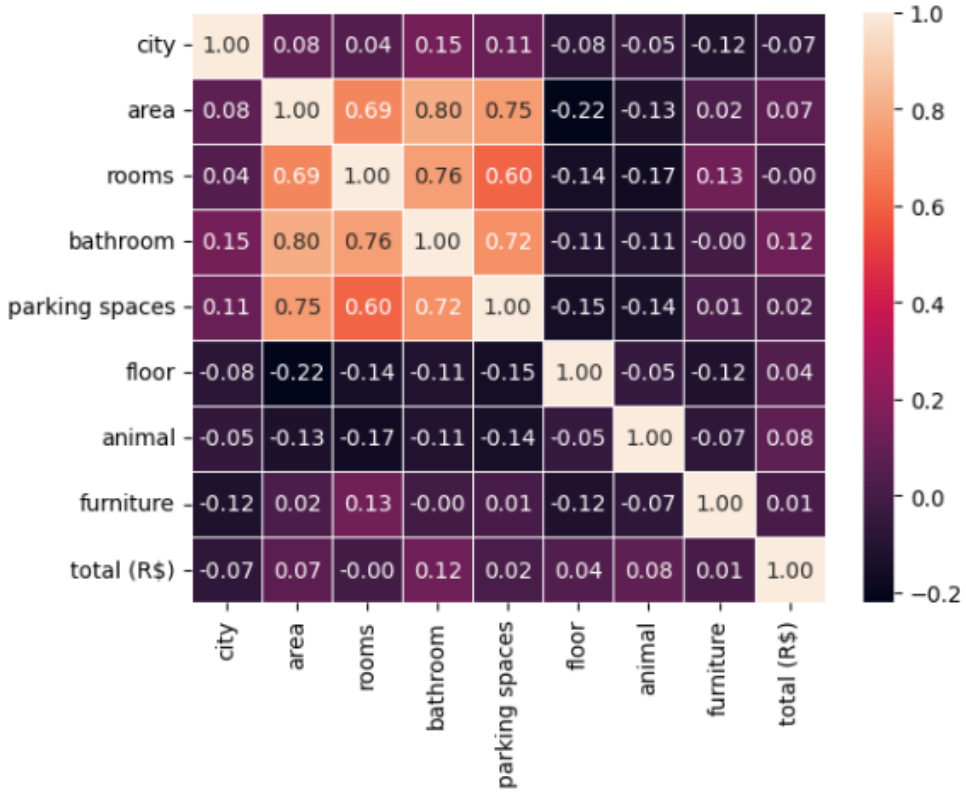
Ввод [36]:
```
data.head()
```

Out[36]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	total (R$)
0	4	70	2	1	1	24	0	0	5618
1	4	320	4	4	0	13	0	1	7973
2	2	80	1	1	1	23	0	1	3841
3	2	51	2	1	0	12	0	1	1421
4	4	25	1	1	0	1	1	1	836

Построим матрицу корреляции

Ввод [38]:
```
corr = data.corr()
sns.heatmap(corr, linewidths=.5, annot=True, fmt=".2f")
plt.show()
```



Разделение выборки на обучающую и тестовую

Ввод [39]:
```
target = "total (R$)"
xArray = data.drop(target, axis=1)
yArray = data[target]
trainX, testX, trainY, testY = train_test_split(xArray, yArray, test_size=0.2, random_state=1)
```

Дерево решений

Ввод [40]:
```
regressor = DecisionTreeRegressor()
regressor.fit(trainX, trainY)
```

Out[40]:
```
DecisionTreeRegressor
DecisionTreeRegressor()
```

Для оценки качества будем использовать:

- коэффициента детерминации, чтобы узнать насколько модель близка к высококачественной

- корень из средней квадратичной ошибки, чтобы выделить большие ошибки в предсказании модели

Ввод [41]:
```
R2_LR = r2_score(testY, regressor.predict(testX))
RMSE_LR = mean_squared_error(testY, regressor.predict(testX), squared=True)
```

Ввод [42]:
```
print("Оценка качества модели с помощью коэффициента детерминации: {}".format(R2_LR))
print("Корень из средней квадратичной ошибки: {}".format(RMSE_LR))
```

```
Оценка качества модели с помощью коэффициента детерминации: 0.11015622171725337
Корень из средней квадратичной ошибки: 16683714.96
```

Случайный лес

Ввод [43]:
```
RT = RandomForestRegressor(n_estimators=10, random_state=1)
RT.fit(trainX, trainY)
```

Out[43]:
```
RandomForestRegressor
RandomForestRegressor(n_estimators=10, random_state=1)
```

Ввод [44]:
```
R2_RT = r2_score(testY, RT.predict(testX))
```

Ввод [45]:
```
RMSE_RT= mean_squared_error(testY, RT.predict(testX), squared=True)
```

Ввод [46]:
```
print("Оценка качества модели с помощью коэффициента детерминации: {}".format(R2_RT))
print("Корень из средней квадратичной ошибки: {}".format(RMSE_RT))
```

```
Оценка качества модели с помощью коэффициента детерминации: 0.5557813433702838
Корень из средней квадратичной ошибки: 8328672.5468000015
```

Вывод: ансамблевая модель случайного леса предсказывает значения с большей точностью в отличие от модели дерева решений. Мы видим более низкий показатель RMSE и более высокий R2.