Рубежный контроль №2

Никулин Данила ИУ5-61Б Вариант 10

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 группы ИУ5-61Б:

1. Линейная/логистическая регрессия
2. Случайный лес

Набор данных:

https://www.kaggle.com/rubenssjr/brasilian-houses-to-rent (файл houses_to_rent_v2.csv)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, LabelEncoder,
StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from sklearn.impute import SimpleImputer

data=pd.read_csv('data/houses_to_rent_v2.csv',sep=",")

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10692 entries, 0 to 10691
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
```

```
 0   city                  10692 non-null  object
 1   area                  10692 non-null  int64
 2   rooms                 10692 non-null  int64
 3   bathroom              10692 non-null  int64
 4   parking spaces        10692 non-null  int64
 5   floor                 10692 non-null  object
 6   animal                10692 non-null  object
 7   furniture             10692 non-null  object
 8   hoa (R$)              10692 non-null  int64
 9   rent amount (R$)      10692 non-null  int64
 10  property tax (R$)     10692 non-null  int64
 11  fire insurance (R$)   10692 non-null  int64
 12  total (R$)            10692 non-null  int64
dtypes: int64(9), object(4)
memory usage: 1.1+ MB
```

```
data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 10692,\n  \"fields\": [\n    {\n      \"column\": \"city\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 5,\n        \"samples\": [\n          \"Porto Alegre\",\n          \"Belo Horizonte\",\n          \"Rio de Janeiro\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"area\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 537,\n        \"min\": 11,\n        \"max\": 46335,\n        \"num_unique_values\": 517,\n        \"samples\": [\n          255,\n          503,\n          474\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"rooms\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 1,\n        \"max\": 13,\n        \"num_unique_values\": 11,\n        \"samples\": [\n          5,\n          2,\n          13\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"bathroom\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 1,\n        \"max\": 10,\n        \"num_unique_values\": 10,\n        \"samples\": [\n          8,\n          4,\n          5\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"parking spaces\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 0,\n        \"max\": 12,\n        \"num_unique_values\": 11,\n        \"samples\": [\n          6,\n          1,\n          10\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"floor\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 35,\n        \"samples\": [\n          \"23\",\n          \"17\",\n          \"22\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"animal\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"not acept\",\n
\"acept\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"furniture\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"not furnished\",\n          \"furnished\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"hoa (R$)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 15592,\n
\"min\": 0,\n        \"max\": 1117000,\n        \"num_unique_values\":
1679,\n        \"samples\": [\n          4799,\n          472\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"rent amount (R$)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3408,\n        \"min\": 450,\n        \"max\": 45000,\n
\"num_unique_values\": 1195,\n        \"samples\": [\n          3820,\
n          8250\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"property tax (R$)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3107,\n        \"min\": 0,\n
\"max\": 313700,\n        \"num_unique_values\": 1243,\n
\"samples\": [\n          2482,\n          1323\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"fire insurance (R$)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
47,\n        \"min\": 3,\n        \"max\": 677,\n
\"num_unique_values\": 216,\n        \"samples\": [\n          184,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total (R$)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 16484,\n        \"min\": 499,\n
\"max\": 1120000,\n        \"num_unique_values\": 5751,\n
\"samples\": [\n          2377,\n          5115\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"data"}
```


```
numeric_elements = data['floor'].replace('-',
np.nan).apply(pd.to_numeric, errors='coerce')
mean_value = int(numeric_elements.mean(skipna=True))
data['floor'] = data['floor'].replace('-',
str(mean_value)).astype(int)

data.isnull().sum()
```

```
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
```

Кодирование и преобразоване категориальных данных

```
data['animal'] = LabelEncoder().fit_transform(data['animal'])
data['furniture'] = LabelEncoder().fit_transform(data['furniture'])

ohe = OneHotEncoder()
city_encoded = ohe.fit_transform(data[['city']]).toarray()
data = pd.concat([data, pd.DataFrame(city_encoded,
columns=ohe.get_feature_names_out())], axis=1)
data.drop('city', axis=1, inplace=True)

data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 10692,\n
\"fields\": [\n    {\n      \"column\": \"area\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
537,\n        \"min\": 11,\n        \"max\": 46335,\n
\"num_unique_values\": 517,\n        \"samples\": [\n          255,\n
503,\n          474\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"rooms\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 1,\n        \"max\": 13,\n
\"num_unique_values\": 11,\n        \"samples\": [\n          5,\n
2,\n          13\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"bathroom\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 1,\n
\"max\": 10,\n        \"num_unique_values\": 10,\n        \"samples\":
[\n          8,\n          4,\n          5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"parking spaces\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 12,\n
\"num_unique_values\": 11,\n        \"samples\": [\n          6,\n
1,\n          10\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"floor\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5,\n        \"min\": 1,\n        \"max\": 301,\n
```

```
\"num_unique_values\": 34,\n        \"samples\": [\n          13,\n
26,\n          25\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"animal\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"furniture\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"hoa (R$)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 15592,\n        \"min\": 0,\n
\"max\": 1117000,\n        \"num_unique_values\": 1679,\n
\"samples\": [\n          4799,\n          472\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"rent amount (R$)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3408,\n        \"min\": 450,\n        \"max\": 45000,\n
\"num_unique_values\": 1195,\n        \"samples\": [\n          3820,\
n          8250\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"property tax (R$)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3107,\n        \"min\": 0,\n
\"max\": 313700,\n        \"num_unique_values\": 1243,\n
\"samples\": [\n          2482,\n          1323\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"fire insurance (R$)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
47,\n        \"min\": 3,\n        \"max\": 677,\n
\"num_unique_values\": 216,\n        \"samples\": [\n          184,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total (R$)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 16484,\n        \"min\": 499,\n
\"max\": 1120000,\n        \"num_unique_values\": 5751,\n
\"samples\": [\n          2377,\n          5115\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"city_Belo Horizonte\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.32221786572987327,\n        \"min\": 0.0,\n        \"max\": 1.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1.0,\n
0.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"city_Campinas\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.2709638510504819,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
```

```
\"samples\": [\n          1.0,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"city_Porto Alegre\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.31486220017176947,\n        \"min\": 0.0,\n        \"max\": 1.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1.0,\n
0.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"city_Rio de Janeiro\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.347402619939056,\n        \"min\":
0.0,\n        \"max\": 1.0,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1.0,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"city_S\\u00e3o Paulo\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.49745645894613316,\n        \"min\": 0.0,\n        \"max\": 1.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0.0,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```

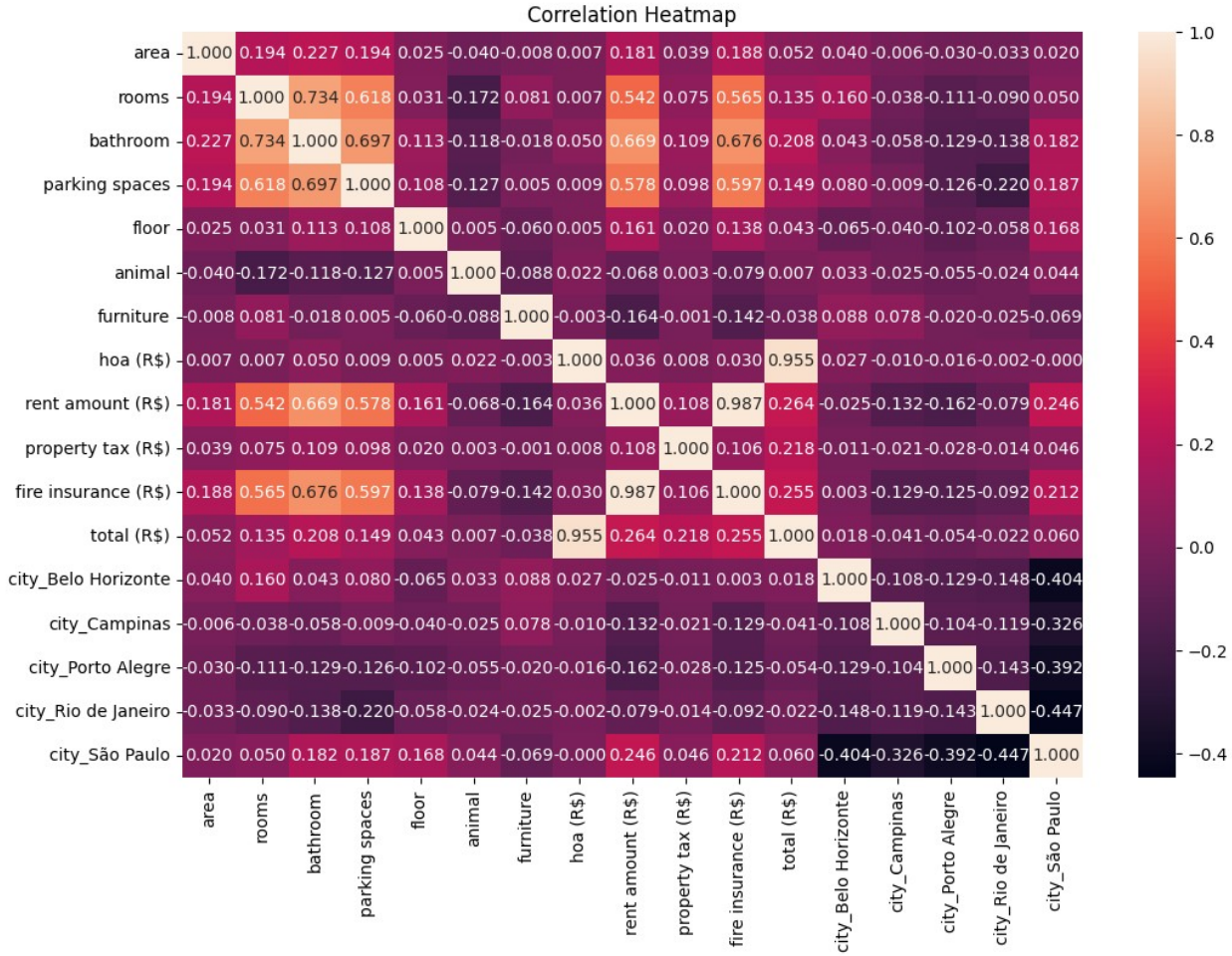

```
data.dtypes
```

```
area                          int64
rooms                         int64
bathroom                      int64
parking spaces                int64
floor                         int64
animal                        int64
furniture                     int64
hoa (R$)                      int64
rent amount (R$)              int64
property tax (R$)             int64
fire insurance (R$)           int64
total (R$)                    int64
city_Belo Horizonte         float64
city_Campinas               float64
city_Porto Alegre           float64
city_Rio de Janeiro         float64
city_São Paulo              float64
dtype: object
```

```
# Масштабирование данных
scaler = StandardScaler()
data = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)

# Построение тепловой карты корреляции
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(method='pearson'), annot=True, fmt='.3f')
```

```
plt.title('Correlation Heatmap')
plt.show()
```



Разделяем данные на обучающую и тестовую выборки

```
X = data.drop(columns=['rent amount (R$)'])
y = data['rent amount (R$)']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Обучение моделей

1. Линейная регресия

```
lr = LinearRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

# Оценка модели линейной регрессии
```

```
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print(f"Модель линейной регрессии:")
print(f"Среднеквадратичная ошибка (MSE) = {mse_lr}")
print(f"Средняя абсолютная ошибка (MAE) = {mae_lr}")
print(f"Коэффициент детерминации (R²) = {r2_lr}")
```

```
Модель линейной регрессии:
Среднеквадратичная ошибка (MSE) = 9.641379452147344e-08
Средняя абсолютная ошибка (MAE) = 0.00015485319269155533
Коэффициент детерминации (R²) = 0.9999998986066924
```

1. Случайный лес

```
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

# Оценка модели случайного леса
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print(f"Модель случайного леса:")
print(f"Среднеквадратичная ошибка (MSE) = {mse_rf}")
print(f"Средняя абсолютная ошибка (MAE) = {mae_rf}")
print(f"Коэффициент детерминации (R²) = {r2_rf}")
```

```
Модель случайного леса:
Среднеквадратичная ошибка (MSE) = 0.004101190839377893
Средняя абсолютная ошибка (MAE) = 0.016902734972421122
Коэффициент детерминации (R²) = 0.9956869936879017
```

Вывод

Сравнивая метрики качества, можно сделать вывод, что обе модели показали себя хорошо, мы получили практически идентичные результаты, модель случайного леса совсем на немного лучше модели линейной регрессии, наибольшее различие сдесь в метрике MSE. Линейная регрессия предполагает, что независимые переменные не коррелируют между собой. Если это предположение нарушается, то линейная регрессия может давать неточные результаты. Случайный лес может обрабатывать взаимосвязанные переменные, поэтому он может быть более точным, чем линейная регрессия, в случаях, когда независимые переменные коррелируют между собой.