

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения».

Отчет по лабораторной работе №6
«Создание веб-приложения для демонстрации моделей машинного
обучения.»

Выполнил:
студент группы ИУ5-61Б
Головацкий А.Д.

Проверил:
Гапанюк Ю. Е.

Москва, 2022 г.

Задание:

Разработайте макет веб-приложения, предназначенного для анализа данных.

Вариант 1. Макет должен быть реализован для одной модели машинного обучения. Макет должен позволять:

- задавать гиперпараметры алгоритма,
- производить обучение,
- осуществлять просмотр результатов обучения, в том числе в виде графиков.

Вариант 2. Макет должен быть реализован для нескольких моделей машинного обучения. Макет должен позволять:

- выбирать модели для обучения,
- производить обучение,
- осуществлять просмотр результатов обучения, в том числе в виде графиков.

Для реализации был выбран вариант 1.

Код программы:

```
import streamlit as st
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

```
st.markdown(
```

```
    ""
```

```

<style>
    .main {
        background-color: #ebebeb;

    }

</style>
"""',
    unsafe_allow_html=True
)

siteHeader = st.container()
dataExploration = st.container()
newFeatures = st.container()
modelTraining = st.container()
allFeaturesModel = st.container()

def build_model(df):
    X = df.iloc[:, :-1]
    Y = df.iloc[:, -1]

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.8)

    st.markdown('**Делим данные**')
    st.write('Обучающая выборка')
    st.info(X_train.shape)
    st.write('Тестовая выборка')
    st.info(X_test.shape)

    st.write('X')

```

```
st.info(list(X.columns))
```

```
st.write('Y')
```

```
st.info(Y.name)
```

```
rf = RandomForestRegressor(n_estimators=number_of_trees,  
                           max_depth=max_depth,  
                           random_state=1)
```

```
rf.fit(X_train, Y_train)
```

```
st.subheader('Метрики модели')
```

```
Y_pred_train = rf.predict(X_train)
```

```
Y_pred_test = rf.predict(X_test)
```

```
st.write('Коэффициент детерминации ( $R^2$ ):')
```

```
st.info(r2_score(Y_test, Y_pred_test))
```

```
st.write('MSE:')
```

```
st.info(mean_squared_error(Y_test, Y_pred_test))
```

```
st.write('MAE:')
```

```
st.info(mean_absolute_error(Y_test, Y_pred_test))
```

```
st.subheader('Параметры модели')
```

```
st.write(rf.get_params())
```

```
with siteHeader:
```

```
st.title('ИУ5-61Б Головацкий Андрей Лабораторная работа №6')
```

```
st.text('Разработайте макет веб-приложения, предназначенного для анализа  
данных.\nВариант 1. Макет должен быть реализован для одной модели
```

машинного обучения.\nМакет должен позволять:\n задавать гиперпараметры алгоритма,\n производить обучение,\n осуществлять просмотр результатов обучения, в том числе в виде графиков.)

with dataExploration:

```
st.header('Датасет: аренда квартир в Бразилии')
df = pd.read_csv('~\ML\datasets\houses_to_rent_v2.csv')
distribution_pickup = pd.DataFrame(df['total (R$)'].value_counts())
st.write(df.head(10))
st.text('Кодирование категориальных признаков')
LE = LabelEncoder()
for col in df.columns:
    if df[col].dtype == "object":
        df[col] = LE.fit_transform(df[col])
st.write(df.head(10))
```

with modelTraining:

```
st.header('Обучение модели (Случайный лес)')
st.text('Можем изменить максимальную глубину, число деревьев или же выбрать исходный признак')
selection_col, display_col = st.columns(2)
max_depth = st.slider('Максимальная глубина:', min_value=10,
max_value=100, value=20, step=10)
number_of_trees = st.selectbox('Число деревьев:', options=[100, 200, 300, 400], index=0)
selection_col.text('Список признаков:')
selection_col.write(df.columns)
input_feature = st.text_input('Исходный признак:', 'hoa (R$)')
regr = RandomForestRegressor(max_depth=max_depth,
n_estimators=number_of_trees, random_state=1)
```

```
X = df[[input_feature]]
y = df[['total (R$)']]

regr.fit(X, y)
prediction = regr.predict(y)
display_col.subheader('MAE:')
display_col.write(mean_absolute_error(y, prediction))
display_col.subheader('MSE:')
display_col.write(mean_squared_error(y, prediction))
display_col.subheader('R2:')
display_col.write(r2_score(y, prediction))
```

with allFeaturesModel:

```
st.header('Обучение модели (все признаки)')
build_model(df)
```

Результаты работы:

ИУ5-61Б Головацкий Андрей

Лабораторная работа №6

Разработайте макет веб-приложения, предназначенного для анализа данных.
Вариант 1. Макет должен быть реализован для одной модели машинного обучения.
Макет должен позволять:

- задавать гиперпараметры алгоритма,
- производить обучение,
- осуществлять просмотр результатов обучения, в том числе в виде графиков.

Датасет: аренда квартир в Бразилии

	city	area	room	bath	park	floor	animal	furniture	hoa (R\$)	rent am	prop	fire ir	t
0	São Paulo	70	2	1	1	7	accept	furnished	2065	3300	211	42	
1	São Paulo	320	4	4	0	20	accept	not furnished	1200	4960	1750	63	
2	Porto Alegre	80	1	1	1	6	accept	not furnished	1000	2800	0	41	
3	Porto Alegre	51	2	1	0	2	accept	not furnished	270	1112	22	17	
4	São Paulo	25	1	1	0	1	not accept	not furnished	0	800	25	11	
5	São Paulo	376	3	3	7	-	accept	not furnished	0	8000	834	121	
6	Rio de Janeiro	72	2	1	0	7	accept	not furnished	740	1900	85	25	
7	São Paulo	213	4	4	4	4	accept	not furnished	2254	3223	1735	41	
8	São Paulo	152	2	2	1	3	accept	furnished	1000	15000	250	191	
9	Rio de Janeiro	35	1	1	0	2	accept	furnished	590	2300	35	30	

Кодирование категориальных признаков

Кодирование категориальных признаков

	city	area	roon	bath	park	floor	anim	furni	hoa (R\$)	rent am	prope	fire ir	total (R\$)
0	4	70	2	1	1	32	0	0	2065	3300	211	42	5618
1	4	320	4	4	0	13	0	1	1200	4960	1750	63	7973
2	2	80	1	1	1	31	0	1	1000	2800	0	41	3841
3	2	51	2	1	0	12	0	1	270	1112	22	17	1421
4	4	25	1	1	0	1	1	1	0	800	25	11	836
5	4	376	3	3	7	0	0	1	0	8000	834	121	8955
6	3	72	2	1	0	32	0	1	740	1900	85	25	2750
7	4	213	4	4	4	27	0	1	2254	3223	1735	41	7253
8	4	152	2	2	1	23	0	0	1000	15000	250	191	16440
9	3	35	1	1	0	12	0	0	590	2300	35	30	2955

Обучение модели (Случайный лес)

Можем изменить максимальную глубину, число деревьев или же выбрать исходный признак

Список признаков:

	0
0	city
1	area
2	rooms
3	bathroom
4	parking spaces
5	floor

MAE:

9290.789021161077

MSE:

105302685.49217108

R2:

0.6124599363567498

Обучение модели (Случайный лес)

Можем изменить максимальную глубину, число деревьев или же выбрать исходный признак

Список признаков:

	0
0	city
1	area
2	rooms
3	bathroom
4	parking spaces
5	floor
6	animal
7	furniture
8	hoa (R\$)
9	rent amount (R\$)

MAE:

9290.789021161077

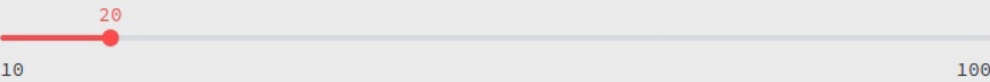
MSE:

105302685.49217108

R2:

0.6124599363567498

Максимальная глубина:



Число деревьев:

400

Исходный признак:

Исходный признак:

hoa (R\$)

Обучение модели (все признаки)

Делим данные

Обучающая выборка

(2138, 12)

Тестовая выборка

(8554, 12)

X

['city', 'area', 'rooms', 'bathroom', 'parking spaces', 'floor', 'animal', 'furniture', 'hoa (R)', 'rentamount(R)', 'property tax (R)', 'fireinsurance(R)']

Y

total (R\$)

Метрики модели

Коэффициент детерминации (R^2):

0.7816885581888781

Метрики модели

Коэффициент детерминации (R^2):

0.7810325024233791

MSE:

41181591.67580741

MAE:

268.20769511026583

Параметры модели

```
{
  "bootstrap" : true
  "ccp_alpha" : 0
  "criterion" : "squared_error"
  "max_depth" : 20
  "max_features" : "auto"
  "max_leaf_nodes" : NULL
  "max_samples" : NULL
  "min_impurity_decrease" : 0
  "min_samples_leaf" : 1
  "min_samples_split" : 2
  "min_weight_fraction_leaf" : 0
}
```