



Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э.
Баумана»
(МГТУ им. Н.Э. Баумана)

Лабораторная работа 6
по курсу «Технологии машинного обучения»

Выполнил
студент группы ИУ5-64
XXX

Москва, 2021

Цель работы

изучение возможностей демонстрации моделей машинного обучения с помощью веб-приложений.

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.

Код

```
1 #!/usr/bin/env python
2 import streamlit as st
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from sklearn.preprocessing import MinMaxScaler, LabelEncoder
7 from sklearn.model_selection import train_test_split
8 from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error,
   mean_absolute_percentage_error
9 from sklearn.ensemble import BaggingRegressor, RandomForestRegressor, AdaBoostRegressor,
   GradientBoostingRegressor
10
11 st.header('Датасет')
12 main_status = st.text('')
13 read_state = st.text('Чтение датасета...')
14 data = pd.read_csv("../2/melbourne_housing.csv")
15 columns_and_types = {
16     "Rooms": np.int64,
17     "Type": None,
18     "Price": np.int64,
19     "Distance": np.float64,
20     "Postcode": np.int64,
21     "Bedroom2": np.int64,
22     "Bathroom": np.int64,
23     "Car": np.int64,
24     "Landsize": np.float64,
25     "BuildingArea": np.float64,
26     "YearBuilt": np.int64,
27     "Latitude": np.float64,
28     "Longitude": np.float64,
29     "Propertycount": np.int64,
30 }
31 data = data[list(columns_and_types.keys())]
32 data.dropna(axis=0, how='any', inplace=True)
33 data = data.astype({k: v for k, v in columns_and_types.items() if v is not None})
34 type_encoder = LabelEncoder()
```

```

35 data["Type"] = type_encoder.fit_transform(data["Type"])
36 read_state.text('Датасет загружен!')
37
38 st.subheader('head:')
39 st.write(data.head())
40
41 test_size = st.sidebar.slider("test_size", 0.1, 0.9, value = 0.3)
42 n_estimators = st.sidebar.slider("n_estimators", 1, 50, value=5)
43 random_state = st.sidebar.slider("random_state", 1, 100, value=10)
44
45 target_option = st.sidebar.selectbox('Target:', data.columns)
46 feature_cols = []
47 st.sidebar.subheader('Features:')
48 for col in data.columns:
49     cb = st.sidebar.checkbox(col, value=True)
50     if cb:
51         feature_cols.append(col)
52
53 st.sidebar.subheader('Масштабирование')
54 if st.sidebar.checkbox('MinMaxScaler'):
55     sc2 = MinMaxScaler()
56
57     for col in data.columns:
58         data[col] = sc2.fit_transform(data[[col]])
59
60 st.subheader('Масштабирование:')
61 st.write(data.head())
62
63 main_status.text('В процессе обучения...')
64 data_X = data.loc[:, [x for x in feature_cols if x != target_option]]
65 data_Y = data.loc[:, target_option]
66 data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(
67     data_X,
68     data_Y,
69     test_size=test_size,
70     random_state=1,
71 )
72
73 bc = BaggingRegressor(n_estimators=n_estimators, oob_score=True, random_state=random_state)
74 bc.fit(data_X_train, data_y_train)
75
76 rfc = RandomForestRegressor(n_estimators=n_estimators, oob_score=True, random_state=random_state)
77 rfc.fit(data_X_train, data_y_train)
78
79 abc = AdaBoostRegressor(n_estimators=n_estimators, random_state=random_state)
80 abc.fit(data_X_train, data_y_train)
81
82 gbc = GradientBoostingRegressor(random_state=random_state)
83 gbc.fit(data_X_train, data_y_train)
84
85 main_status.text("Обучено!")
86
87 metrics = [mean_absolute_error, mean_squared_error, median_absolute_error, mean_absolute_percentage_error]
88 metr = [i.__name__ for i in metrics]
89 metrics_ms = st.sidebar.multiselect("Метрики", metr)
90
91 methods = [bc, rfc, abc, gbc]
92 md = [i.__class__.__name__ for i in methods]
93 methods_ms = st.sidebar.multiselect("Методы обучения", md)
94
95 selMethods = []
96 for i in methods_ms:
97     for j in methods:
98         if i == j.__class__.__name__:
99             selMethods.append(j)
100
101 selMetrics = []
102 for i in metrics_ms:
103     for j in metrics:
104         if i == j.__name__:
105             selMetrics.append(j)
106
107 st.header('Оценка')
108 for name in selMetrics:
109     st.subheader(name.__name__)
110     for func in selMethods:
111         y_pred = func.predict(data_X_test)
112
113         fig, ax = plt.subplots()
114         ax.plot(data_X_test, data_y_test, 'b.')
115         ax.plot(data_X_test, y_pred, 'r.')

```

```

116 st.pyplot(fig)
117
118 st.text("{} - {}".format(func.__class__.__name__, name(y_pred, data_y_test)))

```

Результат работы

