



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, StandardScaler, LabelEncoder
7 from sklearn.model_selection import train_test_split
8 from sklearn.metrics import r2_score, 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 scaler = st.sidebar.radio("Масштабирование", ("Нет", "MinMaxScaler", "StandardScaler"))
54 if scaler == 'MinMaxScaler':
55     sc2 = MinMaxScaler()
56
57     for col in data.columns:
58         if col != target_option:
59             data[col] = sc2.fit_transform(data[[col]])
60
61     st.subheader('Масштабирование:')
62     st.write(data.head())
63 elif scaler == 'StandardScaler':
64     sc2 = StandardScaler()
65
66     for col in data.columns:
67         if col != target_option:
68             data[col] = sc2.fit_transform(data[[col]])
69
70     st.subheader('Масштабирование:')
71     st.write(data.head())
72
73 main_status.text('В процессе обучения...')
74 data_X = data.loc[:, [x for x in feature_cols if x != target_option]]
75 data_Y = data.loc[:, target_option]
76 data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(
77     data_X,
78     data_Y,
79     test_size=test_size,
80     random_state=1,
81 )
82
83 bc = BaggingRegressor(n_estimators=n_estimators, oob_score=True, random_state=random_state)
84 bc.fit(data_X_train, data_y_train)
85
86 rfc = RandomForestRegressor(n_estimators=n_estimators, oob_score=True, random_state=random_state)
87 rfc.fit(data_X_train, data_y_train)
88
89 gbc = GradientBoostingRegressor(random_state=random_state)
90 gbc.fit(data_X_train, data_y_train)
91
92 main_status.text("Обучено!")
93
94 metrics = [r2_score, mean_absolute_error, mean_squared_error, median_absolute_error,
95             mean_absolute_percentage_error]
96 metr = [i.__name__ for i in metrics]
97 metrics_ms = st.sidebar.multiselect("Метрики", metr)
98
99 methods = [bc, rfc, gbc]
100 md = [i.__class__.__name__ for i in methods]
101 methods_ms = st.sidebar.multiselect("Методы обучения", md)
102 selMethods = []
103 for i in methods_ms:
104     for j in methods:
105         if i == j.__class__.__name__:
106             selMethods.append(j)
107
108 selMetrics = []
109 for i in metrics_ms:
110     for j in metrics:
111         if i == j.__name__:
112             selMetrics.append(j)
113
114 st.header('Оценка')

```

```

115 for name in selMetrics:
116     st.subheader(name.__name__)
117
118     array_labels = [ ]
119     array_metric = [ ]
120
121     for func in selMethods:
122         y_pred = func.predict(data_X_test)
123
124         array_labels.append(func.__class__.__name__)
125         array_metric.append(name(y_pred, data_y_test))
126
127         st.text("{} - {}".format(func.__class__.__name__, name(y_pred, data_y_test)))
128
129     fig, ax1 = plt.subplots(figsize=(3,3))
130     pos = np.arange(len(array_metric))
131     rects = ax1.barh(
132         pos,
133         array_metric,
134         align="center",
135         height=0.5,
136         tick_label=array_labels,
137     )
138     for a, b in zip(pos, array_metric):
139         plt.text(0, a - 0.1, str(round(b, 3)), color="white")
140     st.pyplot(fig)

```

Результат работы

☒ Type

☒ Price

☒ Distance

☒ Postcode

☒ Bedroom2

☒ Bathroom

☒ Car

☒ Landsize

☒ BuildingArea

☒ YearBuilt

☒ Latitude

☒ Longitude

☒ Propertycount

Масштабирование

☒ Нет

☐ MinMaxScaler

☐ StandardScaler

Метрики

r2_score

×

mean_squared_error

×

mean_absolute_error

×

mean_absolute_perc...

×

Методы обучения

BaggingRegressor

×

RandomForestRegres...

×

GradientBoostingReg...

×

Датасет

Обучено!

Датасет загружен!

head:

	Rooms	Type	Price	Distance	Postcode
2	2	0	1035000	2.5000	3067
4	3	0	1465000	2.5000	3067
6	4	0	1600000	2.5000	3067
11	3	0	1876000	2.5000	3067
14	2	0	1636000	2.5000	3067

Оценка

r2_score

BaggingRegressor - 0.7171670616269736

RandomForestRegressor - 0.721172254049621

GradientBoostingRegressor - 0.7430418659131932

GradientBoostingRegressor0.743

RandomForestRegressor0.721

BaggingRegressor0.717

0.0

0.2

0.4

0.6

mean_squared_error

4