



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»

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

Выполнила:
студент группы ИУ5-61Б
Павловская А.А.
03.06.2021

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Цель лабораторной 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.

Для разработки рекомендуется использовать следующие (или аналогичные) фреймворки:

- [streamlit](https://streamlit.io/)
- [gradio](https://gradio.app/)
- [dash](https://dash.io/)

Набор данных: U.S. Education Datasets: Unification Project

<https://www.kaggle.com/noriuk/us-education-datasets-unification-project> (файл states_all.csv)

Текст программы:

```
import streamlit as st
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, BaggingRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
import matplotlib.pyplot as plt

def load_data():
    """
    Загрузка данных
    """
```

```

data = pd.read_csv('C:\Py\Lab6_TM0\data\states_all(1).csv', sep=' ')
return data

st.sidebar.header('Модели машинного обучения')

st.header('Данные')
status = st.text('Загрузка данных ...')
data = load_data()
status.text('Загрузка данных завершена')

type_encoder = LabelEncoder()
data["STATE"] = type_encoder.fit_transform(data["STATE"])

if st.checkbox('Показать корреляционную матрицу'):
    fig1, ax = plt.subplots(figsize=(10,5))
    sns.heatmap(data.corr(), annot=True, fmt='.2f')
    st.pyplot(fig1)

data_len = data.shape[0]
st.write('Количество строк в наборе данных - {}'.format(data_len))
st.subheader('Первые пять значений:')
st.write(data.head())

if st.checkbox('Масштабирование'):
    sc2 = MinMaxScaler()
    for col in data.columns:
        if col != 'STATE':
            data[col] = sc2.fit_transform(data[[col]])
    st.write(data.head())

st.header('Оценка качества моделей')

def preprocess_data(data):
    X = data.drop(['TOTAL_REVENUE'], axis = 1)
    Y = data.TOTAL_REVENUE
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state =
0, test_size = 0.1)
    return X_train, X_test, Y_train, Y_test

X_train, X_test, Y_train, Y_test = preprocess_data(data)

LogR = LinearRegression()
LogR.fit(X_train, Y_train)

KNN_5 = KNeighborsRegressor(n_neighbors=5)
KNN_5.fit(X_train, Y_train)

BC = BaggingRegressor()
BC.fit(X_train, Y_train)

Tree = DecisionTreeRegressor()

```

```

Tree.fit(X_train, Y_train)

RF = RandomForestRegressor()
RF.fit(X_train, Y_train)

metrics = [r2_score, mean_absolute_error, mean_squared_error, median_absolute_err
or]

models_list = [LogR, KNN_5, BC, Tree, RF]
model_names = [i.__class__.__name__ for i in models_list]
models = st.sidebar.multiselect("Выберите модели", model_names)

current_models_list = []
for i in models:
    for j in models_list:
        if i == j.__class__.__name__:
            current_models_list.append(j)

if st.sidebar.checkbox('Выполнить оценку качества'):
    for name in metrics:
        st.subheader(name.__name__)

        array_labels = []
        array_metric = []

        for func in current_models_list:
            Y_pred = func.predict(X_test)

            array_labels.append(func.__class__.__name__)
            array_metric.append(name(Y_pred, Y_test))

            st.text("{} - {}".format(func.__class__.__name__, name(Y_pred, Y_test
)))

fig, ax1 = plt.subplots(figsize=(3,3))
pos = np.arange(len(array_metric))
rects = ax1.barh(
    pos,
    array_metric,
    align="center",
    height=0.5,
    tick_label=array_labels,
)
for a, b in zip(pos, array_metric):
    plt.text(0, a - 0.1, str(round(b, 3)), color="white")
st.pyplot(fig)

```

Экранные формы с примерами выполнения программы:

×

Модели машинного обучения

Выберите модели

Choose an option

▼

☐ Выполнить оценку качества

×

Модели машинного обучения

Выберите модели

Choose an option

▼

☐ Выполнить оценку качества

Данные

Загрузка данных завершена

☐ Показать корреляционную матрицу

Количество строк в наборе данных - 500

Первые пять значений:

	STATE	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTA
0	0	2678885	304177	1659028	715680	
1	1	1049591	106780	720711	222100	
2	2	3258079	297888	1369815	1590376	
3	3	1711959	178571	958785	574603	
4	4	26260025	2072470	16546514	7641041	

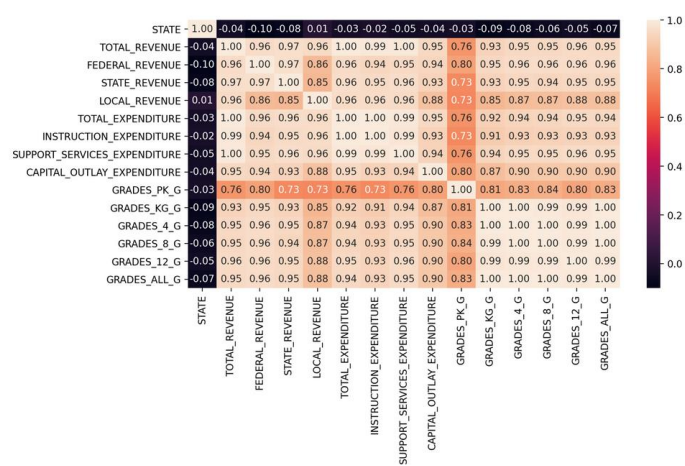
☐ Масштабирование

Оценка качества моделей

Данные

Загрузка данных завершена

☒ Показать корреляционную матрицу



Количество строк в наборе данных - 500

Модели машинного обучения

Выберите модели

Choose an option

☐ Выполнить оценку качества

Модели машинного обучения

Выберите модели

LinearRegression

KNeighborsRegressor

RandomForestRegres...

DecisionTreeRegressor

☒ Выполнить оценку качества

Модели машинного обучения

Выберите модели

LinearRegression

KNeighborsRegressor

RandomForestRegres...

DecisionTreeRegressor

☒ Выполнить оценку качества

Данные

Загрузка данных завершена

☐ Показать корреляционную матрицу

Количество строк в наборе данных - 500

Первые пять значений:

	STATE	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTA
0	0	2678885	304177	1659028	715680	
1	1	1049591	106780	720711	222100	
2	2	3258079	297888	1369815	1590376	
3	3	1711959	178571	958785	574603	
4	4	26260025	2072470	16546514	7641041	

☒ Масштабирование

	STATE	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTA
0	0	0.0419	0.0643	0.0529	0.0394	
1	1	0.0111	0.0178	0.0230	0.0114	
2	2	0.0529	0.0628	0.0437	0.0890	
3	3	0.0236	0.0347	0.0306	0.0314	
4	4	0.4883	0.4804	0.5274	0.4326	

Оценка качества моделей

r2_score

LinearRegression - 0.9999999999999999

KNeighborsRegressor - 0.9083808446328372

RandomForestRegressor - 0.9969549108469065

DecisionTreeRegressor - 0.9965737480071643



mean_absolute_error

LinearRegression - 7.491695089290041e-10

KNeighborsRegressor - 0.015025445081585502

RandomForestRegressor - 0.0034528987391982575

DecisionTreeRegressor - 0.004405913126241485



Модели машинного обучения

Выберите модели

LinearRegression

KNeighborsRegressor

RandomForestRegres...

DecisionTreeRegressor

☒

Выполнить оценку качества

Модели машинного обучения

Выберите модели

LinearRegression

KNeighborsRegressor

RandomForestRegres...

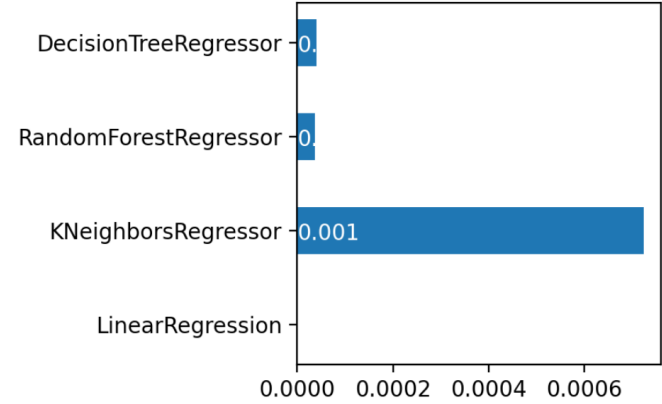
DecisionTreeRegressor

☒

Выполнить оценку качества

mean_squared_error

LinearRegression - 1.1007861676716665e-18
KNeighborsRegressor - 0.0007240786206495659
RandomForestRegressor - 3.693346417187548e-05
DecisionTreeRegressor - 3.956417344947261e-05



median_absolute_error

LinearRegression - 5.913161835180425e-10
KNeighborsRegressor - 0.0054170136647860995
RandomForestRegressor - 0.002089978847021251
DecisionTreeRegressor - 0.003143291350654727

