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)

Работа №4
по курсу «Технологии машинного обучения»

Выполнил

студент группы ИУ5-64Б

Шпак И.Д.

Москва, 2020

1 Исходное задание

Разработайте макет веб-приложения, предназначенного для анализа данных.

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.

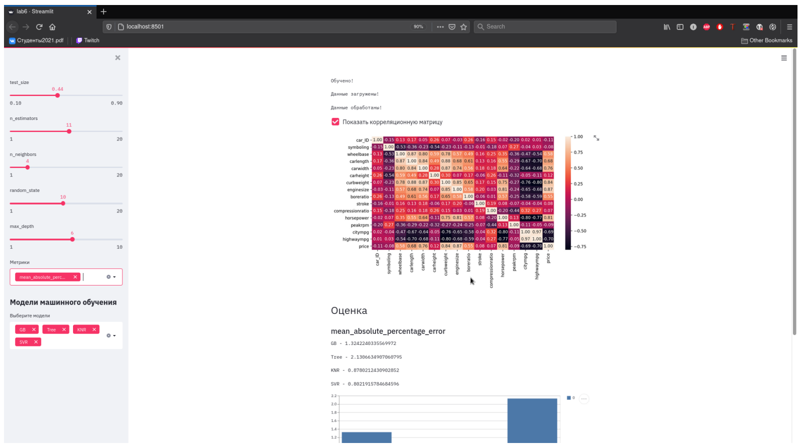
2 Код программы

```
from sklearn.metrics._plot.confusion_matrix import plot_confusion_matrix
import streamlit as st
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.svm import LinearSVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, median_absolute_error, mean_absolut

@st.cache
def load_data():
    '''


    '''
    data = pd.read_csv("/home/igor/Downloads/CarPrice_Assignment.xls",sep=',')
    return data



@st.cache
def preprocess_data(data_in):
    data_out = data_in.copy()
    cleanup_nums = {"doornumber":     {"four": 0, "two": 1},
                "cylindernumber": {"four": 4, "six": 6, "five": 5, "eight": 8,
                                  "two": 2, "twelve": 12, "three":3 },
                "fueltype" :       {"gas":0,"diesel":1},
                "aspiration":     {"std":0,"turbo":1},
                "enginelocation": {"front":0,"rear":1}}
    data_out = data_out.replace(cleanup_nums)
    data_out=pd.get_dummies(data_out, columns=["carbody"], prefix=["carbody"])
    data_out=pd.get_dummies(data_out, columns=["drivewheel"], prefix=["drive"])
    data_out["OHC_Code"] = np.where(data_out["enginetype"].str.contains("ohc"), 1, 0)
    data_out.drop(["CarName","enginetype","fuelsystem","symboling","car_ID"],axis=1,inplace=True)
    cols = data_out.columns
    sc1 = StandardScaler()
    data_out = sc1.fit_transform(data_out)
    data_out = pd.DataFrame(data_out,columns=cols)
    data_Y = data_out.loc[:, 'price']
    data_X = data_out.drop(["price"],axis=1,inplace=False)
    return  data_X, data_Y

main_status = st.text('')



data_load_state = st.text('         ...')
data = load_data()
data_load_state.text('         !')

data_load_state = st.text('                   ...')
data_X,data_Y = preprocess_data(data)
data_load_state.text('          !')
#
data_len = data_X.shape[0]



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

test_size = st.sidebar.slider("test_size", 0.1, 0.9, value = 0.3)
n_estimators = st.sidebar.slider("n_estimators", 1, 20, value=5)
```

```
n_neighbors = st.sidebar.slider("n_neighbors", 1, 20, value=5)
random_state = st.sidebar.slider("random_state", 1, 20, value=10)
max_depth = st.sidebar.slider("max_depth", 1, 10, value=4)

main_status.text('             ...')

data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(
    data_X, data_Y,test_size=test_size, random_state=1)

#
models_list = ['BagR', 'KNR', 'SVR', 'Tree', 'RF', 'GB']
clas_models = {'BagR': BaggingRegressor(n_estimators=n_estimators, random_state=random_state),
               'KNR':KNeighborsRegressor(n_neighbors=n_neighbors),
               'SVR': LinearSVR(random_state=random_state),
               'Tree':DecisionTreeRegressor(max_depth=max_depth, random_state=random_state),
               'RF':RandomForestRegressor(max_depth=max_depth, n_estimators=n_estimators, random_state=ra
               'GB':GradientBoostingRegressor(n_estimators = n_estimators, random_state=random_state),
               'AB':AdaBoostRegressor(n_estimators = n_estimators,random_state=random_state)}


main_status.text('    !')

metrics = [mean_absolute_error, mean_squared_error, median_absolute_error, mean_absolute_percentage_erro
metr = [i.__name__ for i in metrics]
metrics_ms = st.sidebar.multiselect("    ", metr)

st.sidebar.header('              ')
models_select = st.sidebar.multiselect('         ', models_list)

selMetrics = []
for i in metrics_ms:
    for j in metrics:
        if i == j.__name__:
            selMetrics.append(j)



st.header('    ')
for name in selMetrics:
    st.subheader(name.__name__)
    scorelist={}
    for model_name in models_select:
```

```
    model = clas_models[model_name]
    model.fit(data_X_train, data_y_train)
    #
    Y_pred = model.predict(data_X_test)
    score =  name(Y_pred, data_y_test)
    scorelist[model_name]=score
    st.text("{} - {}".format(model_name, score))
if not scorelist:
    continue
print(scorelist)
scorelist = pd.DataFrame.from_dict(scorelist, orient = 'index')
st.bar_chart(scorelist)
```

3 Демонстрация работы программы



test_size: 0.44 (0.10 – 0.90)
n_estimators: 11 (1 – 20)
n_neighbors: 4 (1 – 20)
random_state: 10 (1 – 20)
max_depth: 6 (1 – 10)
Метрики: mean_absolute_perc...

Модели машинного обучения

Выберите модели: GB, Tree, KNR, SVR

```
Обучено!
Данные загружены!
Данные обработаны!
```

[✓] Показать корреляционную матрицу

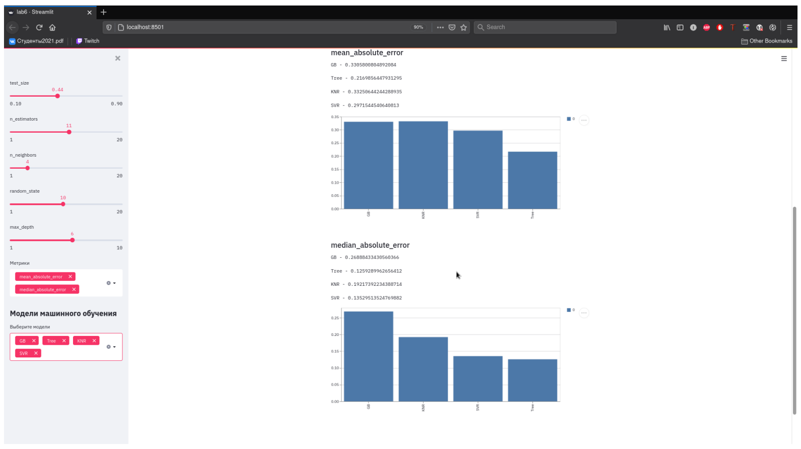
	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressionratio	horsepower	peakrpm	citympg	highwaympg	price
car_ID	1.00	-0.15	0.13	0.17	0.05	0.26	0.07	-0.03	0.26	-0.16	0.15	-0.02	-0.20	0.02	0.01	-0.11
symboling	-0.15	1.00	-0.53	-0.36	-0.23	-0.54	-0.23	-0.11	-0.13	-0.01	-0.18	0.07	0.27	-0.04	0.03	-0.08
wheelbase	0.13	-0.53	1.00	0.87	0.80	0.59	0.78	0.57	0.49	0.16	0.25	0.35	-0.36	-0.47	-0.54	0.58
carlength	0.17	-0.36	0.87	1.00	0.84	0.49	0.88	0.68	0.61	0.13	0.16	0.55	-0.29	-0.67	-0.70	0.68
carwidth	0.05	-0.23	0.80	0.84	1.00	0.28	0.87	0.74	0.56	0.18	0.18	0.64	-0.22	-0.64	-0.68	0.76
carheight	0.26	-0.54	0.59	0.49	0.28	1.00	0.30	0.07	0.17	-0.06	0.26	-0.11	-0.32	-0.05	-0.11	0.12
curbweight	0.07	-0.23	0.78	0.88	0.87	0.30	1.00	0.85	0.65	0.17	0.15	0.75	-0.27	-0.76	-0.80	0.84
enginesize	-0.03	-0.11	0.57	0.68	0.74	0.07	0.85	1.00	0.58	0.20	0.03	0.81	-0.24	-0.65	-0.68	0.87
boreratio	0.26	-0.13	0.49	0.61	0.56	0.17	0.65	0.58	1.00	-0.06	0.01	0.57	-0.25	-0.58	-0.59	0.55
stroke	-0.16	-0.01	0.16	0.13	0.18	-0.06	0.17	0.20	-0.06	1.00	0.19	0.08	-0.07	-0.04	-0.04	0.08
compressionratio	0.15	-0.18	0.25	0.16	0.18	0.26	0.15	0.03	0.01	0.19	1.00	-0.20	-0.44	0.32	0.27	0.07
horsepower	-0.02	0.07	0.35	0.55	0.64	-0.11	0.75	0.81	0.57	0.08	-0.20	1.00	0.13	-0.80	-0.77	0.81
peakrpm	-0.20	0.27	-0.36	-0.29	-0.22	-0.32	-0.27	-0.24	-0.25	-0.07	-0.44	0.13	1.00	-0.11	-0.05	-0.09
citympg	0.02	-0.04	-0.47	-0.67	-0.64	-0.05	-0.76	-0.65	-0.58	-0.04	0.32	-0.80	-0.11	1.00	0.97	-0.69
highwaympg	0.01	0.03	-0.54	-0.70	-0.68	-0.11	-0.80	-0.68	-0.59	-0.04	0.27	-0.77	-0.05	0.97	1.00	-0.70
price	-0.11	-0.08	0.58	0.68	0.76	0.12	0.84	0.87	0.55	0.08	0.07	0.81	-0.09	-0.69	-0.70	1.00

Оценка

mean_absolute_percentage_error

```
GB - 1.3242240335569972

Tree - 2.1306634907060795

KNR - 0.8780212430902852

SVR - 0.8021915784684596
```

Рис. 3.1:



test_size: 0.44 (0.10 – 0.90)
n_estimators: 11 (1 – 20)
n_neighbors: 4 (1 – 20)
random_state: 10 (1 – 20)
max_depth: 6 (1 – 10)
Метрики: mean_absolute_error, median_absolute_error

Модели машинного обучения

Выберите модели: GB, Tree, KNR, SVR

mean_absolute_error

```
GB - 0.3305800804892084

Tree - 0.2169856447931295

KNR - 0.33250644244288935

SVR - 0.2971544540640813
```

median_absolute_error

```
GB - 0.26888433430560366

Tree - 0.1259289962656412

KNR - 0.19217392234388714

SVR - 0.13529513524769882
```

Рис. 3.2: