



**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, искусственный и системы управления»
Кафедра «Системы обработки информации и управления»**

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

**Выполнил:
студент группы ИУ5-61Б
И.А. Калинин**

**Проверил:
Ю.Е. Гапанюк**

2023 г.

```

#1. Импортировать библиотеки:

import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score,
learning_curve
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder

from sklearn.neighbors import KNeighborsClassifier

#2. Создать заголовок и описание страницы:

st.title('Макет веб-приложения для анализа данных')
st.write('Это макет веб-приложения для анализа данных, который позволяет
производить обучение модели линейной регрессии и просматривать результаты
обучения.')

#3. Загрузить данные:

data = pd.read_csv('Admission_Predict.csv')

X = data.drop(['Chance of Admit '], axis=1)
y = data['Chance of Admit ']
le = LabelEncoder()
y = le.fit_transform(y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
shuffle=False, random_state=45)

#4. Вывести информацию о данных:

st.write('Информация о данных:')
st.write(data.head())

#5. Создать форму для задания гиперпараметров:

st.write('Задайте гиперпараметры:')
epochs = st.slider('Количество эпох', min_value=1, max_value=100, value=10)
batch_size = st.slider('Размер батча', min_value=1, max_value=100, value=10)
hidden_layer_size = st.slider('Размер скрытого слоя', min_value=1,
max_value=100, value=10)
input_dimension = st.slider('Размерность входных данных', min_value=1,
max_value=100, value=10)

#6. Создать кнопку для обучения модели:

if st.button('Разделить данные на обучающую и тестовую выборки'):
    # Разделить данные на обучающую и тестовую выборки
    X = data.drop(['Chance of Admit '], axis=1) # Наименования признаков
    y = data['Chance of Admit ']
    le = LabelEncoder()
    y = le.fit_transform(y)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
shuffle=False, random_state=45)
    st.write('Размер обучающей выборки:')
    st.write(X_train.shape, y_train.shape)
    st.write('Размер тестовой выборки:')
    st.write(X_test.shape, y_test.shape)

if st.button('Обучить модель для линейной регрессии'):
    model = LinearRegression()

```

```

        model.fit(X_train, y_train)

# Задаем значение K
K = epochs
# Создаем объект классификатора
knn = KNeighborsClassifier(n_neighbors=K)
if st.button('Создаем объект классификатора'):
    # Обучаем модель на тренировочных данных
    knn.fit(X_train, y_train)
    # Предсказываем метки классов для тестовых данных
    y_pred = knn.predict(X_test)
    # Вычисляем точность модели с помощью 5-кратной кросс-валидации
    scores = cross_val_score(knn, X, y, cv=5)
    # Выводим среднее значение и стандартное отклонение точности
    st.write("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() *
2))

# Вывести информацию о точности модели

#7. Вывести график изменения точности на каждой эпохе:

if st.button('Показать график точности'):
    # Создать график точности
    plt.plot(np.arange(epochs), np.random.rand(epochs))
    plt.title('График изменения точности на каждой эпохе')
    plt.xlabel('Эпохи')
    plt.ylabel('Точность')
    st.pyplot()

if st.button('График кривых и валидации'):
    train_sizes, train_scores, test_scores = learning_curve(knn, X, y, cv=5)

    train_mean = np.mean(train_scores, axis=1)
    train_std = np.std(train_scores, axis=1)
    test_mean = np.mean(test_scores, axis=1)
    test_std = np.std(test_scores, axis=1)

    plt.plot(train_sizes, train_mean, label='Training score')
    plt.plot(train_sizes, test_mean, label='Cross-validation score')

    plt.fill_between(train_sizes, train_mean - train_std,
                     train_mean + train_std, alpha=0.1)
    plt.fill_between(train_sizes, test_mean - test_std,
                     test_mean + test_std, alpha=0.1)

    plt.legend()
    st.pyplot()

#8. Вывести таблицу с результатами тестирования модели на отложенных данных:
model = LinearRegression()
model.fit(X_train, y_train)
if st.button('Показать таблицу с результатами'):
    # Создать таблицу с результатами
    results = pd.DataFrame({'y_test': y_test, 'y_pred':
model.predict(X_test)})
    st.write(results.head())

#9. Вывести примеры предсказаний модели на тестовых данных:

if st.button('Показать примеры предсказаний'):
    # Создать график предсказаний
    plt.scatter(y_test, model.predict(X_test))
    plt.title('Примеры предсказаний модели на тестовых данных')

```

```
plt.xlabel('Реальные значения')
plt.ylabel('Предсказанные значения')
st.pyplot()
```

Изображения веб-приложения

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of A
0	1	337	118	4	4.5	4.5	9.65	1	
1	2	324	107	4	4	4.5	8.87	1	
2	3	316	104	3	3	3.5	8	1	
3	4	322	110	3	3.5	2.5	8.67	1	
4	5	314	103	2	2	3	8.21	0	

Задайте гиперпараметры:

Количество эпох



Размер батча



Размер скрытого слоя



Размерность входных данных



Разделить данные на обучающую и тестовую выборки

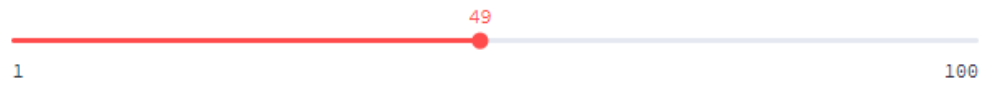
Обучить модель для линейной регрессии

Создаем объект классификатора

Показать график точности

График кривых и валидации

Количество эпох



Размер батча



Размер скрытого слоя



Размерность входных данных



Разделить данные на обучающую и тестовую выборки

Размер обучающей выборки:

(320, 8) (320,)

Размер тестовой выборки:

(80, 8) (80,)

which is not thread-safe.

To future-proof this code, you should pass in a figure as shown below:

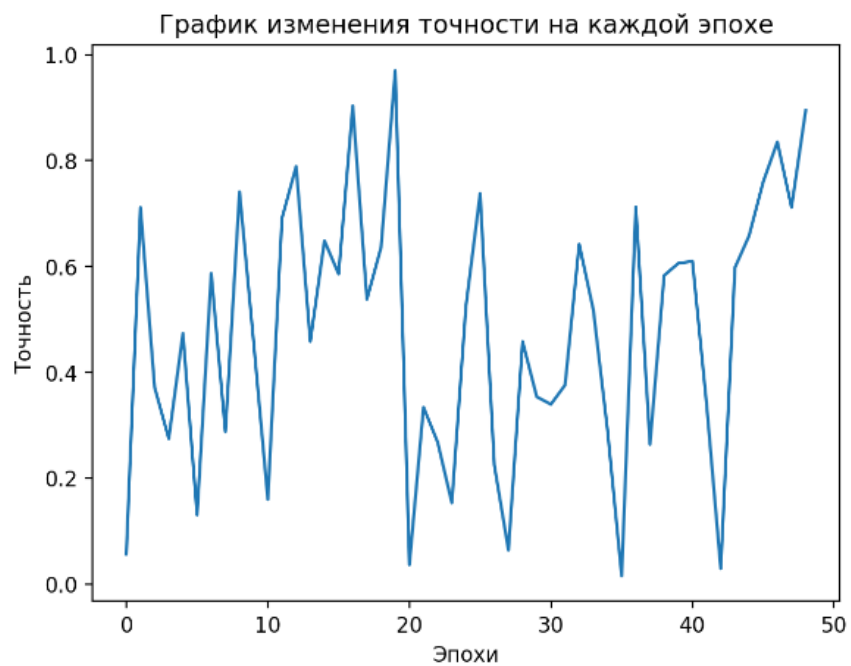
```
>>> fig, ax = plt.subplots()
>>> ax.scatter([1, 2, 3], [1, 2, 3])
>>> ... other plotting actions ...
>>> st.pyplot(fig)
```

You can disable this warning by disabling the config option: `deprecation.showPyplotGlobalUse`

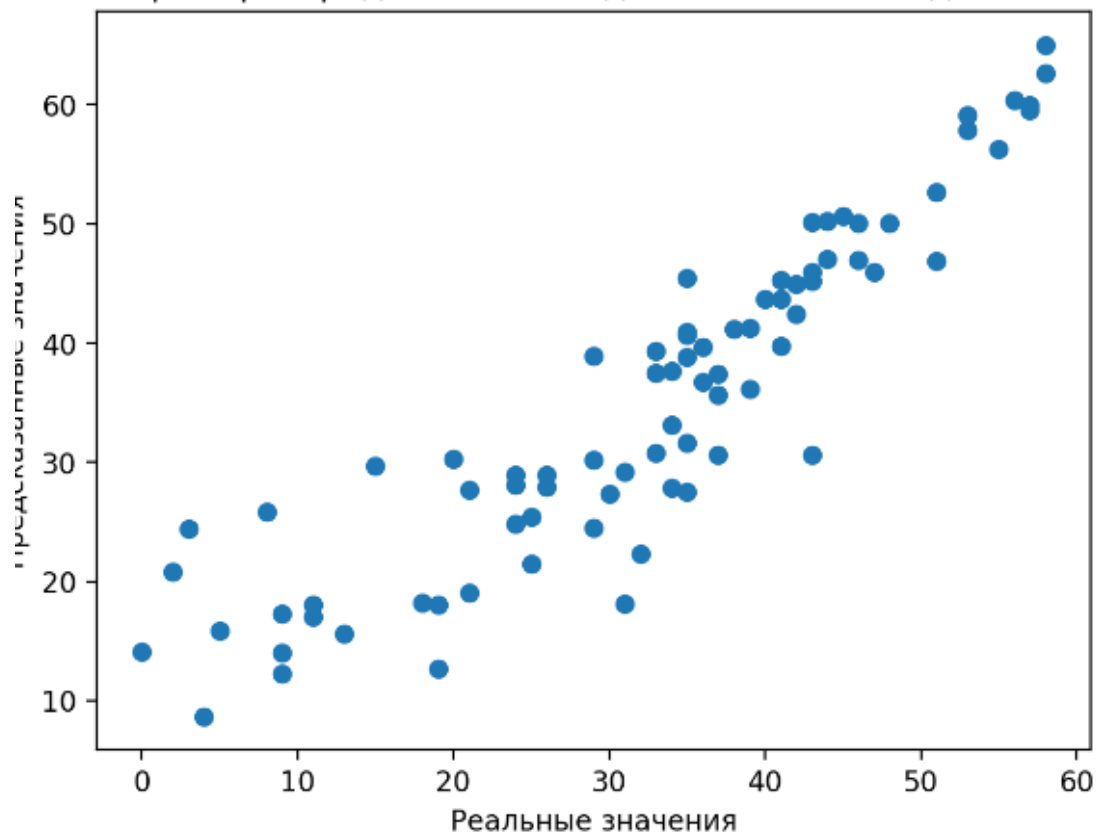
```
st.set_option('deprecation.showPyplotGlobalUse', False)
```

or in your `.streamlit/config.toml`

```
[deprecation]
showPyplotGlobalUse = false
```



Примеры предсказаний модели на тестовых данных



Разделить данные на обучающую и тестовую выборки

Обучить модель для линейной регрессии

Создаем объект классификатора

Показать график точности

График кривых и валидации

Показать таблицу с результатами

	y_test	y_pred
0	37	37.4292
1	35	38.8069
2	34	33.099
3	24	24.8389
4	29	30.2429