

**Московский государственный технический  
университет им. Н. Э. Баумана**

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

Выполнила:  
Шимолина П.К.,  
группа ИУ5-61Б

Проверил:  
Нардид А.Н.,  
каф. ИУ5

Дата:

Дата:

Подпись:

Подпись:

2023  
Москва

## Оглавление

Задание .....	3
Ход работы.....	3
Вывод.....	7

## Задание

Реализовать макет с использованием фреймворка streamlit для нескольких моделей машинного обучения. Макет должен позволять: выбирать модели для обучения, производить обучение, осуществлять просмотр результатов обучения, в том числе в виде графиков.

## Ход работы

Код представляет собой простое веб-приложение, разработанное с использованием библиотеки Streamlit, для тренировки и оценки различных моделей регрессии

### 1. Импорт необходимых библиотек:

- `streamlit` - библиотека для создания интерактивных веб-приложений.
- `pandas` - библиотека для работы с данными в табличной форме.
- `category\_encoders` - библиотека для кодирования категориальных признаков.
- `sklearn` - библиотека для машинного обучения, включая модели регрессии и метрики оценки.

```
import streamlit as st
import pandas as pd
import category_encoders as ce
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

### 2. Заголовок страницы и заголовок приложения.

```
st.set_page_config(page_title='Model Training', page_icon=':books:',
layout='wide')
```

```
st.title('Model Training App')
```

```
st.markdown('This app allows you to train and evaluate different regression models.')
```

### 3. Загрузка набора данных:

- `st.file_uploader` - виджет для загрузки файла данных.

- Если файл загружен успешно, то он считывается в DataFrame `df`.

```
uploaded_file = st.file_uploader("Choose a file")
```

```
if uploaded_file is not None:
```

```
    df = pd.read_csv(uploaded_file)
```

### 4. Предварительная обработка данных:

- `st.checkbox` и `st.multiselect` используются для выбора столбцов для анализа.

- `st.checkbox` используется для удаления строк с отсутствующими значениями.

- Категориальные столбцы кодируются с помощью `ce.BinaryEncoder`.

```
if st.checkbox('Select columns for analysis'):
```

```
    selected_columns = st.multiselect('Select columns:', df.columns)
```

```
    df = df[selected_columns]
```

```
if st.checkbox('Remove rows with missing values'):
```

```
    df.dropna(inplace=True)
```

```
cat_cols = list(df.select_dtypes(include=['object']).columns)
```

```
if len(cat_cols) > 0:
```

```
    encoder = ce.BinaryEncoder(cols=cat_cols)
```

```
    df = encoder.fit_transform(df)
```

### 5. Выбор целевого столбца:

- `st.selectbox` используется для выбора целевого столбца из `df`.

```
target_col = st.selectbox('Select target column:', df.columns)
```

## 6. Тренировка модели и отображение результатов:

- `st.selectbox` используется для выбора типа модели (линейная регрессия или случайный лес).
- `st.slider` используется для выбора размера тестовой выборки.
- Данные разбиваются на обучающую и тестовую выборки с помощью `train_test_split`.
- В зависимости от выбранного типа модели создается и обучается соответствующая модель.
- Вычисляются прогнозы на тестовых данных, и вычисляются метрики `'Mean Squared Error'` и `'R-squared'`.
- Графики сравнения фактических и предсказанных значений, а также важности признаков, отображаются с помощью `st.line_chart` и `st.bar_chart`.
- Результаты предсказания выводятся в виде таблицы.

```
model_type = st.selectbox('Select a model:', ['Linear Regression',  
                                              'Random Forest'])
```

```
test_size = st.slider('Select test size:', 0.1, 0.5, 0.2, 0.1)
```

```
X = df.drop(target_col, axis=1)
```

```
y = df[target_col]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size,  
                                                    random_state=42)
```

```
model = LinearRegression()
```

```
if st.button('Train Model'):
```

```
    if model_type == 'Linear Regression':
```

```

        model = LinearRegression()
    elif model_type == 'Random Forest':
        model = RandomForestRegressor()

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    st.write('Mean Squared Error:', mean_squared_error(y_test, y_pred))
    st.write('R-squared:', r2_score(y_test, y_pred))

    # графики
    st.write('Actual vs Predicted:')
    results_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
    st.line_chart(results_df)

    st.write('Feature Importances:')
    if model_type == 'Random Forest':
        importances = pd.Series(model.feature_importances_,
                                index=X_train.columns)
    else:
        importances = pd.Series(model.coef_, index=X_train.columns)
    importances.sort_values(ascending=False, inplace=True)
    st.bar_chart(importances)

    # add result table
    y_pred = model.predict(X_test)
    results = pd.concat([y_test.reset_index(drop=True), pd.Series(y_pred)],
                        axis=1)
    st.write('Prediction Results:')

```

```
st.dataframe(results)
```

## 7. Информация об авторе.

```
st.markdown("""  
*Created by Polina (https://github.com/shimolina-polina)*  
""")
```

Файл `requirements.txt` содержит список зависимостей, необходимых для запуска приложения, включая версии библиотек. Этот файл используется для установки зависимостей при развертывании приложения.

```
scikit-learn==1.2.2  
streamlit==1.21.0  
pandas==1.5.3  
category-encoders==2.6.0
```

## Вывод

В данной лабораторной работе было создано простое веб-приложение с использованием библиотеки Streamlit для тренировки и оценки моделей регрессии. Приложение позволяет загружать набор данных, выбирать столбцы для анализа, удалять строки с отсутствующими значениями, кодировать категориальные признаки, выбирать целевой столбец, а затем тренировать модели линейной регрессии и случайного леса.

После тренировки модели приложение выводит метрики оценки, такие как среднеквадратичная ошибка (Mean Squared Error) и коэффициент детерминации (R-squared). Также отображаются графики сравнения фактических и предсказанных значений, а также график важности признаков.

Пользовательский интерфейс приложения разработан с помощью простых элементов управления, таких как флажки, ползунки и выпадающие списки, что делает его удобным в использовании для тренировки и оценки моделей регрессии с различными параметрами и на разных данных.

Таким образом, данное веб-приложение облегчает процесс тренировки и оценки моделей регрессии, предоставляя удобный и интерактивный интерфейс для пользователей.