

Бекетов Роман ИУ5-62Б Лаб.6

```
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.datasets import load_digits
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

@st.cache_data
def load_data():
    digits = load_digits()
    data = pd.DataFrame(data=digits['data'],
        columns=digits['feature_names'])
    data['target'] = digits['target']
    return data

@st.cache_resource
def preprocess_data(data_in):
    """
    Масштабирование признаков, функция возвращает X и y для обучения
    """
    data_out = data_in.copy()
    # Масштабирование признаков
    scaler = StandardScaler()
    data_out[data_in.columns[:-1]] =
    scaler.fit_transform(data_out[data_in.columns[:-1]])
    return data_out, data_out['target']

# Загрузка и предварительная обработка данных
data = load_data()
data_X, data_y = preprocess_data(data)

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(data_X, data_y,
    test_size=0.2, random_state=42)

# Интерфейс пользователя
st.sidebar.header('Random Forest Classifier')
n_estimators_slider = st.sidebar.slider('Количество деревьев:',
    min_value=10, max_value=200, value=100, step=10)
max_depth_slider = st.sidebar.slider('Глубина дерева:', min_value=1,
    max_value=20, value=10, step=1)

# Обучение модели
```

```
model = RandomForestClassifier(n_estimators=n_estimators_slider,
max_depth=max_depth_slider, random_state=42)
model.fit(X_train, y_train)

# Предсказание на тестовой выборке
y_pred = model.predict(X_test)

# Оценка качества модели
st.subheader('Оценка качества модели')
st.write('Отчет о классификации:')
st.write(classification_report(y_test, y_pred))

# Визуализация матрицы ошибок
st.subheader('Матрица ошибок')
conf_mat = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues', ax=ax)
st.pyplot(fig)
```