



**«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

О т ч е т

по лабораторной работе №6

Создание веб-приложения для демонстрации моделей машинного обучения

Дисциплина: Технологии машинного обучения

Студент гр. ИУ5-63Б

(Подпись, дата)

Терентьев В.О.

(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Гапанюк Ю.Е.

(Фамилия И.О.)

Москва, 2021

1. Цель работы

Цель лабораторной работы: изучение возможностей демонстрации моделей машинного обучения с помощью веб-приложений.

2. Описание задания

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.

3. Текст программы

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics import plot_confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from operator import itemgetter

@st.cache
def load_data():
    """
    Загрузка данных
    """
    org_data = pd.read_csv('train.csv')
    org_data.drop(columns=['Unnamed: 0'], inplace=True)
    return org_data
```

```

def preprocess_data(data_in, rows):
    """
    Разделение выборки на обучающую и тестовую
    """
    data_train, data_test, data_y_train, data_y_test = train_test_split(data_in[0:rows][data_in.columns.drop('isFraud')], data_in[0:rows]['isFraud'], random_state=1)
    return data_train, data_test, data_y_train, data_y_test

# Отрисовка ROC-кривой
def draw_roc_curve(y_true, y_score, ax, pos_label=1, average='micro'):
    fpr, tpr, thresholds = roc_curve(y_true, y_score, pos_label=pos_label)
    roc_auc_value = roc_auc_score(y_true, y_score, average=average)
    #plt.figure()
    lw = 2
    ax.plot(fpr, tpr, color='darkorange', lw=lw, label='ROC curve (area = %0.2f)' % roc_auc_value)
    ax.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
    ax.set_xlim([0.0, 1.0])
    ax.set_xlim([0.0, 1.05])
    ax.set_xlabel('False Positive Rate')
    ax.set_ylabel('True Positive Rate')
    ax.set_title('Receiver operating characteristic')
    ax.legend(loc="lower right")

st.title('Гиперпараметры моделей')
data = load_data()
row_slider = st.sidebar.slider('Количество строк в датасете:', min_value=10000, max_value=data.shape[0]-1, value=10000, step=1000)
data_train, data_test, data_y_train, data_y_test = preprocess_data(data, row_slider)
st.sidebar.header('Выберите модель:')
sb = st.sidebar.selectbox('Модель:', ('Метод ближайших соседей', 'Решающее дерево'))
if sb == 'Метод ближайших соседей':
    neig_slider = st.sidebar.slider('Количество ближайших соседей:', min_value=1, max_value=100, value=1, step=1)

    KNC = KNeighborsClassifier(n_neighbors=neig_slider, n_jobs=-1).fit(data_train, data_y_train)
    data_test_predicted_knc = KNC.predict_proba(data_test)[:,-1]

    st.subheader('Оценка качества модели:')
    fig, ax = plt.subplots(ncols=2, figsize=(10,5))
    draw_roc_curve(data_y_test, data_test_predicted_knc, ax[0])
    plot_confusion_matrix(KNC, data_test, data_y_test, ax=ax[1], display_labels=['0', '1'], cmap=plt.cm.Blues, normalize='true')
    st.pyplot(fig)

```

```

if sb == 'Решающее дерево':
    max_depth = st.sidebar.slider('Глубина дерева:', min_value=1, max_value=100,
value=50, step=1)
    max_features = st.sidebar.slider('Число признаков для выбора расщепления:', m
in_value=1, max_value=30, value=14, step=1)
    min_samples_leaf = st.sidebar.slider('Минимальное количество выборок:', min_v
alue=1, max_value=5, value=1, step=1)

    dtc = DecisionTreeClassifier(random_state=1, max_depth=max_depth, max_feature
s=max_features, min_samples_leaf=min_samples_leaf).fit(data_train, data_y_train)
    data_test_predicted_dtc = dtc.predict_proba(data_test)[:,:1]

    st.subheader('Оценка качества модели:')
    fig, ax = plt.subplots(ncols=2, figsize=(10,5))
    draw_roc_curve(data_y_test, data_test_predicted_dtc, ax[0])
    plot_confusion_matrix(dtc, data_test, data_y_test, ax=ax[1],
display_labels=['0', '1'],
cmap=plt.cm.Blues, normalize='true')
    st.pyplot(fig)

    st.subheader('Важность признаков:')
    # Сортировка значений важности признаков по убыванию
    list_to_sort = list(zip(data_train.columns.values, dtc.feature_importances_))
    sorted_list = sorted(list_to_sort, key=itemgetter(1), reverse = True)
    # Названия признаков
    labels = [x for x,_ in sorted_list]
    # Важности признаков
    data = [x for _,x in sorted_list]
    # Вывод графика
    fig, ax = plt.subplots(figsize=(18,5))
    ind = np.arange(len(labels))
    plt.bar(ind, data)
    plt.xticks(ind, labels, rotation='vertical')
    # Вывод значений
    for a,b in zip(ind, data):
        plt.text(a-0.05, b+0.01, str(round(b,3)))
    st.pyplot(fig)

    st.subheader('Список признаков, отсортированный на основе важности:')
    labels

```

4. Экранные формы с примерами выполнения программы

Количество строк в датасете:
100000
100000 590539

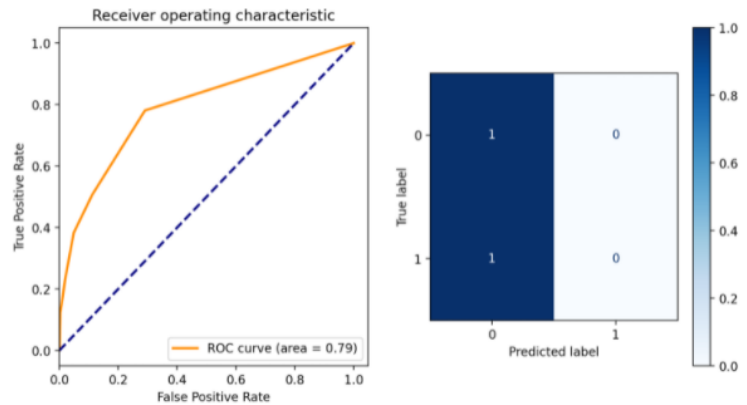
Выберите модель:

Модель:
Метод ближайших соседей

Количество ближайших соседей:
20
1 100

Гиперпараметры моделей

Оценка качества модели:



Количество строк в датасете:
100000
100000 590539

Выберите модель:

Модель:
Решающее дерево

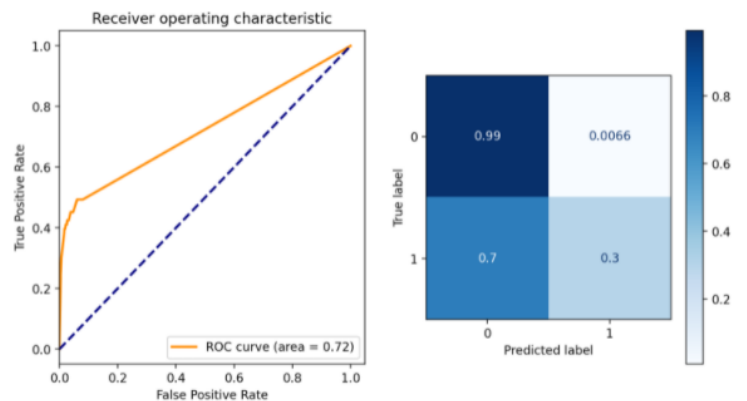
Глубина дерева:
90
1 100

Число признаков для выбора расщепления:
14
1 30

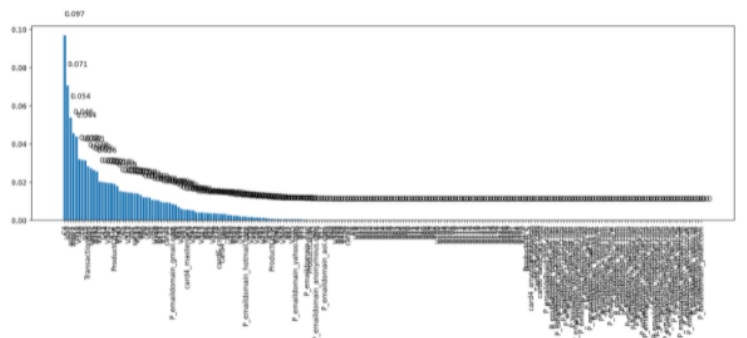
Минимальное количество выборок:
2
1 5

Гиперпараметры моделей

Оценка качества модели:



Важность признаков:



Список признаков, отсортированный на основе важности:

0 : "C4"
1 : "V53"