

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2022 г.

"__" _____ 2022 г.

**Отчет по лабораторной работе № 6 по курсу
Технологии машинного обучения
ГУИМЦ**

**Тема работы: " Создание веб-приложения для демонстрации
моделей машинного обучения "**

18

(количество листов)

Вариант № 3

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-82Б

Чиварзин А.Е.

(подпись)

"__" _____ 2022 г.

СОДЕРЖАНИЕ

1. Описание задания	3
2. Текст программы	4
3. Результаты выполнения программы	9

1. 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.

Для разработки рекомендуется использовать следующие (или аналогичные) фреймворки:

- [streamlit](#)
- [gradio](#)
- [dash](#)

2. Текст программы

```
import os
import subprocess

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import streamlit as st
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR

def about():
    st.header('Лабораторная работа №6')
    st.write('Чиварзин А. Е. ИУ5Ц-82Б')
    st.markdown('-----')

@st.cache
def load_data():
    '''
    Загрузка данных
    '''
    data_original = pd.read_csv('data/cwurData.csv', sep=",")
    return data_original

@st.cache
def delete_NULLs(data_in):
    data_out = data_in.copy()
    # Удаление строк, содержащих пустые значения
    data_out = data_out.dropna(axis=0, how='any')
    return data_out

@st.cache
def preprocess_data(data_in):
    '''
    Масштабирование и кодирование признаков, функция возвращает X и y
    для кросс-валидации
    '''
    data_out = data_in.copy()
    le = LabelEncoder()
    institution_le = le.fit_transform(data_out['institution'])
    le_country = LabelEncoder()
    country_le = le_country.fit_transform(data_out['country'])
    data_digit = data_out.copy()
    data_digit["institution"] = institution_le
    data_digit['country'] = country_le
```

```

sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data_digit[['broad_impact']])
sc2 = MinMaxScaler()
sc2_data = sc2.fit_transform(data_digit[['institution']])
sc3 = MinMaxScaler()
sc3_data = sc3.fit_transform(data_digit[['country']])
sc4 = MinMaxScaler()
sc4_data = sc4.fit_transform(data_digit[['national_rank']])
sc5 = MinMaxScaler()
sc5_data = sc5.fit_transform(data_digit[['quality_of_education']])
sc6 = MinMaxScaler()
sc6_data = sc6.fit_transform(data_digit[['alumni_employment']])
sc7 = MinMaxScaler()
sc7_data = sc7.fit_transform(data_digit[['quality_of_faculty']])
sc8 = MinMaxScaler()
sc8_data = sc8.fit_transform(data_digit[['publications']])
sc9 = MinMaxScaler()
sc9_data = sc9.fit_transform(data_digit[['influence']])
sc10 = MinMaxScaler()
sc10_data = sc10.fit_transform(data_digit[['citations']])
sc11 = MinMaxScaler()
sc11_data = sc11.fit_transform(data_digit[['broad_impact']])
sc12 = MinMaxScaler()
sc12_data = sc12.fit_transform(data_digit[['patents']])
sc13 = MinMaxScaler()
sc13_data = sc13.fit_transform(data_digit[['score']])
sc14 = MinMaxScaler()
sc14_data = sc14.fit_transform(data_digit[['year']])

data_normal = data_digit.copy()
data_normal['world_rank'] = sc1_data
data_normal['institution'] = sc2_data
data_normal['country'] = sc3_data
data_normal['national_rank'] = sc4_data
data_normal['quality_of_education'] = sc5_data
data_normal['alumni_employment'] = sc6_data
data_normal['quality_of_faculty'] = sc7_data
data_normal['publications'] = sc8_data
data_normal['influence'] = sc9_data
data_normal['citations'] = sc10_data
data_normal['broad_impact'] = sc11_data
data_normal['patents'] = sc12_data
data_normal['score'] = sc13_data
data_normal['year'] = sc14_data
data_out = data_normal
return pd.DataFrame(data_out).drop(['world_rank'], axis=1),
data_out['world_rank']

```

```

#####
#####

```

```

def KNN(x, y):

```

```

    reg_gs = GridSearchCV(KNeighborsRegressor(), tuned_parameters,
cv=cv_slider, scoring='neg_median_absolute_error')
    reg_gs.fit(x, y)

    st.subheader('Оценка качества модели')

    st.write('Лучшее значение параметров -
{}'.format(reg_gs.best_params_))

    # Изменение качества на тестовой выборке в зависимости от K-соседей
    fig1 = plt.figure(figsize=(7, 5))
    ax = plt.plot(n_range, reg_gs.cv_results_['mean_test_score'])
    plt.xlabel('Количество соседей')
    st.pyplot(fig1)

def SVM(X: pd.DataFrame, Y):
    if X.shape[1] == 0:
        st.write('Ни один столбец не выбран')
    else:
        gp = st.selectbox('Гиперпараметр для построения', X.columns)
        X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
random_state=2022, test_size=0.1)
        svr = SVR(kernel='linear')
        svr.fit(X_train, Y_train)
        pred_y = svr.predict(X_test)
        fig2 = plt.figure(figsize=(7, 5))
        plt.scatter(X_test[gp], Y_test, marker='s', label='Тестовая
выборка')
        plt.scatter(X_test[gp], pred_y, marker='o', label='Предсказанные
данные')
        plt.legend(loc='lower right')
        plt.xlabel('рейтинг за широкое влияние')
        plt.ylabel('Целевой признак')
        st.pyplot(fig2)

about()
use_msg = True
if os.name == 'nt':
    if st.sidebar.checkbox('Оповещать об окончании обучения',
value=True):
        use_msg = True
    else:
        use_msg = False

data = load_data()
data_no_null = delete_NULLs(data)

if st.checkbox('Показать данные'):
    st.write(data_no_null)

if st.checkbox('Показать парные диаграммы (+ 30 секунд)':
    st.pyplot(sns.pairplot(data_no_null, height=5))

```

```

st.write('Разглядеть не получится. Известный баг:
https://github.com/streamlit/streamlit/issues/796')

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

st.markdown('-----
-----')

st.sidebar.header('Метод ближайших соседей')
#####
#
do_analyze_KNN = False # По-умолчанию отключим анализ данных, чтобы не
ждать 15 секунд - 5 минут!!!
if st.sidebar.checkbox('Показать гиперпараметры'):
    cv_slider = st.sidebar.slider('Количество фолдов:', min_value=2,
max_value=10, value=3, step=1)
    step_slider = st.sidebar.slider('Шаг для соседей:', min_value=1,
max_value=50, value=10, step=1)
    # Количество записей
    data_len = data.shape[0]
    data_no_null_len = data_no_null.shape[0]
    # Вычислим количество возможных ближайших соседей
    rows_in_one_fold = int(data_no_null_len / cv_slider)
    allowed_knn = int(rows_in_one_fold * (cv_slider - 1))
    st.write('Количество строк в наборе данных до очистки строк -
{}'.format(data_len))
    st.write('Количество строк в наборе данных после очистки строк -
{}'.format(data_no_null_len))
    st.write('Максимальное допустимое количество ближайших соседей с
учётом выбранного количества фолдов - {}'.format(allowed_knn))
    # Подбор гиперпараметра
    n_range_list = list(range(1, allowed_knn, step_slider))
    n_range = np.array(n_range_list)
    st.write('Возможные значения соседей - {}'.format(n_range))
    tuned_parameters = [{'n_neighbors': n_range}]

    if st.sidebar.checkbox('Выполнять обучение'):
        do_analyze_KNN = True
    else:
        do_analyze_KNN = False
    st.sidebar.write(
        '⚠ Устанавливайте этот флажок только после окончательной
установки гиперпараметров\n'
        'После установки флажка рекомендуется идти пить чай на 30
секунд - 30 минут...')

st.sidebar.header('SVM')
#####
#####
do_analyze_SVM = False
show_SVM_params = False

```

```

if st.sidebar.checkbox('Показать гиперпараметры '):
    show_SVM_params = True
    if st.sidebar.checkbox('Выполнять обучение'):
        do_analyze_SVM = True
    else:
        do_analyze_SVM = False
else:
    show_SVM_params = False

data_X, data_y = preprocess_data(data_no_null)

if do_analyze_KNN:
    KNN(data_X, data_y)

if show_SVM_params:
    columns = st.multiselect('Столбцы для построения модели',
data_X.columns)
    if do_analyze_SVM:
        SVM(data_X[columns], data_y)

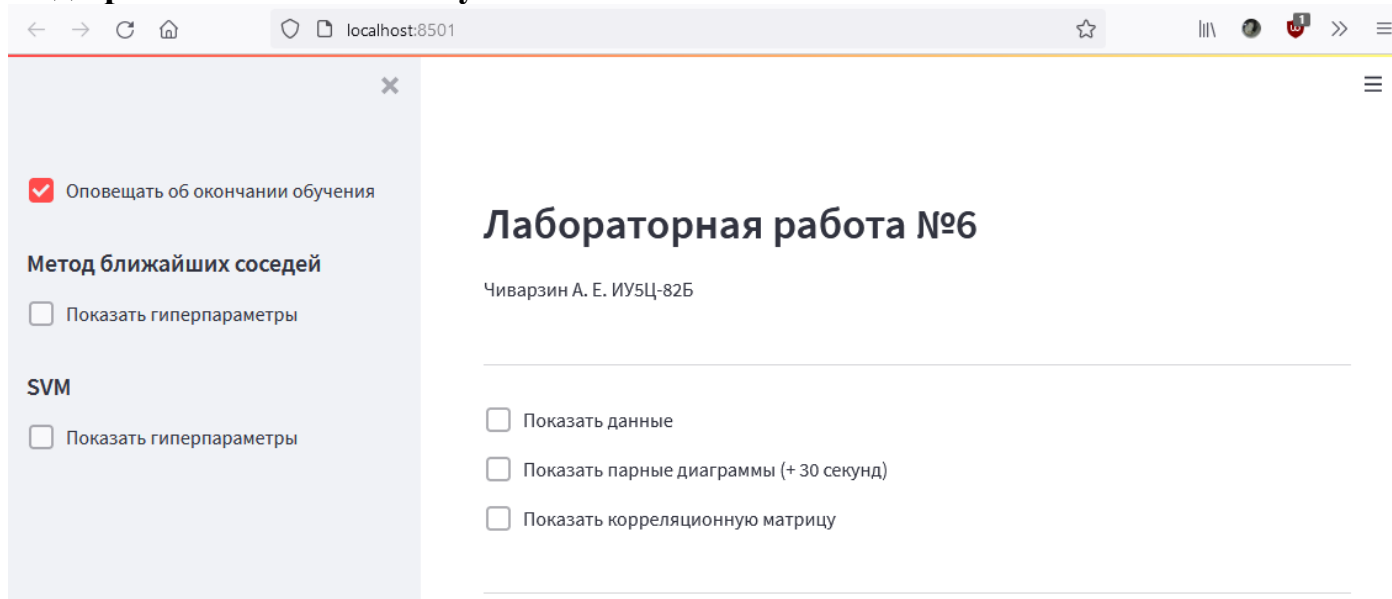
# Выводим пользователю сообщение об окончании работы скрипта (на
редакции Enterprise multi-session оно полноэкранное)
# Работает только на Windows
if do_analyze_KNN or do_analyze_SVM:
    if os.name == 'nt' and use_msg:
        cmd_output = subprocess.check_output(['cmd', '/c chcp 65001 &
msg * Модель успешно построена!'])

if __name__ == '__main__':
    pass

```

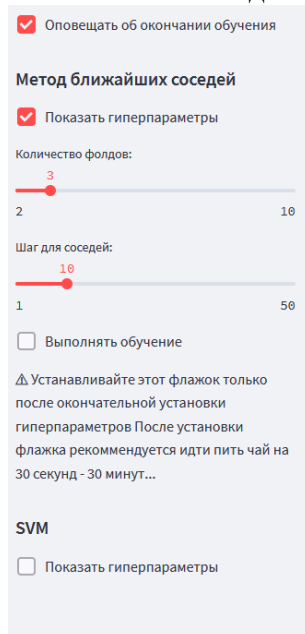

3. Результаты выполнения программы

Вид приложения после запуска



Левая панель содержит выбор моделей машинного обучения. На правой панели можно вывести исходные данные, парные диаграммы и показать корреляционную матрицу. На левой панели можно включить или отключить показ сообщения об успешном завершении обучения, что полезно при

Ближайшие соседи



Лабораторная работа №6

Чиварзин А. Е. ИУ5Ц-82Б

- ☐ Показать данные
- ☐ Показать парные диаграммы (+ 30 секунд)
- ☐ Показать корреляционную матрицу

Количество строк в наборе данных до очистки строк - 2200

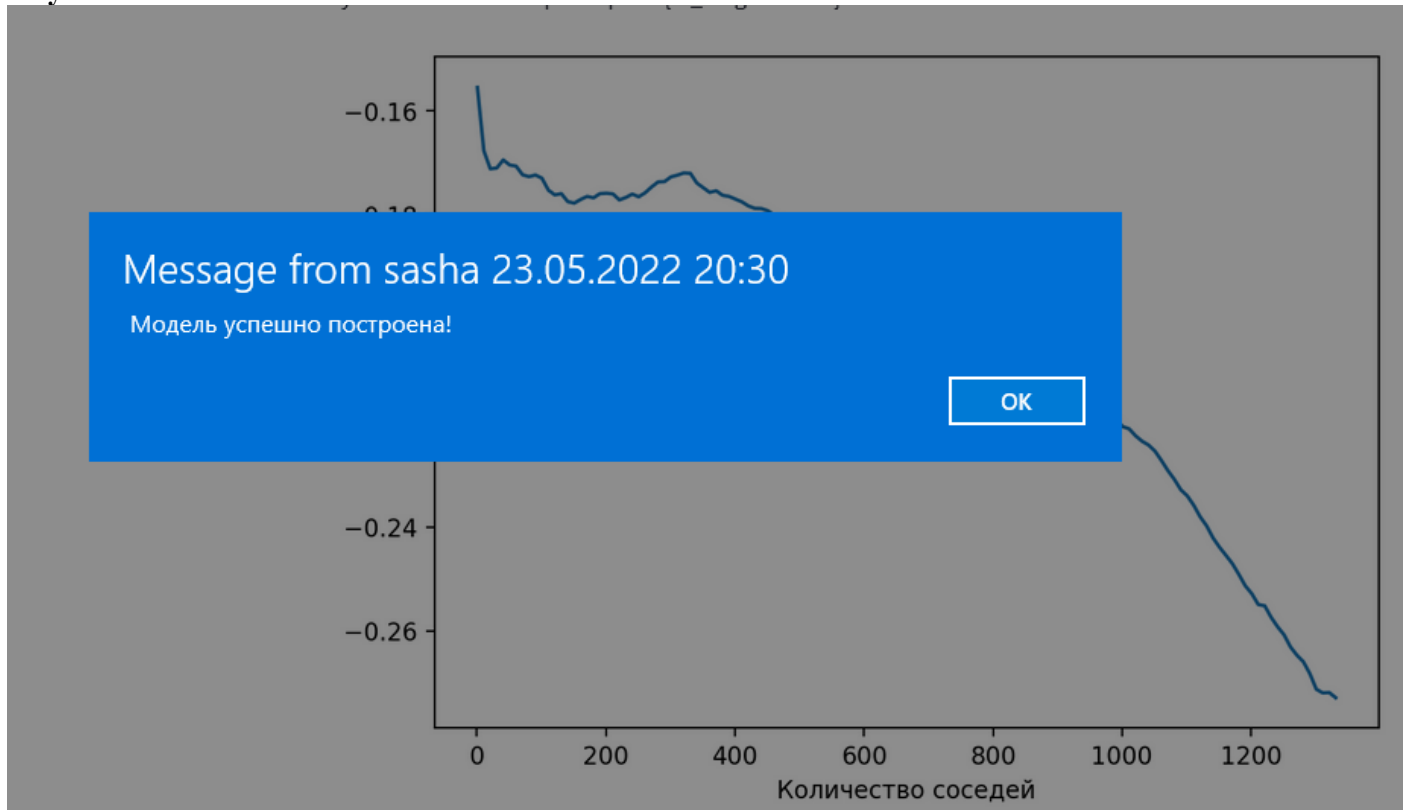
Количество строк в наборе данных после очистки строк - 2000

Максимальное допустимое количество ближайших соседей с учётом выбранного количества фолдов - 1332

Возможные значения соседей - [1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251 261 271 281 291 301 311 321 331 341 351 361 371 381 391 401 411 421 431 441 451 461 471 481 491 501 511 521 531 541 551 561 571 581 591 601 611 621 631 641 651 661 671 681 691 701 711 721 731 741 751 761 771 781 791 801 811 821 831 841 851 861 871 881 891 901 911 921 931 941 951 961 971 981 991 1001 1011 1021 1031 1041 1051 1061 1071 1081 1091 1101 1111 1121 1131 1141 1151 1161 1171 1181 1191 1201 1211 1221 1231 1241 1251 1261 1271 1281 1291 1301 1311 1321 1331]

Поле включения флажка «Показать гиперпараметры» на левой панели появляется выбор гиперпараметров для KNN, а на правой панели выводятся информация, такая как список из количества соседей,

Обучение KNN



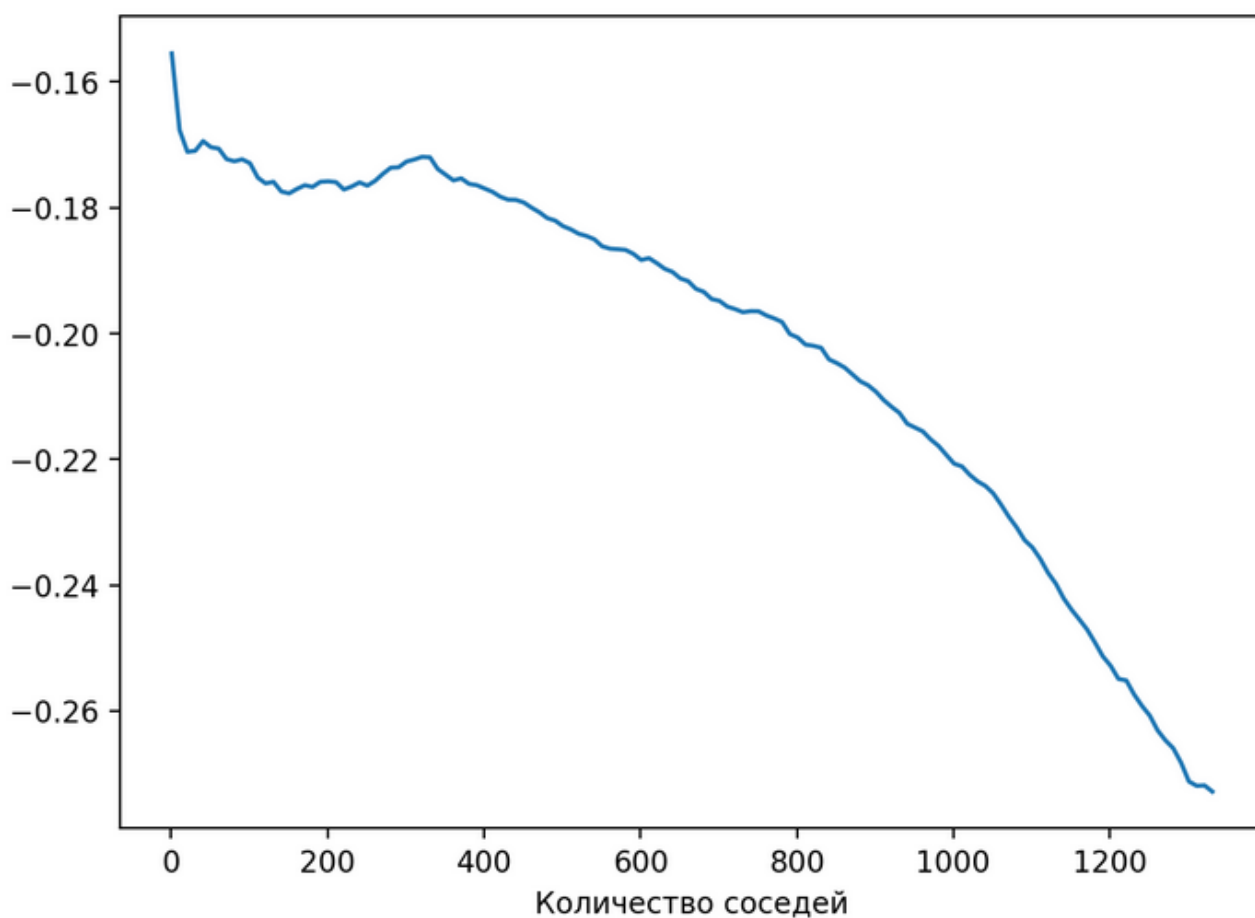
После установки флажка «Выполнять обучение» было выполнено обучение модели, которое длилось 21 секунду, после чего было получено полноэкранное сообщение. (Если на ОС не активна роль RDS, то будем маленькое окно с тем же текстом)

На следующей странице отчёта представлен полученный график (без перекрытия).

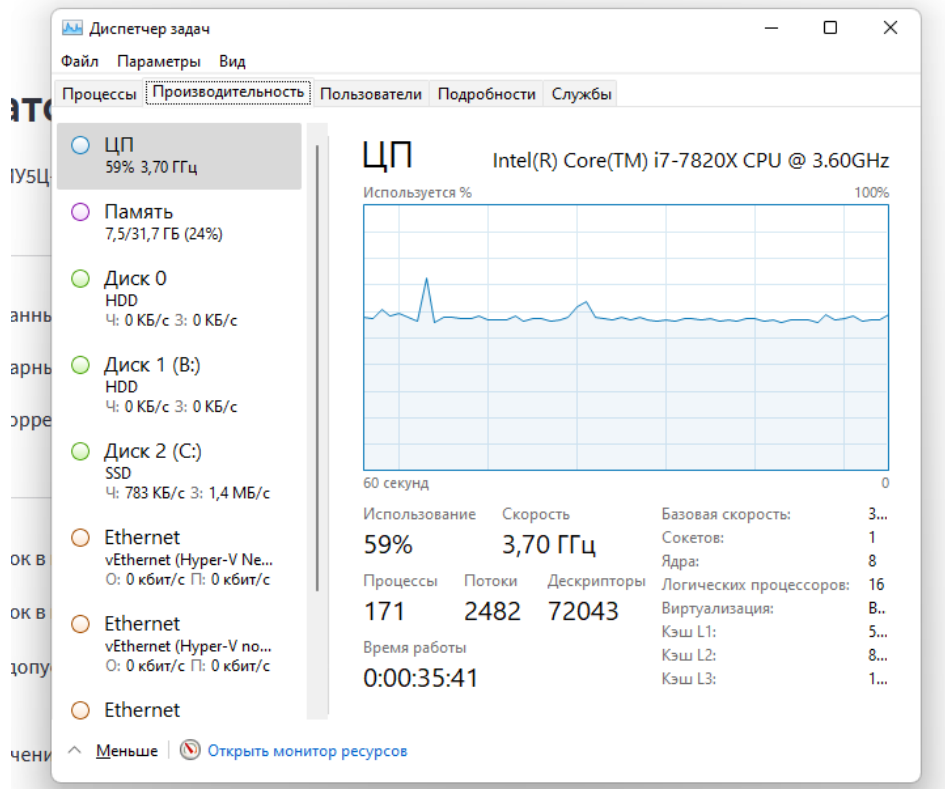
Возможные значения соседей - [1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191
201 211 221 231 241 251 261 271 281 291 301 311 321 331 341 351 361 371 381 391 401 411 421 431 441 451
461 471 481 491 501 511 521 531 541 551 561 571 581 591 601 611 621 631 641 651 661 671 681 691 701 711
721 731 741 751 761 771 781 791 801 811 821 831 841 851 861 871 881 891 901 911 921 931 941 951 961 971
981 991 1001 1011 1021 1031 1041 1051 1061 1071 1081 1091 1101 1111 1121 1131 1141 1151 1161 1171
1181 1191 1201 1211 1221 1231 1241 1251 1261 1271 1281 1291 1301 1311 1321 1331]

Оценка качества модели

Лучшее значение параметров - {'n_neighbors': 1}



Если выставить шаг соседей = 1 и количество фолдов = 4, то придётся ждать 4 минуты 30 секунд. При этом в диспетчере задач ЦП будет загружен на 59% при этом на машине всего один активный сеанс (console), отключённых сеансов тоже нет.

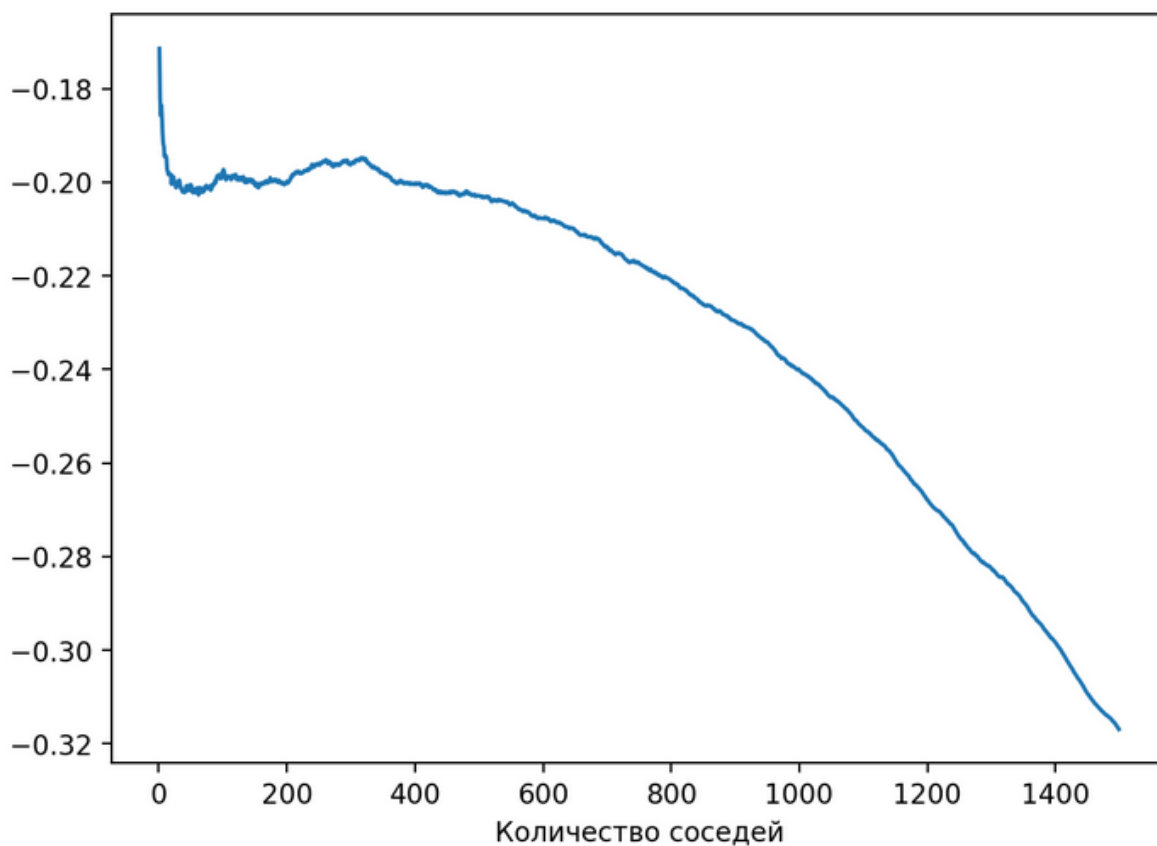


Ниже представлен получившийся график

Возможные значения соседей - [1 2 3 ... 1497 1498 1499]

Оценка качества модели

Лучшее значение параметров - {'n_neighbors': 1}



SVM

Были сняты все флажки в категории «Метод ближайших соседей». Оповещение об окончании обучения также отключено из-за отсутствия длительных ожиданий.

После установки флажка «Показать гиперпараметры» в SVM приложение предложит выбрать столбцы для построения модели:

☐ Оповещать об окончании обучения

Метод ближайших соседей

☐ Показать гиперпараметры

SVM

☒ Показать гиперпараметры

☐ Выполнять обучение

Лабораторная работа №6

Чиварзин А. Е. ИУ5Ц-82Б

☐ Показать данные

☐ Показать парные диаграммы (+ 30 секунд)

☐ Показать корреляционную матрицу

Столбцы для построения модели

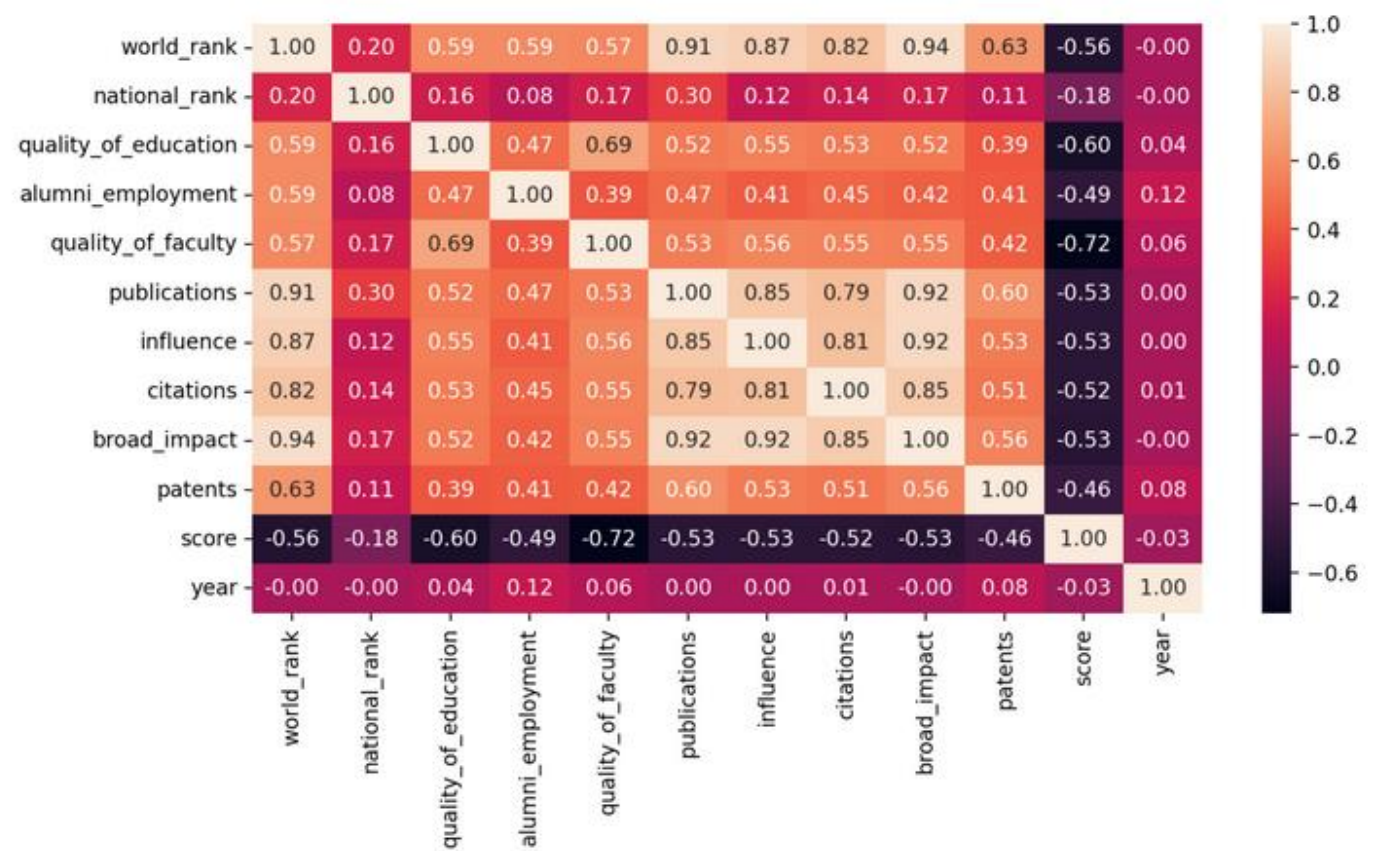
Choose an option

Для того, чтобы сделать более логичный выбор, в приложении имеется возможность отобразить корреляционную матрицу. Для этого установим флажок «Показать корреляционную матрицу».

Лабораторная работа №6

Чиварзин А. Е. ИУ5Ц-82Б

- ☐ Показать данные
- ☐ Показать парные диаграммы (+ 30 секунд)
- ☒ Показать корреляционную матрицу



Столбцы для построения модели

Choose an option

Как видно из матрицы, более всего с целевым признаком «world_rank» коррелируют broad_impact и publications. Отключим вывод матрицы и построим модель по этим признакам.

SVM

☒ Показывать гиперпараметры

☐ Выполнять обучение

☐ Показывать данные

☐ Показывать парные диаграммы (+ 30 секунд)

☐ Показывать корреляционную матрицу

Столбцы для построения модели

broad_impact ✕

publications ✕

После установки параметров отметим флажок «Выполнять обучение».

Столбцы для построения модели

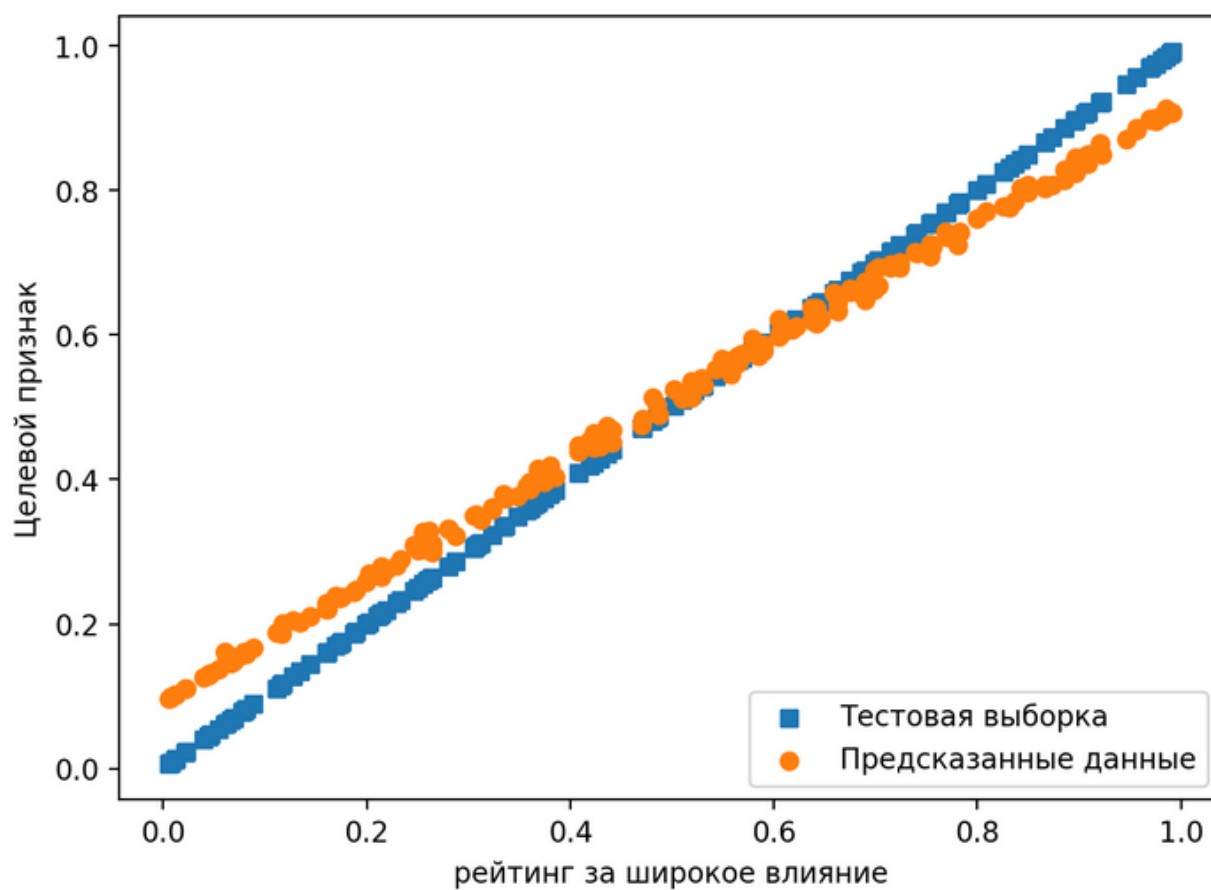
broad_impact ✕

publications ✕



Гиперпараметр для построения

broad_impact



Имеется возможность изменить гиперпараметр построения

Столбцы для построения модели

broad_impact ✕

publications ✕

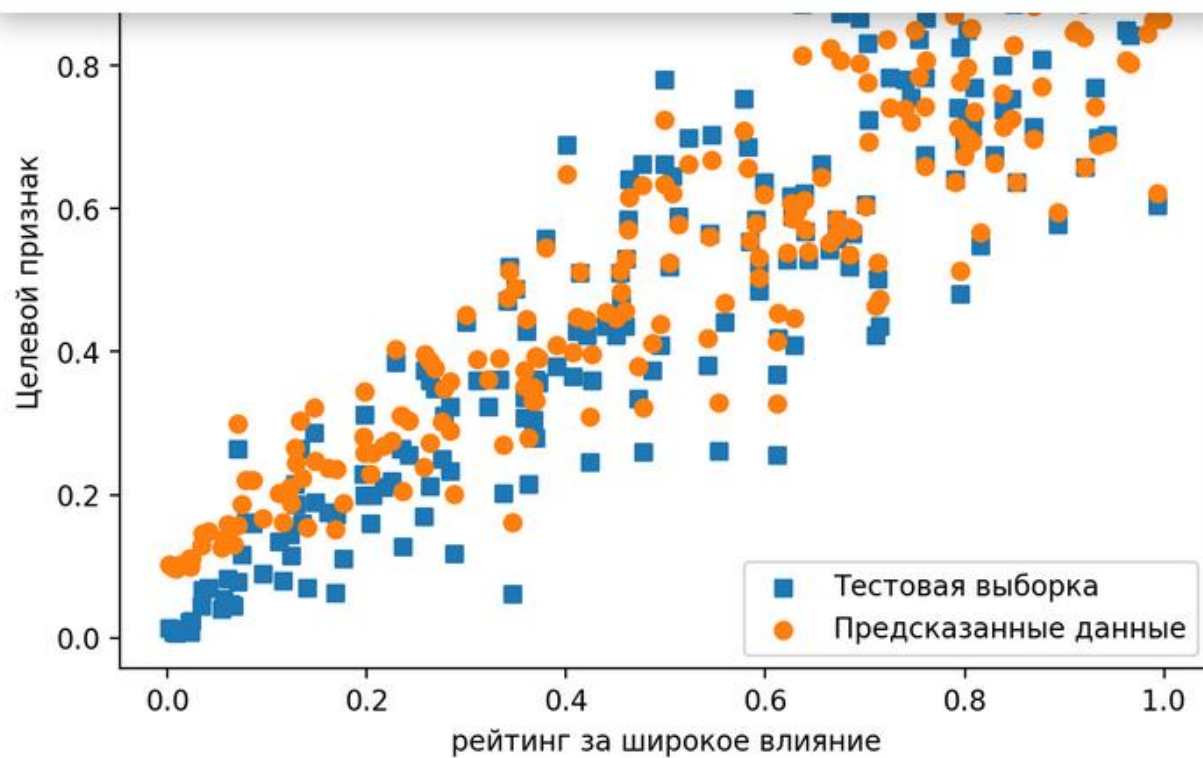


Гиперпараметр для построения

publications

broad_impact

publications



Были сняты все флажки и установлен флажок «Показать данные»

Чиварзин А. Е. ИУ5Ц-82Б

☐ Показать парные диаграммы (+ 30 секунд)

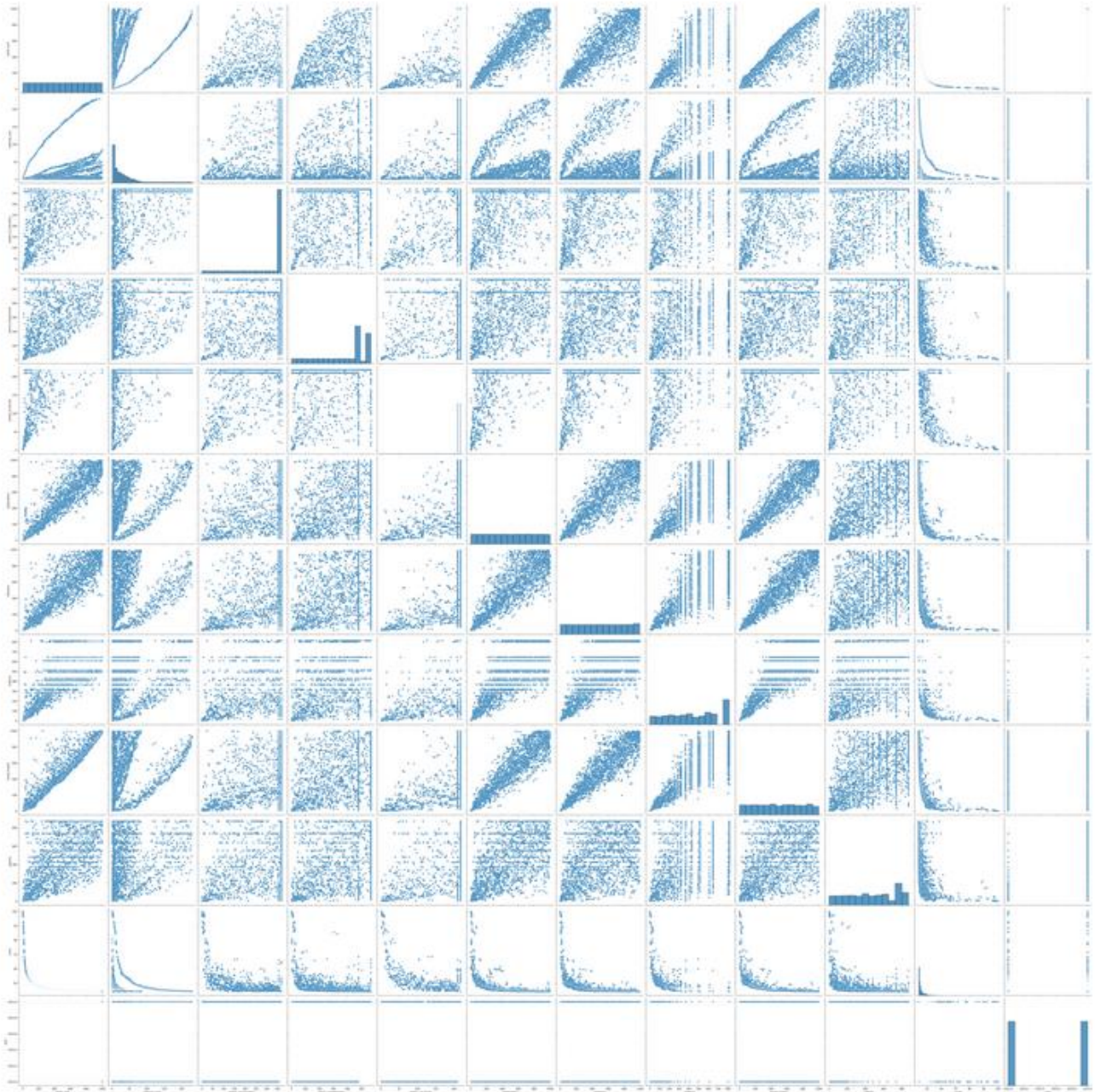
.

Вывод парных диаграмм

Были сняты все флажки и установлен флажок «Показать парные диаграммы»

☐ Показать данные

☒ Показать парные диаграммы (+ 30 секунд)



Разглядеть не получится. Известный баг: <https://github.com/streamlit/streamlit/issues/796>

☐ Показать корреляционную матрицу