Защищено:
Гапанюк Ю.Е.

"__"_____2022 г.

Демонстрация:
Гапанюк Ю.Е.

"__"_____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 _____
(подпись)

Чиварзин А.Е.

"__"_____2022 г.

Москва, МГТУ - 2022

СОДЕРЖАНИЕ

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):
```

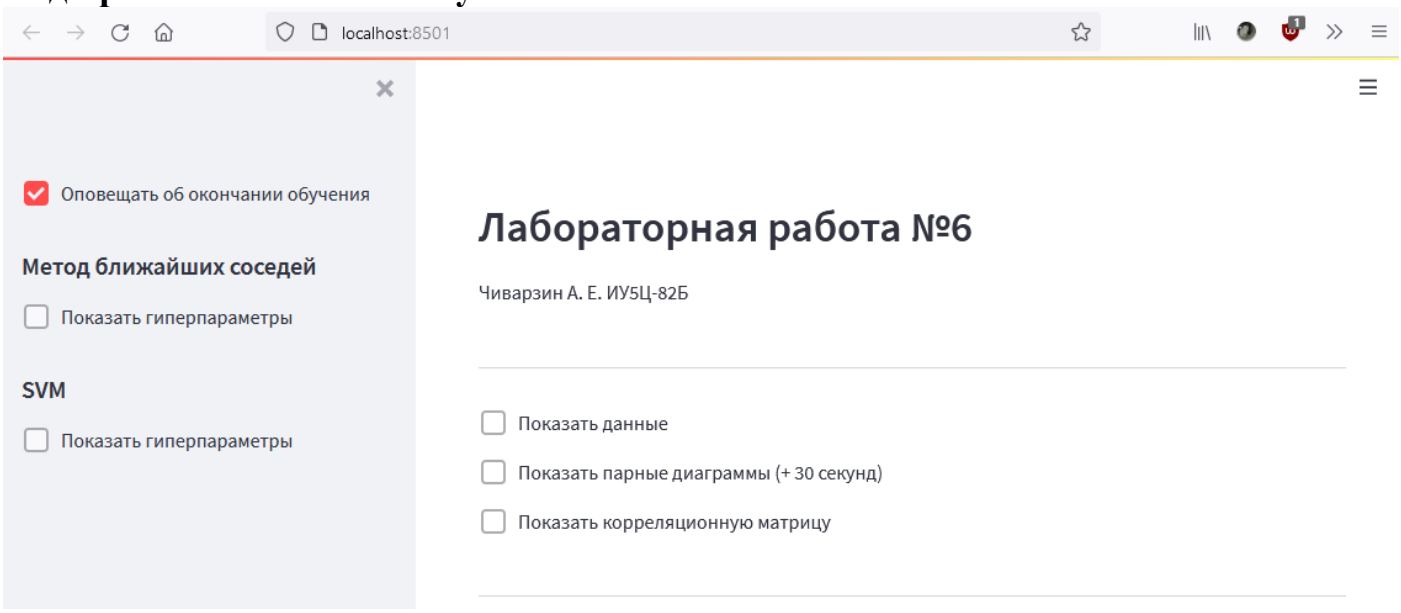
```
    reg_gs = GridSearchCV(KNeighborsRegressor(), tuned_parameters,
cv=cv_slider, scoring='neg_median_absolute_error')
    reg_gs.fit(x, y)

    st.subheader('Оценка качества модели')

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

    # Изменение качества на тестовой выборке в зависимости от К-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_analize_KNN = False  # По-умолчанию отключим анализ данных, чтобы не
ждать 15 секунд - 5 ммнут!!!
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_analize_KNN = True
    else:
        do_analize_KNN = False
    st.sidebar.write(
        '⚠ Устанавливайте этот флажок только после окончательной
установки гиперпараметров\n'
        'После установки флажка рекоммендуется идти пить чай на 30
секунд - 30 минут...')

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

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

data_X, data_y = preprocess_data(data_no_null)

if do_analize_KNN:
    KNN(data_X, data_y)

if show_SVM_params:
    columns = st.multiselect('Столбцы для построения модели',
data_X.columns)
    if do_analize_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_analize_KNN or do_analize_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
```

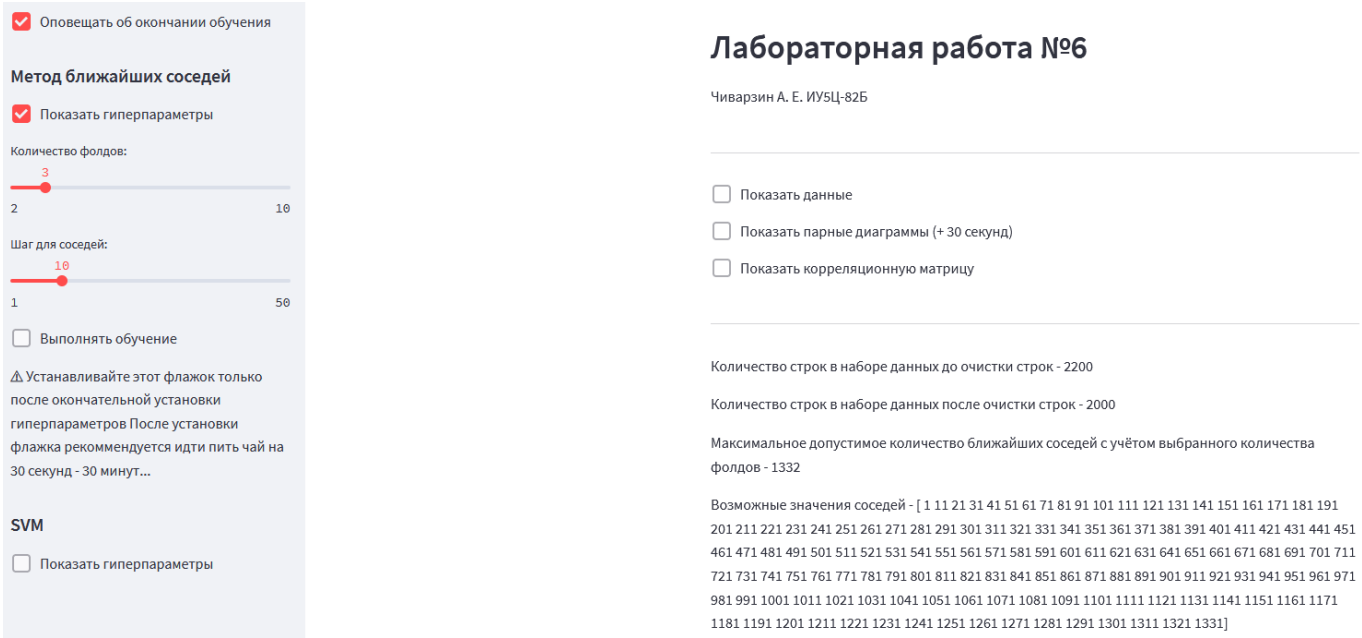
3. Результаты выполнения программы

Вид приложения после запуска



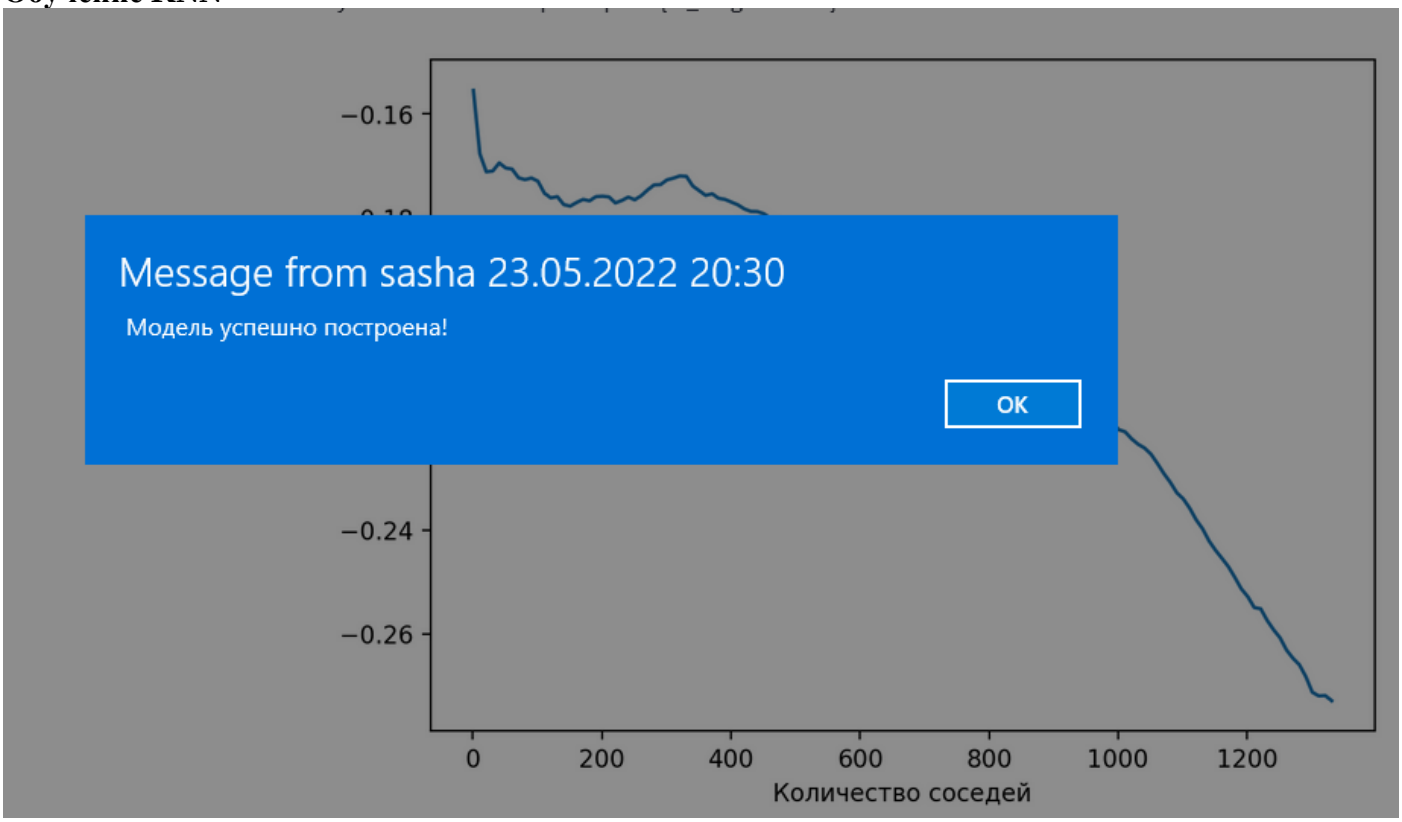
Левая панель содержит выбор модели машинного обучения. На правой панели можно вывести исходные данные, парные диаграммы и показать корреляционную матрицу.
На левой панели можно включить или отключить показ сообщения об успешном завершении обучения, что полезно при

Ближайшие соседи



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



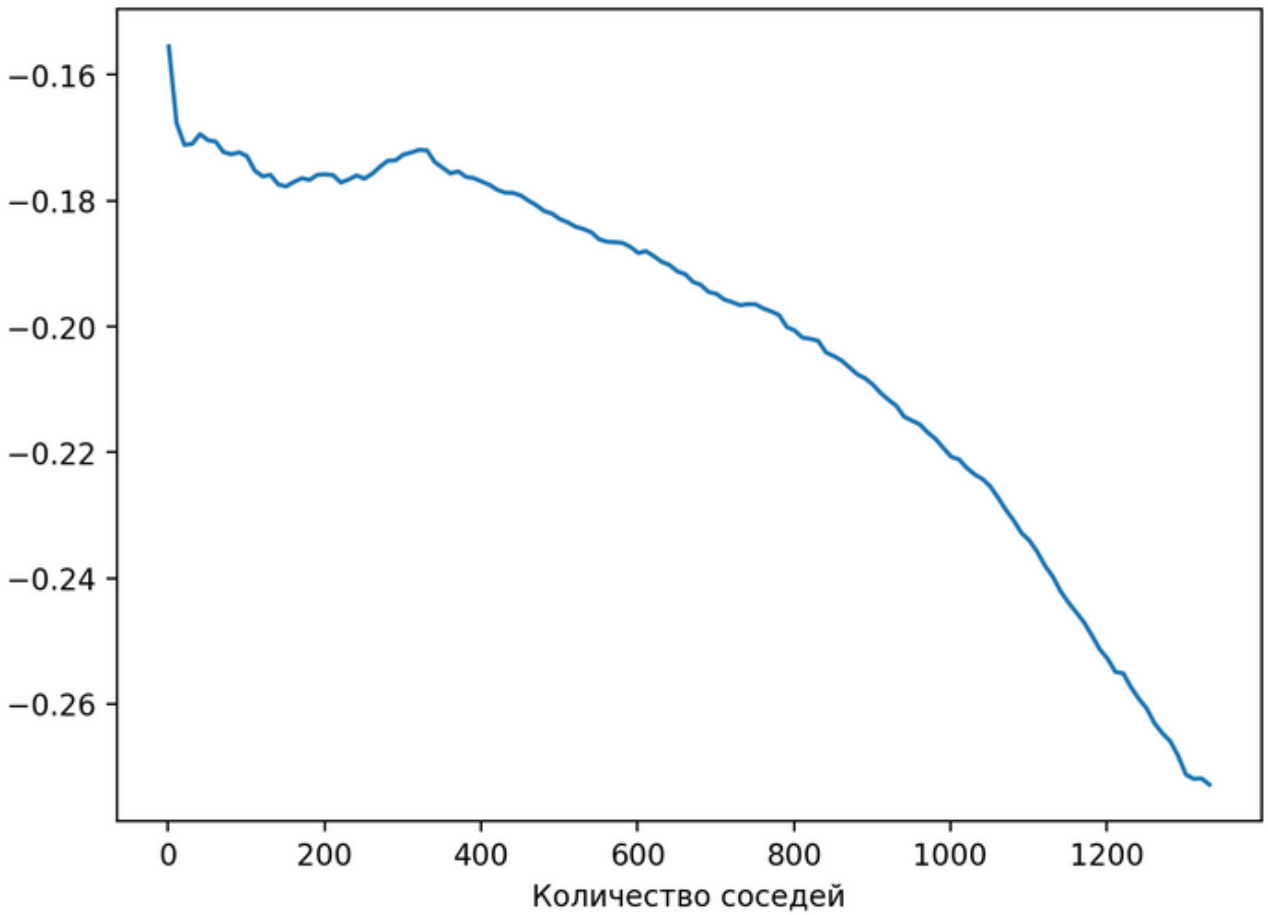
После установки флажка «Выполнять обучение» было выполнено обучение модели, которое длилось 21 секунду, после чего было получено полноэкранное сообщение. (Если на ОС не активна роль RDS, то будем маленькое окно с тем же текстом)

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

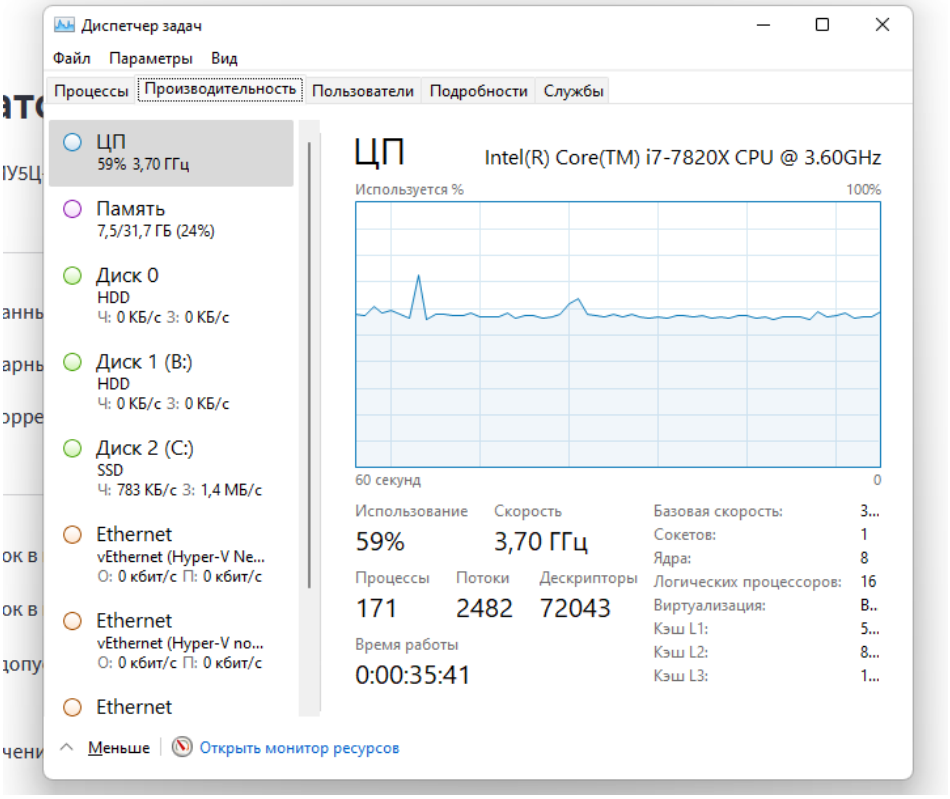
Возможные значения соседей - [1 11 21 31 41 51 61 71 81 91 101 111 121 131 141 151 161 171 181 191 201 211 221 231 241 251 261 271 281 291 301 311 321 331 341 351 361 371 381 391 401 411 421 431 441 451 461 471 481 491 501 511 521 531 541 551 561 571 581 591 601 611 621 631 641 651 661 671 681 691 701 711 721 731 741 751 761 771 781 791 801 811 821 831 841 851 861 871 881 891 901 911 921 931 941 951 961 971 981 991 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.

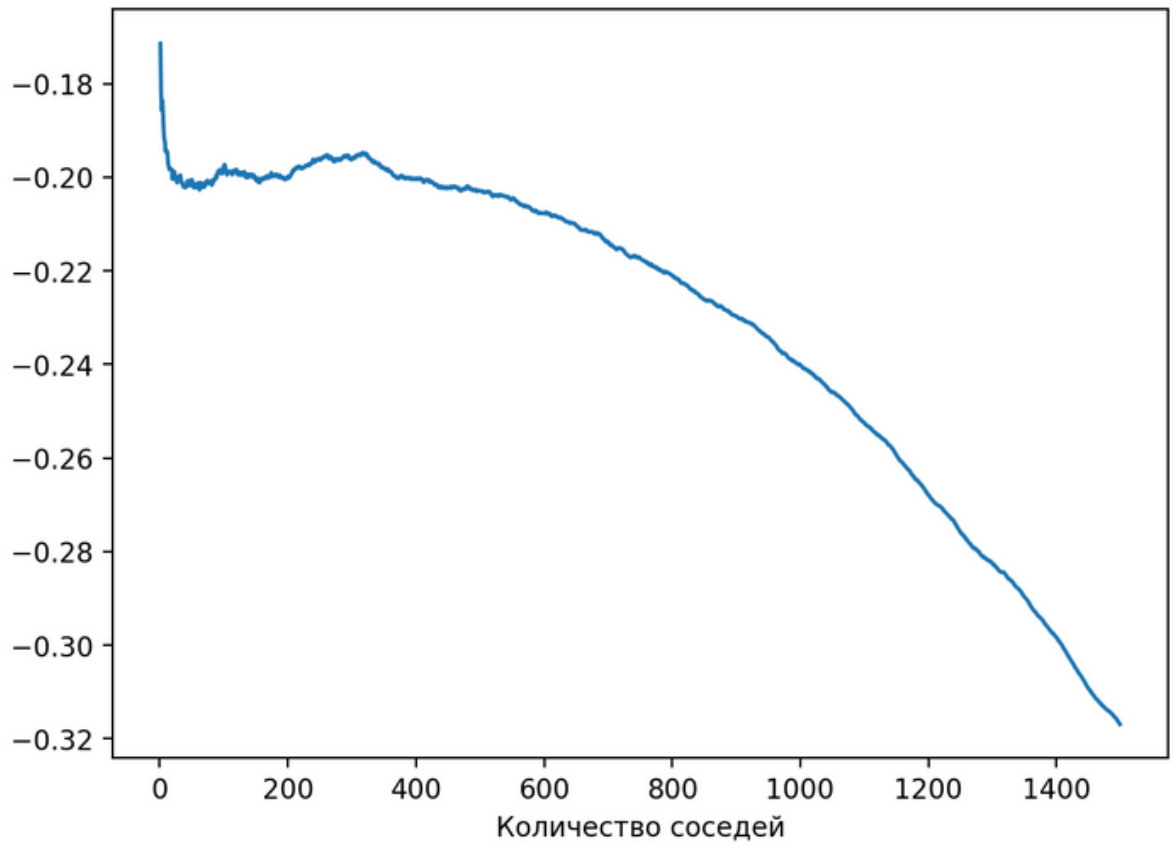


Ниже представлен получившийся график

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

Оценка качества модели

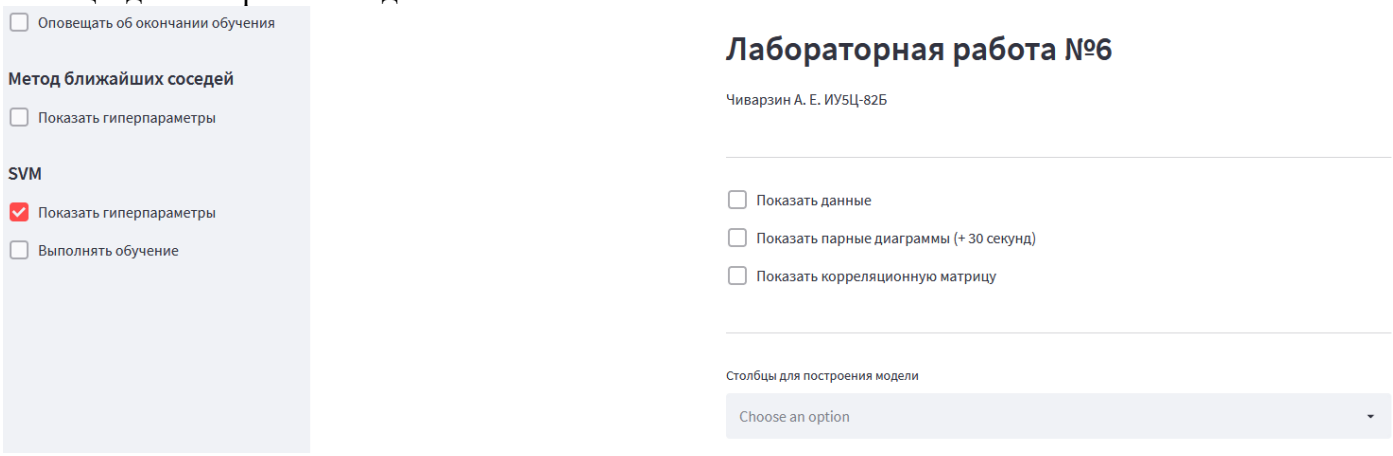
Лучшее значение параметров - {'n_neighbors': 1}



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 предложет выбрать столбцы для построения модели:



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

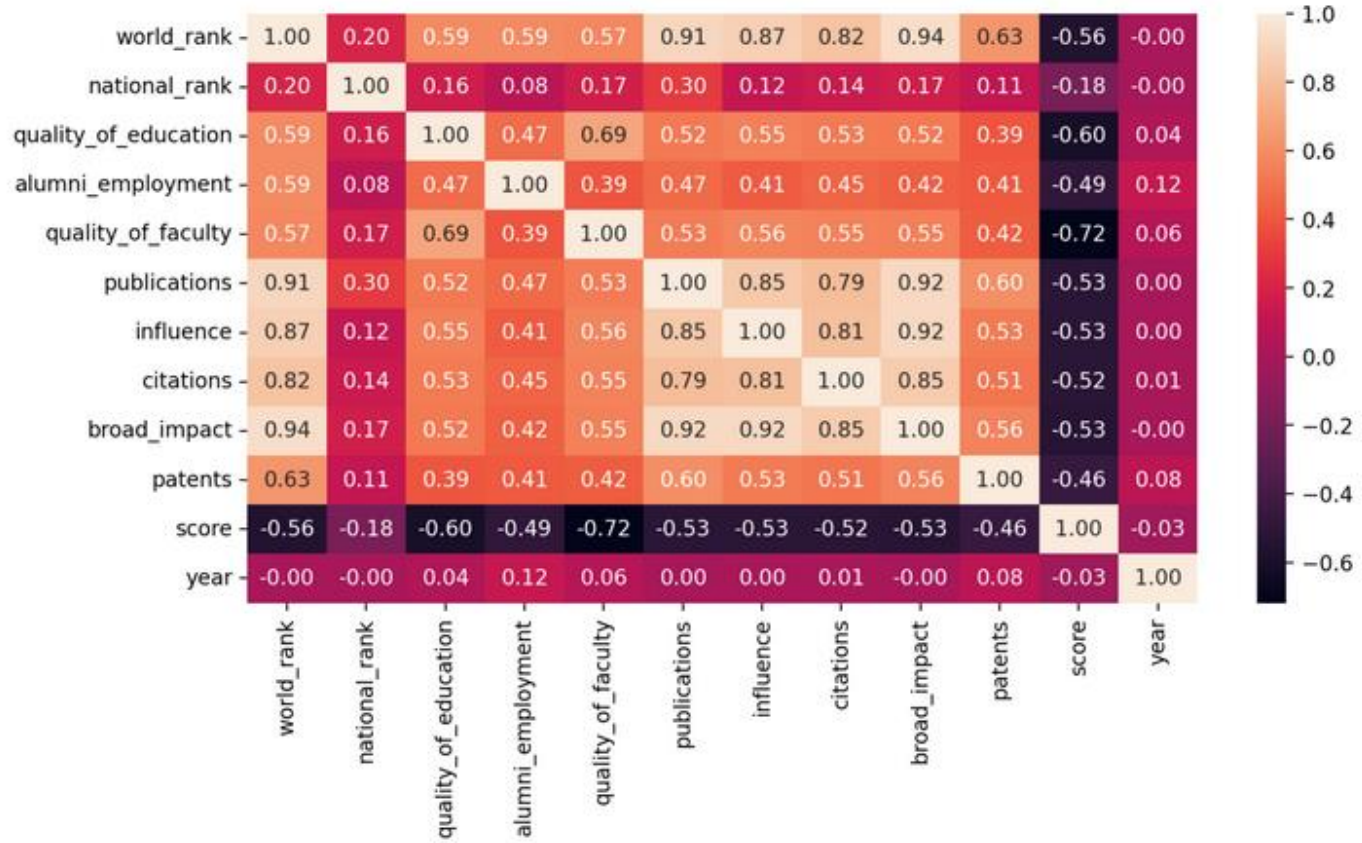
Лабораторная работа №6

Чиварзин А. Е. ИУ5Ц-82Б

☐ Показать данные

☐ Показать парные диаграммы (+ 30 секунд)

☑ Показать корреляционную матрицу

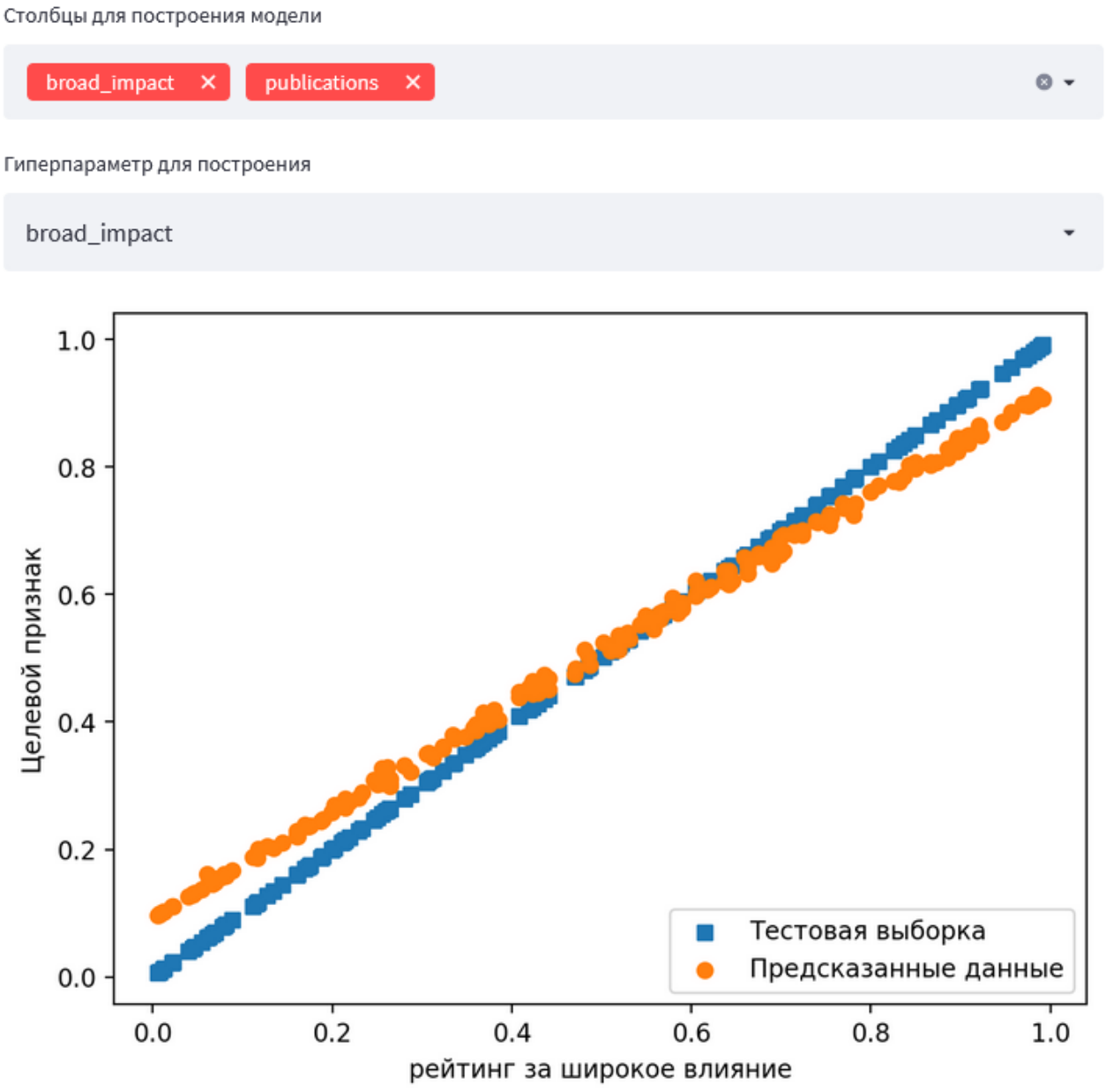


Столбцы для построения модели

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.

После установки параметров отметим флажок «Выполнять обучение».



Имеется возможность изменить гиперпараметр построения

Столбцы для построения модели

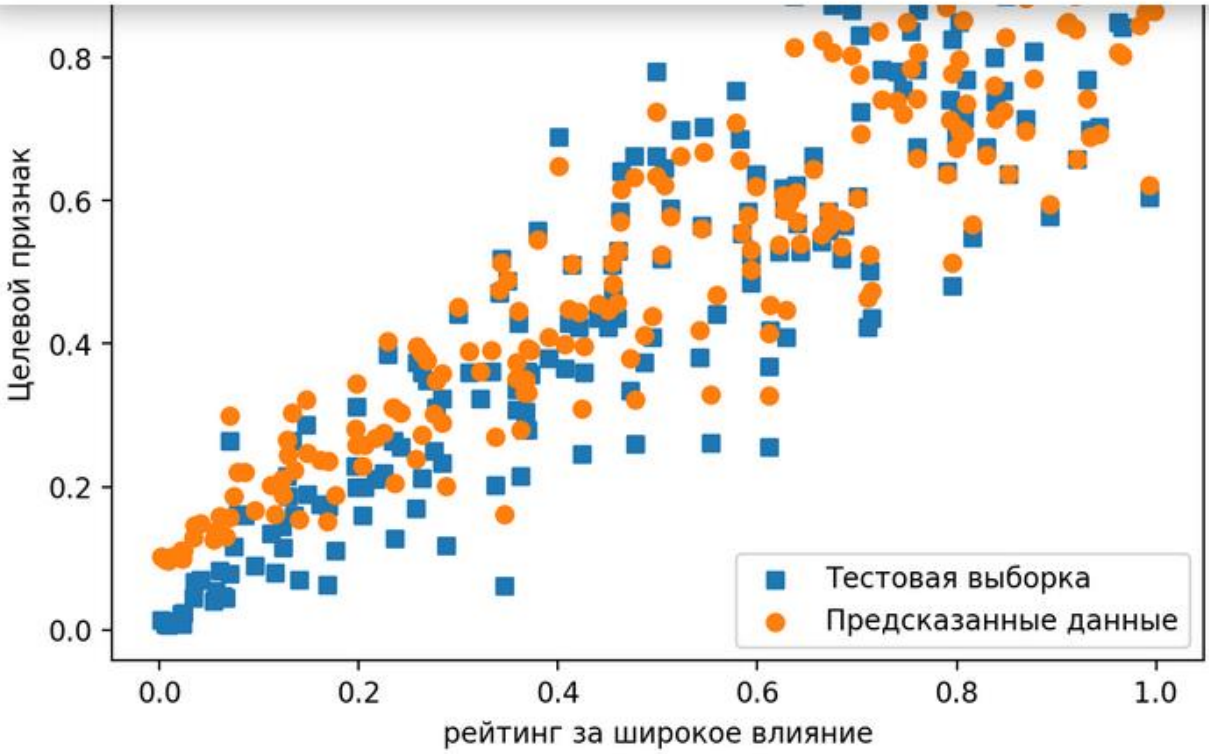
broad_impact ✕ publications ✕

Гиперпараметр для построения

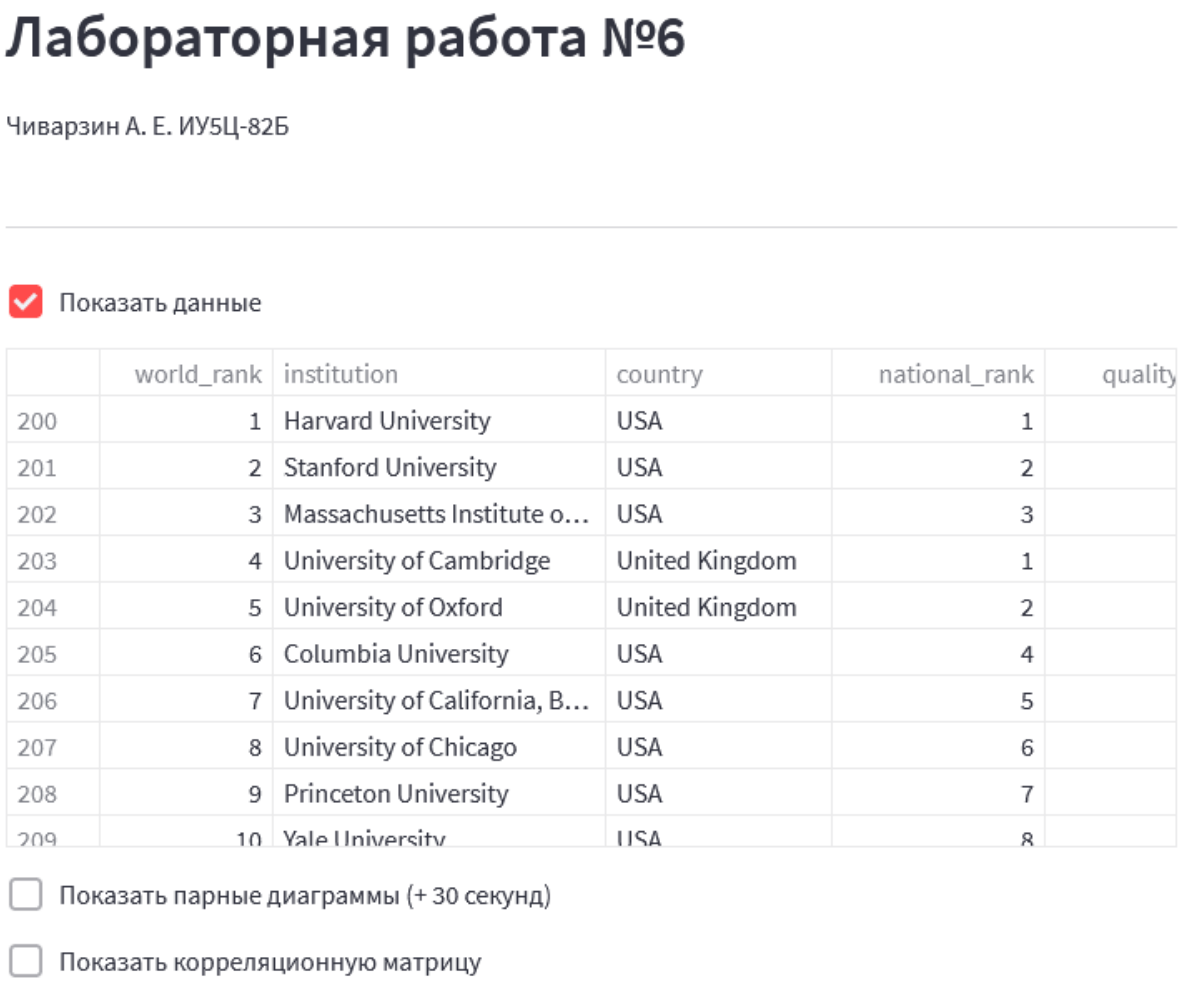
publications

broad_impact

publications

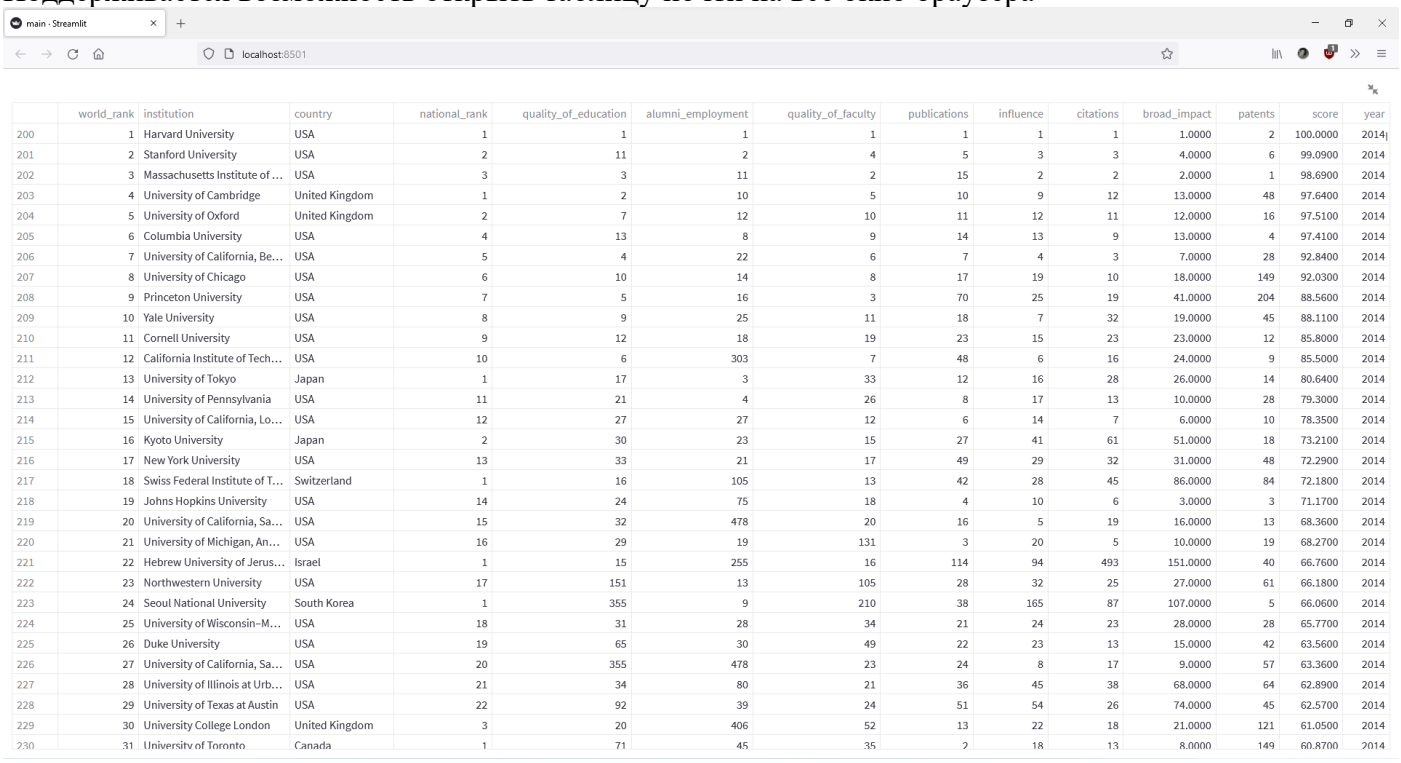


Вывод исходных данных
Были сняты все флажки и установлен флажок «Показать данные»



	world_rank	institution	country	national_rank	quality
200	1	Harvard University	USA	1	
201	2	Stanford University	USA	2	
202	3	Massachusetts Institute o…	USA	3	
203	4	University of Cambridge	United Kingdom	1	
204	5	University of Oxford	United Kingdom	2	
205	6	Columbia University	USA	4	
206	7	University of California, B…	USA	5	
207	8	University of Chicago	USA	6	
208	9	Princeton University	USA	7	
209	10	Yale University	USA	8	

Поддерживается возможность открыть таблицу почти на всё окно браузера



	world_rank	institution	country	national_rank	quality_of_education	alumni_employment	quality_of_faculty	publications	influence	citations	broad_impact	patents	score	year
200	1	Harvard University	USA	1	1	1	1	1	1	1	1.0000	2	100.0000	2014
201	2	Stanford University	USA	2	11	2	4	5	3	3	4.0000	6	99.0900	2014
202	3	Massachusetts Institute of …	USA	3	3	11	2	15	2	2	2.0000	1	98.6900	2014
203	4	University of Cambridge	United Kingdom	1	2	10	5	10	9	12	13.0000	48	97.6400	2014
204	5	University of Oxford	United Kingdom	2	7	12	10	11	12	11	12.0000	16	97.5100	2014
205	6	Columbia University	USA	4	13	8	9	14	13	9	13.0000	4	97.4100	2014
206	7	University of California, Be…	USA	5	4	22	6	7	4	3	7.0000	28	92.8400	2014
207	8	University of Chicago	USA	6	10	14	8	17	19	10	18.0000	149	92.0300	2014
208	9	Princeton University	USA	7	5	16	3	70	25	19	41.0000	204	88.5600	2014
209	10	Yale University	USA	8	9	25	11	18	7	32	19.0000	45	88.1100	2014
210	11	Cornell University	USA	9	12	18	19	23	15	23	23.0000	12	85.8000	2014
211	12	California Institute of Tech…	USA	10	6	303	7	48	6	16	24.0000	9	85.5000	2014
212	13	University of Tokyo	Japan	1	17	3	33	12	16	28	26.0000	14	80.6400	2014
213	14	University of Pennsylvania	USA	11	21	4	26	8	17	13	10.0000	28	79.3000	2014
214	15	University of California, Lo…	USA	12	27	27	12	6	14	7	6.0000	10	78.3500	2014
215	16	Kyoto University	Japan	2	30	23	15	27	41	61	51.0000	18	73.2100	2014
216	17	New York University	USA	13	33	21	17	49	29	32	31.0000	48	72.2900	2014
217	18	Swiss Federal Institute of T…	Switzerland	1	16	105	13	42	28	45	86.0000	84	72.1800	2014
218	19	Johns Hopkins University	USA	14	24	75	18	4	10	6	3.0000	3	71.1700	2014
219	20	University of California, Sa…	USA	15	32	478	20	16	5	19	16.0000	13	68.3600	2014
220	21	University of Michigan, An…	USA	16	29	19	131	3	20	5	10.0000	19	68.2700	2014
221	22	Hebrew University of Jerus…	Israel	1	15	255	16	114	94	493	151.0000	40	66.7600	2014
222	23	Northwestern University	USA	17	151	13	105	28	32	25	27.0000	61	66.1800	2014
223	24	Seoul National University	South Korea	1	355	9	210	38	165	87	107.0000	5	66.0600	2014
224	25	University of Wisconsin–M…	USA	18	31	28	34	21	24	23	28.0000	28	65.7700	2014
225	26	Duke University	USA	19	65	30	49	22	23	13	15.0000	42	63.5600	2014
226	27	University of California, Sa…	USA	20	355	478	23	24	8	17	9.0000	57	63.3600	2014
227	28	University of Illinois at Urb…	USA	21	34	80	21	36	45	38	68.0000	64	62.8900	2014
228	29	University of Texas at Austin	USA	22	92	39	24	51	54	26	74.0000	45	62.5700	2014
229	30	University College London	United Kingdom	3	20	406	52	13	22	18	21.0000	121	61.0500	2014
230	31	University of Toronto	Canada	1	71	45	35	2	18	13	8.0000	149	60.8700	2014

.

Вывод парных диаграмм
Были сняты все флажки и установлен флажок «Показать парные лмаграммы»