

Разведочный анализ данных об успеваемости обучающихся по курсу математики в средней школе в Google Colaboratory

Голубева Евгения Павловна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Цель данной статьи – выполнить разведочный анализ данных об успеваемости учащихся по курсу математики в средней школе. Для разведочного анализа была использована интерактивная облачная среда Google Colaboratory и данные об успеваемости обучающихся по курсу математики в средней школе. С использованием Google Colaboratory была произведена предобработка данных, анализ статистических показателей, построение графиков для визуализации результатов, а также обучена и протестирована модель.

Ключевые слова: Google Colaboratory, библиотека, визуализация данных, разведочный анализ.

Exploratory analysis of data on the academic performance of students in a high school mathematics course in Google Coolaboratory

Golubeva Evgeniya Pavlovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to perform an exploratory analysis of data on student academic performance in a high school mathematics course. For the exploratory analysis, the interactive cloud environment Google Coolaboratory and data on the academic performance of students in the course of mathematics in high school were used. Using Google Coolaboratory, data was preprocessed, statistical indicators were analyzed, graphs were built to visualize the results, and the model was trained and tested.

Keywords: Google Coolaboratory, library, data visualization, exploratory analysis.

1 Введение

1.1 Актуальность

Разведочный анализ данных является важным этапом в исследовательском процессе. Он помогает обнаружить особенности и закономерности в данных, выделить ключевые факторы и сформулировать гипотезы для дальнейших исследований. В данной статье будет представлен подробный анализ статистических показателей и построены графические

визуализации результатов, что делает данное исследование полезным для научного сообщества.

Использование Google Colaboratory в качестве инструмента для проведения разведочного анализа данных позволяет упростить и автоматизировать процесс обработки и анализа больших объемов информации. Это является актуальным и интересным направлением исследований в области образования.

1.2 Обзор исследований

Т.С. Волокитина в статье описала возможности Google Colab для изучения технологий машинного обучения и нейронных сетей. [1]. В статье рассматривали разведочный анализ данных, описывали инструменты реализации анализа, библиотеки Python, и представили пример, выполненный на данных обнаружения присутствия людей в помещении Е.А. Григорьев, Н.С. Климов [2]. О.В. Кудринская и Е.А. Лутцева в статье показали возможности языка Python и популярные библиотеки, позволяющие визуализировать данные [3]. Продемонстрировали использование системы Google Colaboratory в обучении курса «Методы интеллектуального анализа данных» Д.И. Зенюкова и С.Н. Байбекова [4]. О.В. Кудринская и Е.А. Лутцева в статье рассматривали необходимость изучения сервисов в образовательном процессе в рамках программирования на языке Python, а также основные возможности и достоинства двух интерактивных сред (Jupyter Notebook, Google Colaboratory). [5].

1.3 Цель исследования

Цель исследования - выполнить разведочный анализ данных об успеваемости учащихся по курсу математики в средней школе с помощью Google Colaboratory.

2 Материалы и методы

Для разведочного анализа используется интерактивная облачная среда Google Colaboratory. Работа будет происходить на готовых данных об успеваемости обучающихся по курсу математики в средней школе, скачать которые можно по ссылке:

<https://www.kaggle.com/datasets/devansodariya/student-performance-data>

3 Результаты и обсуждения

Открываем Google Colaboratory, и создаем новый блокнот (см.рис.1).

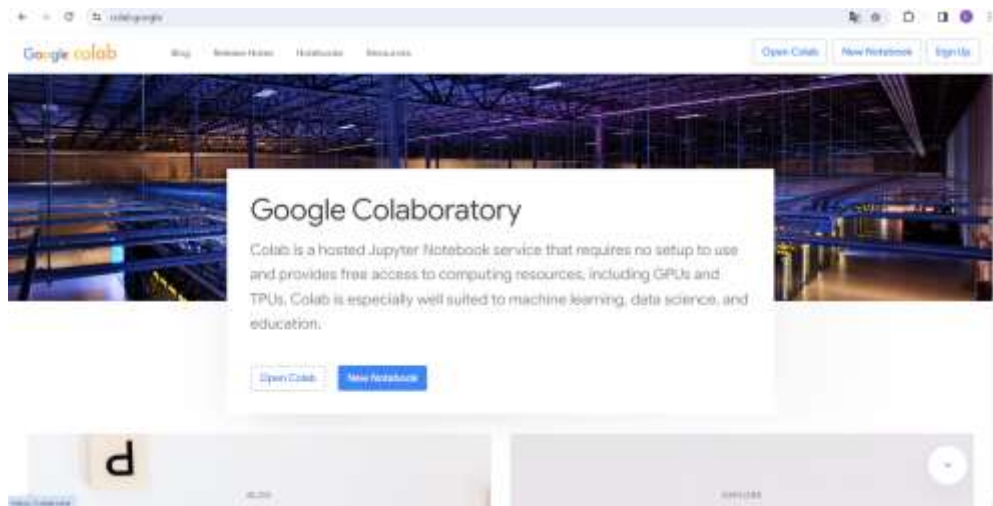


Рисунок 1- Создание блокнота

Подключаем необходимые библиотеки для разведочного анализа (см.рис.2).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import warnings
warnings.filterwarnings("ignore")
```

Рисунок 2- Подключение библиотеки

Добавляем файл student_data.csv. Для этого необходимо открыть вкладку «файл», далее нажимаем иконку «Загрузить в сессионное хранилище», и выбираем файл «student_data.csv» (см.рис.3).

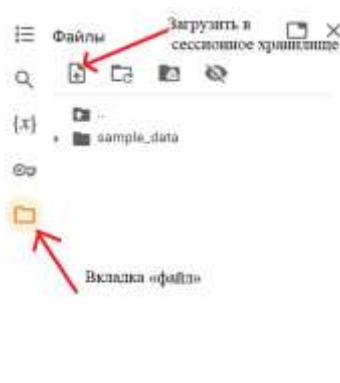


Рисунок 3- Добавление файла «student_data.csv»

Далее считываем содержимое файла, и проверяем, что данные прочитаны (см.рис.4).

```
[ ] data = pd.read_csv("student_data.csv")
data.head()
```

	school	sex	age	address	failsize	Patatus	Medu	Fedu	Hjob	Fjob	...	faexel	freetime	gnot	Dalc	Male	health	absences	G1	G2	G3
0	GP	F	16	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	6	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	6	4	6	10	10

5 rows * 23 columns

Рисунок 4- Чтение данных csv

С помощью метода `data.describe()` получим описательную статистику для числовых значений (см.рис.5).

```
[4] data.describe()
```

	age	Medu	Fedu	traveltime	studytime	#failres	faexel	freetime	gnot	Dalc	Male	health	absences	G1
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.000203	2.740307	2.521119	1.448301	2.035443	0.334177	3.944304	2.235443	3.103201	1.401013	2.291139	3.554430	5.702863	10.5028
std	1.276240	1.554738	1.000201	0.607508	0.839240	0.783161	0.895401	0.993502	1.113270	0.693741	1.307607	1.290501	0.803096	3.3191
min	10.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	3.0000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	3.000000	1.000000	1.000000	0.000000	0.000000	6.0000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.000000	4.000000	11.0000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	6.000000	4.000000	4.000000	2.000000	3.000000	6.000000	6.000000	13.0000
max	22.000000	4.000000	4.000000	4.000000	3.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	75.000000	19.0000

Рисунок 5- Описательную статистику для числовых значений

Далее используем метод `data.info()`. Данный метод соотносит максимальное количество записей в датафрейме с количеством записей в каждом столбце.

После запуска кода появился результат, в котором показано, что файл «student_data.csv» содержит 395 записей и не имеет пропуски в столбцах (см.рис.6).

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
#   Column          Non-Null Count  Dtype
---  ---            -
0   school          395 non-null    object
1   sex             395 non-null    object
2   age            395 non-null    int64
3   address        395 non-null    object
4   famsize        395 non-null    object
5   Pstatus        395 non-null    object
6   Medu           395 non-null    int64
7   Fedu           395 non-null    int64
8   Mjob           395 non-null    object
9   Fjob           395 non-null    object
10  reason         395 non-null    object
11  guardian       395 non-null    object
12  traveltime     395 non-null    int64
13  studytime      395 non-null    int64
14  failures       395 non-null    int64
15  schoolsup       395 non-null    object
16  famsup         395 non-null    object
17  paid           395 non-null    object
18  activities     395 non-null    object
19  nursery        395 non-null    object
20  higher         395 non-null    object
21  internet       395 non-null    object
22  romantic       395 non-null    object
23  famrel         395 non-null    int64
24  freetime       395 non-null    int64
25  goout          395 non-null    int64
26  Dalc           395 non-null    int64
```

Рисунок 6- Результат метода data.info()

С помощью `categorical_columns` мы получаем значение, который предоставляет информацию о столбцах. Чтобы вывести информацию, необходимо прописать `len(categorical_columns)` (см.рис.7).

```
{x} [ ] categorical_columns = data.select_dtypes(include=['object']).columns
length = len(categorical_columns)
length

17
```

Рисунок 7- Информация о столбцах

С помощью класса `plt.figure` визуализируем данные `categorical_columns` (см.рис.8)

```
plt.figure(figsize=(20, 15))
for i, col in enumerate(categorical_columns, start=1):
    plt.subplot(5, 4, i)
    data[col].value_counts().plot.pie(autopct='%1.1f%%', startangle=90)
    plt.title(f'{col} Pie Chart of Categorical Variable')

plt.tight_layout()
plt.show()
```

Рисунок 8- Код для визуализации данных `categorical_columns`

После выполнения кода, появились диаграммы, на которых можем увидеть категории столбцов и их значение (см.рис.9).

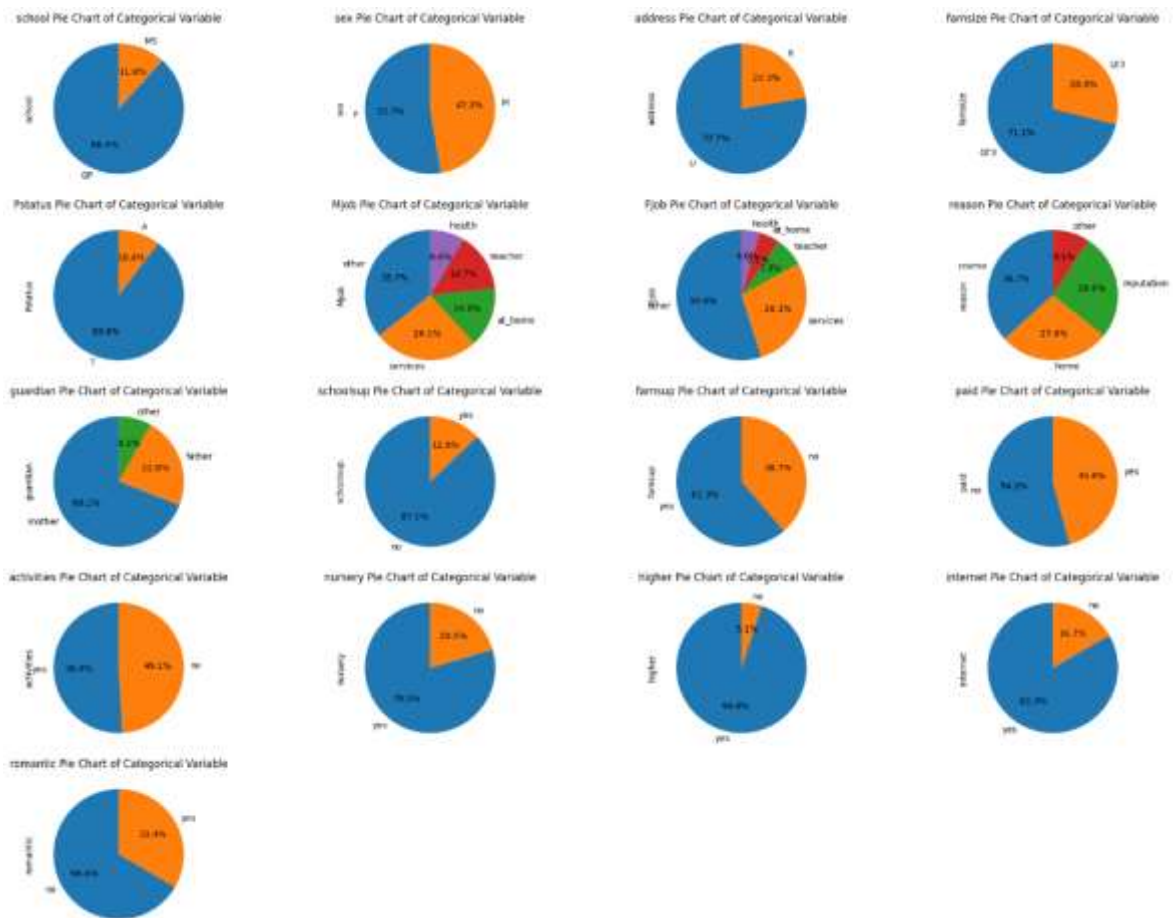


Рисунок 9- Диаграмма столбцов

Для разведочного анализа данных используем виды графиков, такие как: гистограмма, scatterplot, boxplot и barplot (см.рис.10).

```

def diagnostic_plots(df, variable, target):
    # histogram
    plt.figure(figsize=(10, 7))
    plt.subplot(1, 4, 1)
    sns.histplot(df[variable], kde=True, color='r')
    plt.title(f'{variable} Histogram')

    # scatterplot
    plt.subplot(1, 4, 2)
    plt.scatter(df[variable], df[target], color='g')
    plt.title(f'{variable} vs {target} Scatterplot')

    # boxplot
    plt.subplot(1, 4, 3)
    sns.boxplot(y=df[variable], color='b')
    plt.title(f'{variable} Boxplot')

    # Barplot
    plt.subplot(1, 4, 4)
    sns.barplot(data=df, x=target, y=variable)
    plt.title(f'{variable} vs {target} Barplot ')

    plt.show()

for col in data.select_dtypes(include=['int64']).columns[:-1]:
    print(col)
    diagnostic_plots(data, col, 'G3')
    
```

Рисунок 10- Код визуализации данных

С помощью данного кода визуализировали характеристики учащихся и итоговую оценку(G3). Для примера возьмем графики оценка за первый период(G1) и оценка за второй период (G2). На графиках видно, что за первый период у учеников успеваемость лучше, чем за второй период (см.рис.11).

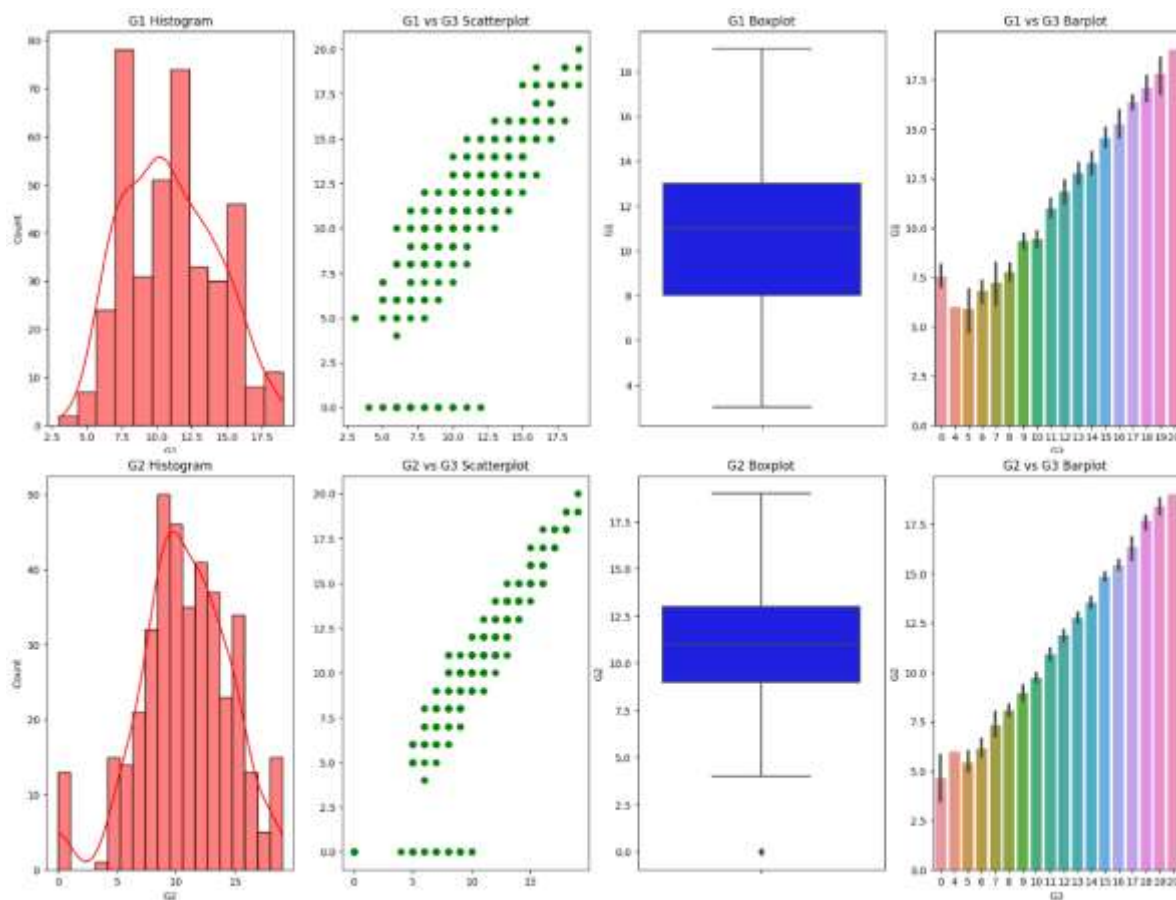


Рисунок 11- Графики визуализировали характеристики учащихся и итоговой оценки(G3)

Далее проведем анализ взаимосвязей между переменными. Для этого создадим список числовых индексов, который извлекает данные из списка столбцов (см.рис.12).

```
for features_list in data.columns:
    features = data[features_list].unique()
    print("#"*25)
    print(f"{features_list.title()} Variables : {features} \nPiece:{len(features)}")

#####
School Variables : ['GP' 'MS']
Piece:2
#####
Sex Variables : ['F' 'M']
Piece:2
#####
Age Variables : [18 17 15 16 19 22 20 21]
Piece:8
#####
Address Variables : ['U' 'R']
Piece:2
#####
Famsize Variables : ['GT3' 'LE3']
Piece:2
#####
Pstatus Variables : ['A' 'T']
Piece:2
#####
Medu Variables : [4 1 3 2 0]
Piece:5
#####
Fedu Variables : [4 1 2 3 0]
Piece:5
#####
Mink Variables : ['at home' 'health' 'other' 'services' 'teacher']
```

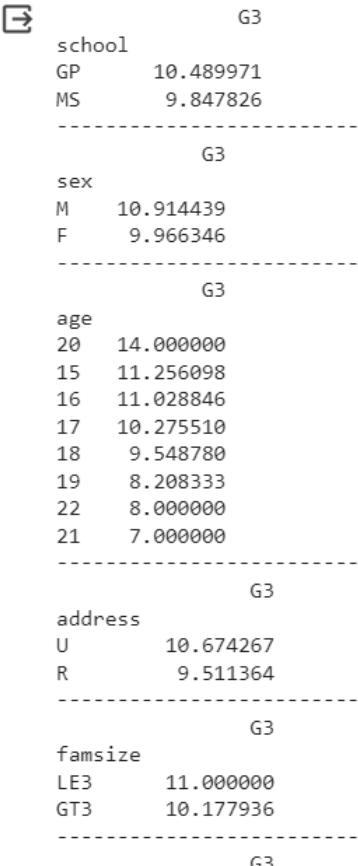
Рисунок 12- Данные из списка столбцов

С помощью данного кода преобразуем список столбцов (см.рис.13).


```

▶ for i in data.columns[:-1]:
    print(data[[i, "G3"].groupby([i]).mean().sort_values(by="G3")[:-1])
    print("-"*25)

```



```

          G3
school
GP      10.489971
MS       9.847826
-----

          G3
sex
M       10.914439
F       9.966346
-----

          G3
age
20      14.000000
15      11.256098
16      11.028846
17      10.275510
18       9.548780
19       8.208333
22       8.000000
21       7.000000
-----

          G3
address
U       10.674267
R       9.511364
-----

          G3
famsize
LE3     11.000000
GT3     10.177936
-----

          G3

```

Рисунок 13- Преобразование списка

Далее построим корреляционную матрицу для оценки взаимосвязей между признаками (см.рис.14).

```

▶ list_value = ["G1", "G2", "G3"]
sns.heatmap(data[list_value].corr(), annot=True, fmt=".2f")
plt.show()

```

Рисунок 14- Код для построение корреляционной матрицы

По корреляционной матрицы можно определить, что элемент оценки за второй период (G2) и итоговая оценка(G3) имеет значение 0.90, и это значит, что значение близкое к 1 указывает на положительную корреляцию, т.е переменные движутся в одном направлении (см.рис.15).

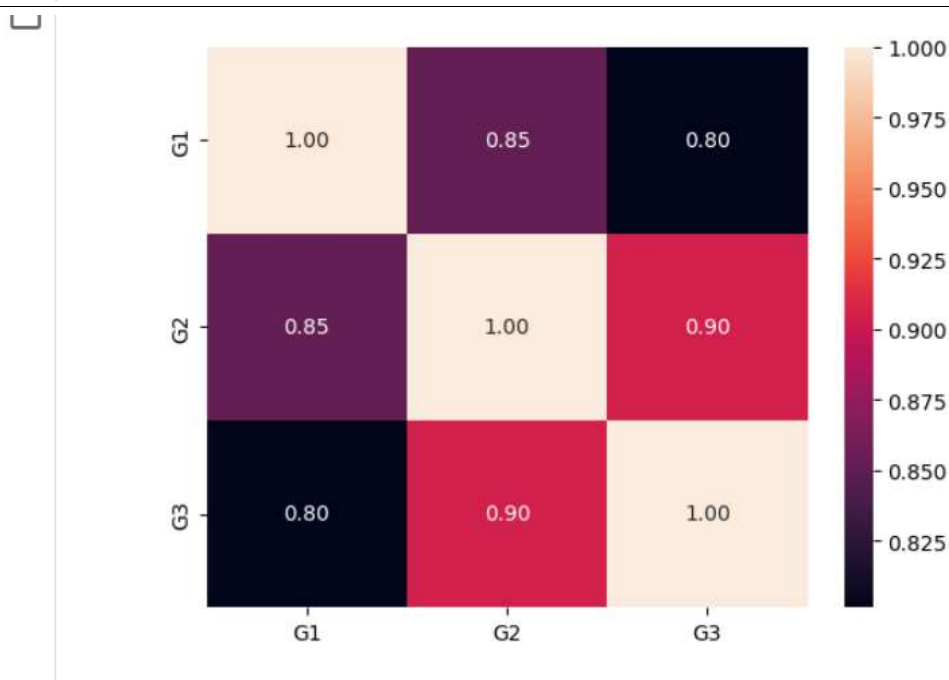


Рисунок 15- Корреляционная матрица

4 Выводы

В данной работе был выполнен разведочный анализ данных об успеваемости учащихся по курсу математики в средней школе с помощью интерактивной облачной среды Google Colaboratory. Данное исследование может быть использовано методическим пособием в учебной деятельности.

Библиографический список

1. Григорьев Е. А., Климов Н. С. Разведочный анализ данных с помощью python //E-Scio. 2020. №. 2 (41). С. 165-176.
2. Волокитина Т. С. Анализ возможностей Google Colab //Современные научные исследования и инновации. 2020. №. 12. С. 1-1.
3. Кудринская О. В. Визуализация данных с использованием возможностей языка python //Теория и практика современных гуманитарных и естественных наук. 2021. С. 176-180.
4. Насруева М.А.Визуализация аналитических данных с использованием языка программирования python // Сборник научных статей по материалам V Международной научно-практической конференции. Уфа, 2021. С. 25-36.
5. Федотова М. С., Барышева Н. Н. Обзор информационных систем на языке python в государственном управлении //Наука и образование. 2022. Т. 5. №.2. С. 374.