|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_ Оперативный анализ данных \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по теме\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_«16000+ Movies 1910-2024 (Metacritic)» \_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент ИУ5-55Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Ходырев Р.В.

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  МаслениковК.Ю.

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Маслеников К.Ю.

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

по теме \_\_Анализ данных по базе данных «Adidas US Sales» \_\_

Студент группы \_\_\_\_\_\_\_ИУ5-55Б\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ходырев Роман Владиславович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Исследовательская\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики(кафедра,предприятие,НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_НИР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения НИР: 25% к 3 нед., 50% к 9 нед., 75% к 12 нед., 100% к 15 нед.

***Техническое задание***  Спроектировать систему анализа и визуализации по теме «Adidas US Sales», провести визуализацию полученных данных посредством языка программирования Python

***Оформление научно-исследовательской работы:***

Расчетно-пояснительная записка на 22 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «14» сентября 2024 г.

**Руководитель НИР**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Маслеников К.Ю.

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Ходырев Р.В.

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Аннотация**

По заданию требуется проанализировать базу данных, очистить её от

ненужных данных, также нужно выдвинуть гипотезы, проанализировать их и составить графики по данным гипотезам. Для данного задания была выбрана база данных «16000+ Movies 1910-2024 (Metacritic)». В ходе работы будут использоваться такие библиотеки как: «Pandas» и «Seaborn». Анализ будет проводиться через программу «Visual Studio Code».

Также будет производиться корреляционный анализ, агрегирование данных, оптимизация памяти, удаление дубликатов, очистка данных. Будет осуществлён анализ гипотез по известным данным и построены соответствующие графики и схемы.

В работе будут рассмотрены данные о различных фильма, снятых с 1910 по 2024г. В конце, будет сделан вывод о проделанной работе с данным датасетом.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc182044356)

1. [Определение данных для анализа 6](#_Toc182044357)

[2. Формулирование гипотез 6](#_Toc182044359)

[3. Подготовка данных для работы 8](#_Toc182044360)

[4. Исследовательский анализ данных 11](#_Toc182044362)

[Гипотеза 1 11](#_Toc182044363)

[Гипотеза 2 13](#_Toc182044364)

[Гипотеза 3 15](#_Toc182044365)

[Гипотеза 4 17](#_Toc182044366)

[Анализ корреляции 19](#_Toc182044368)

[ЗАКЛЮЧЕНИЕ 21](#_Toc182044369)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc182044370)

**ВВЕДЕНИЕ**

**Цель работы:**

Исследовать базу данных и изучить данные о фильмах, снятых в 1910-2024 годах, выявить закономерность между жанром фильма и его оценкой, чтобы дать представление киностудиям о том, фильмы какого жанра нравятся зрителям больше всего. Это позволит студиям снимать фильмы, которые принесут им большую прибыль.

**Задачи:**

1. Определение данных.
2. Формулирование гипотез.
3. Загрузка данных в Python.
4. Проверка данных.
5. Очистка данных.
6. Преобразование данных.
7. Выбор данных для анализа.
8. Агрегирование данных.
9. Визуализация данных.
10. Подтверждение или опровержение поставленных гипотез.
11. Корреляционный анализ.
12. Формулирование ограничений и выводов.
13. **Определение данных для анализа**

Для анализа был выбран DataSet «Movies 1910-2024 (Metacritic)». Набор данных содержит информацию об фильмах, снятых в промежутке между 1910 и 2024 годом.

Таблица имеет 16000 записей. Каждая запись содержит информацию о названии фильма; дате выхода; краткое описание; рейтинг на портале Metacritic; количестве людей, поставивших оценку; режиссёре; сценаристах; продолжительности фильма и его жанре.

1. **Формулирование гипотез**
2. Фильмы в жанре драма имеют самый высокий средний рейтинг, относительно других жанров.
3. Средняя продолжительность большинства фильмов находится между 1 ч. 30 мин. и 1 ч. 40 мин.
4. 1994 год – лучший год в истории кинематографа.
5. Количество выпускаемых фильмов растет с каждым годом.

**Обоснование сформулированных гипотез**

1. Было выдвинуто предположение о том, что фильмы в жанре драма имеют наиболее высокий средний рейтинг по сравнению с фильмами других жанров, так как драматические фильмы находят отклик в глазах зрителей, что повышает их мнение о произведении.
2. Выдвинутое предположение о средней продолжительности фильма, лежащего в интервале между 1 ч. 30 мин. и 1 ч. 40 мин., основывается на том, что фильмы меньшей продолжительности не привлекают зрителей, желающих полностью погрузиться в историю; а фильмы большей продолжительности успевают зрителям надоесть, порождая негативное мнение о произведении.
3. 1994 год – лучший год в истории кинематографа. Данная гипотеза основана на том, что наиболее известные, культовые фильмы, почитаемые большинством кинолюбителей, такие как «Побег из Шоушенка», «Форест Гамп», «Криминальное чтиво» и др. были сняты именно в этот год.
4. Гипотеза о том, что количество выпускаемых фильмов растет с каждым годом, основана на том, что покупательная способность людей в развитых странах ежегодно увеличивается, а просмотр кинофильмов стал более доступным (вследствие развития технологий, роста количества кинотеатров и пр.), что в свою очередь стимулирует крупные киностудии к съемкам всё большего числа фильмов.
5. **Подготовка данных для работы**

Для анализа данных было использовано программное обеспечение «Pandas» совместно с «matplotlib».

Изображение выглядит как Шрифт, текст, Графика, снимок экрана

Автоматически созданное описание

Рисунок 1 – Импортирование зависимостей

**3.1 Загрузка данных в Python**

Для загрузки /csv файла с данными в pandas была использована функция read\_csv().

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 2 – Загрузка .csv файла

**3.2 Проверка данных**

Набор данных был проверен на наличие данных по столбцам. Для этого был использован метод info() для объектов dataframe.

Листинг:

def main():

    df = openFile()

    df.info()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

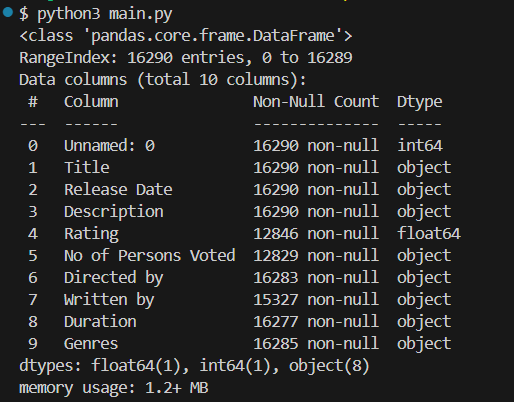


Рисунок 3 – Результат df.info()

**3.3 Очистка данных**

**Работа с дубликатами**

При помощи метода drop\_dubliсates() была совершена попытка удалить записи с полностью совпадающим набором по столбцам: название, дата выпуска, описание, рейтинг, количество оценок, режиссёр, сценаристы, продолжительность, жанр.

Листинг:

def get\_count\_of\_duplicates(df):

    count\_duplicates = len(df) - len(df.drop\_duplicates())

    print(f"Очищено записей: {count\_duplicates}")

def main():

    df = openFile()

    get\_count\_of\_duplicates(df)



Рисунок 4 – Отсутствие повторяющихся данных

**3.4. Устранение ошибок**

У некоторых строк в столбце Duration замечено использование некорректных данных – пустых значений или посторонних ссылок. Очистим данные от недопустимых значений с помощью функции df.loc()

Листинг:

import re

class Cleaner:

    def clean\_duration(self, df):

        pattern = r'^(\d+)( h (\d+)( m)?)?$'

        invalid\_count = 0

        valid\_rows = []

        for idx, value in enumerate(df['Duration']):

            if re.match(pattern, str(value)):

                valid\_rows.append(idx)

            else:

                invalid\_count += 1

        df\_cleaned = df.loc[valid\_rows]

        print(f"Удалено строк из-за неверного значения Duration: {invalid\_count}")

        return df\_cleaned

from clean import Cleaner

def main():

    df = openFile()

    cleaner = Cleaner()

    df = cleaner.clean\_duration(df)

    print(df)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 5 – Очистка столбца Duration

1. **Исследовательский анализ данных**

**Гипотеза 1 - Фильмы в жанре драма имеют самый высокий средний рейтинг, относительно других жанров.**

Для этого составим гистограмму, которая будет отображать среднее значение для фильмов каждого жанра

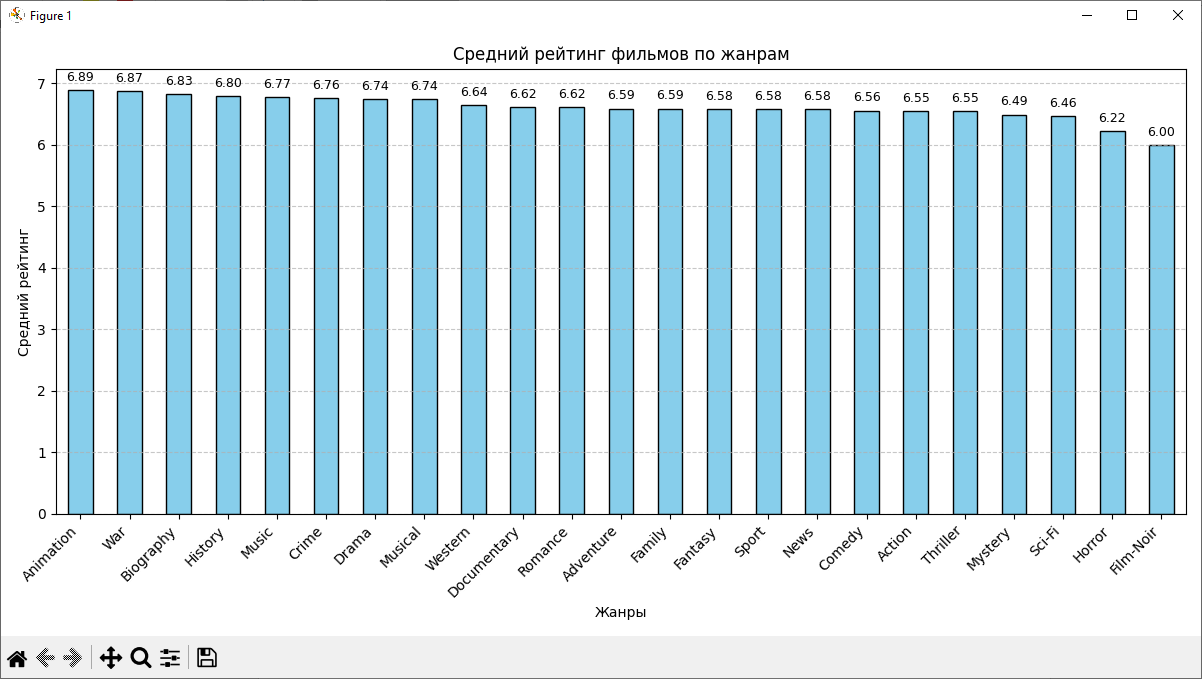


Рисунок 6 – гистограмма средних рейтингов по жанрам

Код для проверки гипотезы

# Функция для анализа среднего рейтинга по жанрам

def analyze\_genres(df):

    df['Genres'] = df['Genres'].str.split(',')

    df = df.explode('Genres')

    df = df[df['Genres'] != 'Unknown']

    genre\_ratings = df.groupby('Genres')['Rating'].mean().sort\_values(ascending=False)

    plt.figure(figsize=(12, 6))

    ax = genre\_ratings.plot(kind='bar', color='skyblue', edgecolor='black')

    for i, value in enumerate(genre\_ratings):

        ax.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=9)

    plt.title('Средний рейтинг фильмов по жанрам')

    plt.xlabel('Жанры')

    plt.ylabel('Средний рейтинг')

    plt.xticks(rotation=45, ha='right')

    plt.grid(axis='y', linestyle='--', alpha=0.7)

    plt.tight\_layout()

    plt.show()

    print("Средние рейтинги по жанрам:")

    print(genre\_ratings)

**Вывод:**

На графике видно, что наибольшее среднее значение по рейтингам имею фильмы в жанре Анимация (6.89), Военные (6.87) и Биографические (6.83). Фильмы в жанре Драма (6.74) занимают лишь 7 место. Гипотеза неверна.

**Гипотеза 2 - Средняя продолжительность большинства фильмов находится между 1 ч. 30 мин. и 1 ч. 40 мин.**

Построим гистограмму и проведем через ее середины диаграмму вероятностного распределения. Также, для очистки данных вычислим 2- и 98-перцентили, чтобы отбросить многосерийные фильмы, продолжительность которых в датасете суммируется, и рекламные ролики, продолжительность которых крайне мала.

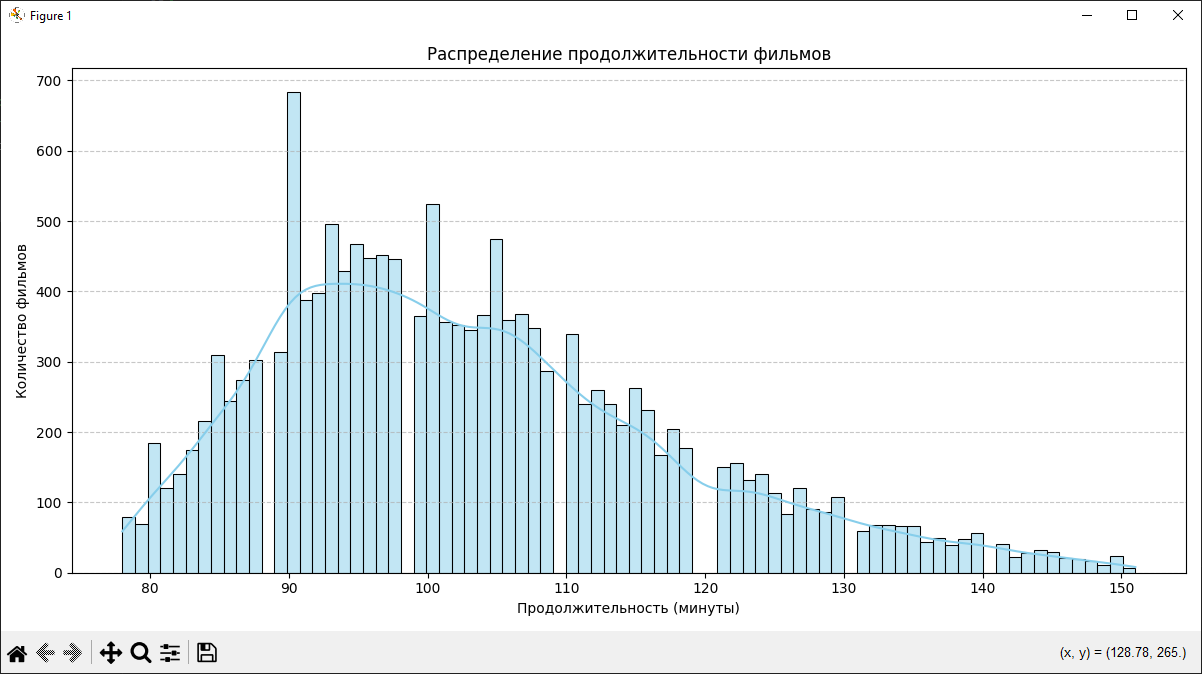


Рисунок 7 – нормальное распределение продолжительности фильма

Листинг:

import re

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from clean import Cleaner

# Функция для очистки и преобразования продолжительности

def clean\_duration(df):

    def duration\_to\_minutes(duration):

        match = re.match(r'(\d+)\s\*h\s\*(\d+)\s\*m', str(duration))

        if match:

            hours = int(match.group(1))

            minutes = int(match.group(2))

            return hours \* 60 + minutes

        match = re.match(r'(\d+)\s\*m', str(duration))

        if match:

            return int(match.group(1))

        return None

    df['Duration'] = df['Duration'].apply(duration\_to\_minutes)

    df = df.dropna(subset=['Duration'])

    return df

# Функция для анализа продолжительности фильмов

def analyze\_duration(df):

    df = df[df['Duration'] > 0]

    lower\_percentile = df['Duration'].quantile(0.02)

    upper\_percentile = df['Duration'].quantile(0.98)

    df = df[(df['Duration'] >= lower\_percentile) & (df['Duration'] <= upper\_percentile)]

    plt.figure(figsize=(12, 6))

    sns.histplot(df['Duration'], kde=True, color='skyblue', bins=80)

    plt.title('Распределение продолжительности фильмов')

    plt.xlabel('Продолжительность (минуты)')

    plt.ylabel('Количество фильмов')

    plt.grid(axis='y', linestyle='--', alpha=0.7)

    plt.tight\_layout()

    plt.show()

    print(f"Основные статистики по продолжительности:\n{df['Duration'].describe()}")

    print(f"Средняя продолжительность: {df['Duration'].mean():.2f} минут")

def main():

    df = openFile()

    cleaner = Cleaner()

    df = cleaner.clean\_duration(df)

    df = clean\_duration(df)

    analyze\_duration(df)

**Вывод:**

Из графика можно заметить, что среднее значение продолжительности фильма лежит в промежутке от 90 до 100 минут, что подтверждает сформулированную гипотезу.

**Гипотеза 3 – 1994 год – лучший год в истории кинематографа.**

Для определения лучшего года построим гистограммы средних рейтингов по годам.

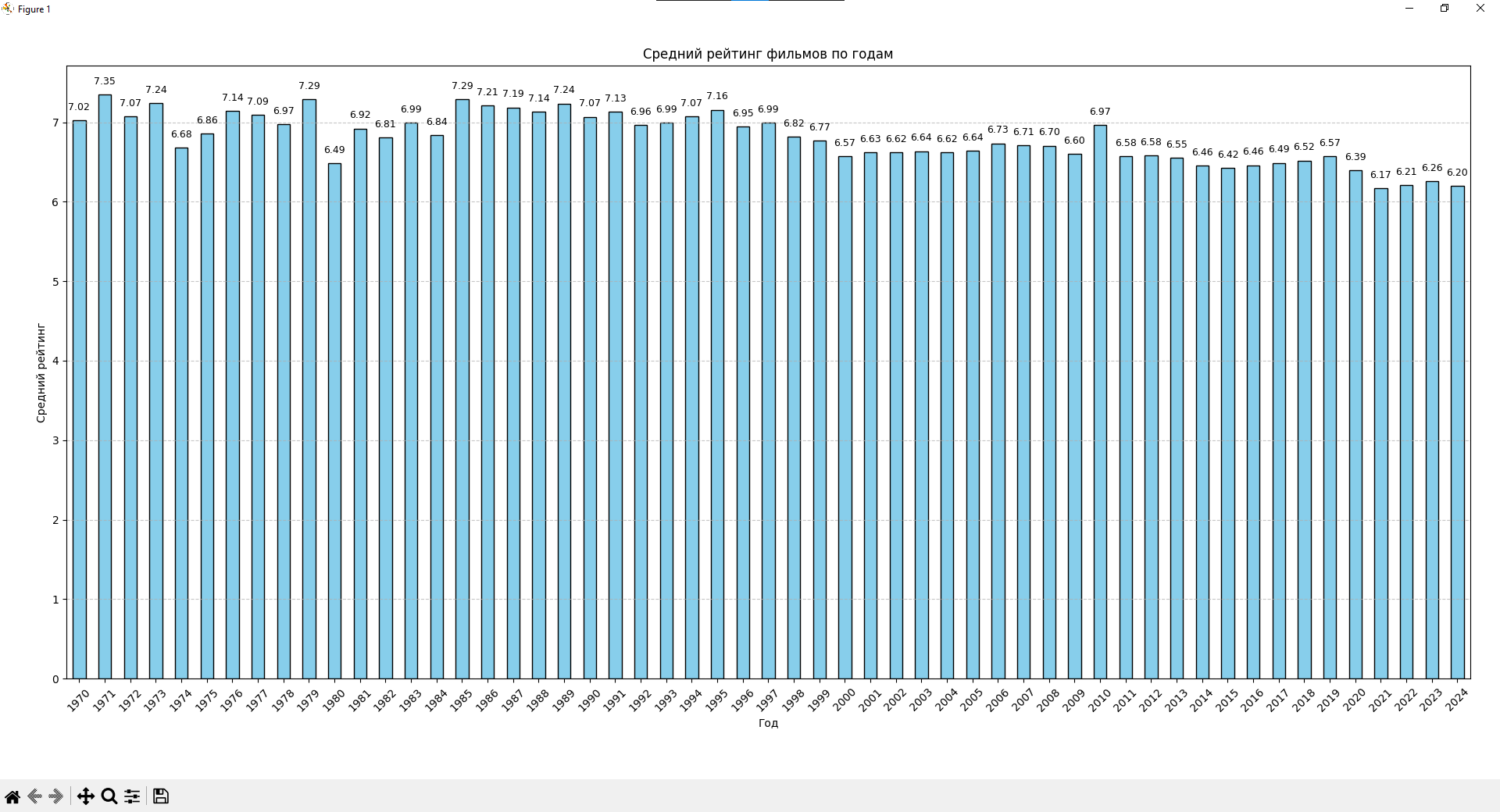


Рисунок 8 – гистограмма средних рейтингов фильмов по годам

Листинг:

# Функция для очистки и преобразования даты релиза

def extract\_year(df):

    df['Year'] = df['Release Date'].str[-4:]

    return df

# Функция для анализа среднего рейтинга фильмов по годам

def analyze\_average\_rating\_by\_year(df):

    df\_avg\_rating\_by\_year = df.groupby('Year')['Rating'].mean()

    df\_avg\_rating\_by\_year = df\_avg\_rating\_by\_year.sort\_index()

    plt.figure(figsize=(12, 6))

    df\_avg\_rating\_by\_year.plot(kind='bar', color='skyblue', edgecolor='black')

    for i, value in enumerate(df\_avg\_rating\_by\_year):

        plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=9)

    plt.title('Средний рейтинг фильмов по годам')

    plt.xlabel('Год')

    plt.ylabel('Средний рейтинг')

    plt.xticks(rotation=45)

    plt.grid(axis='y', linestyle='--', alpha=0.7)

    plt.tight\_layout()

    plt.show()

    print(f"Средний рейтинг по годам:\n{df\_avg\_rating\_by\_year}")

def main():

    df = openFile()

    cleaner = Cleaner()

    df = cleaner.clean\_duration(df)

    df = clean\_duration(df)

    df = extract\_year(df)

    df = df.dropna(subset=['Year', 'Rating'])

    analyze\_average\_rating\_by\_year(df)

**Вывод:**

Как можно увидеть из графика, лучшим годом в истории кинематографа является 1979 год, средний рейтинг фильмов, вышедших в этот год – 7.29. Средний рейтинг же фильмов 1994 года – 7.07. Гипотеза неверна.

**Гипотеза 4 - Количество выпускаемых фильмов растет с каждым годом.**

Для проверки данной гипотезы высчитаем количество фильмов датасета по годам. Для того, чтобы не было сильных спадов и подъемов, применим метод скользящей средней. Кроме того, поскольку датасет новый, он не содержит достаточно сведений о фильмах, снятых в последние годы, поэтому конечной датой будем считать 2022г.

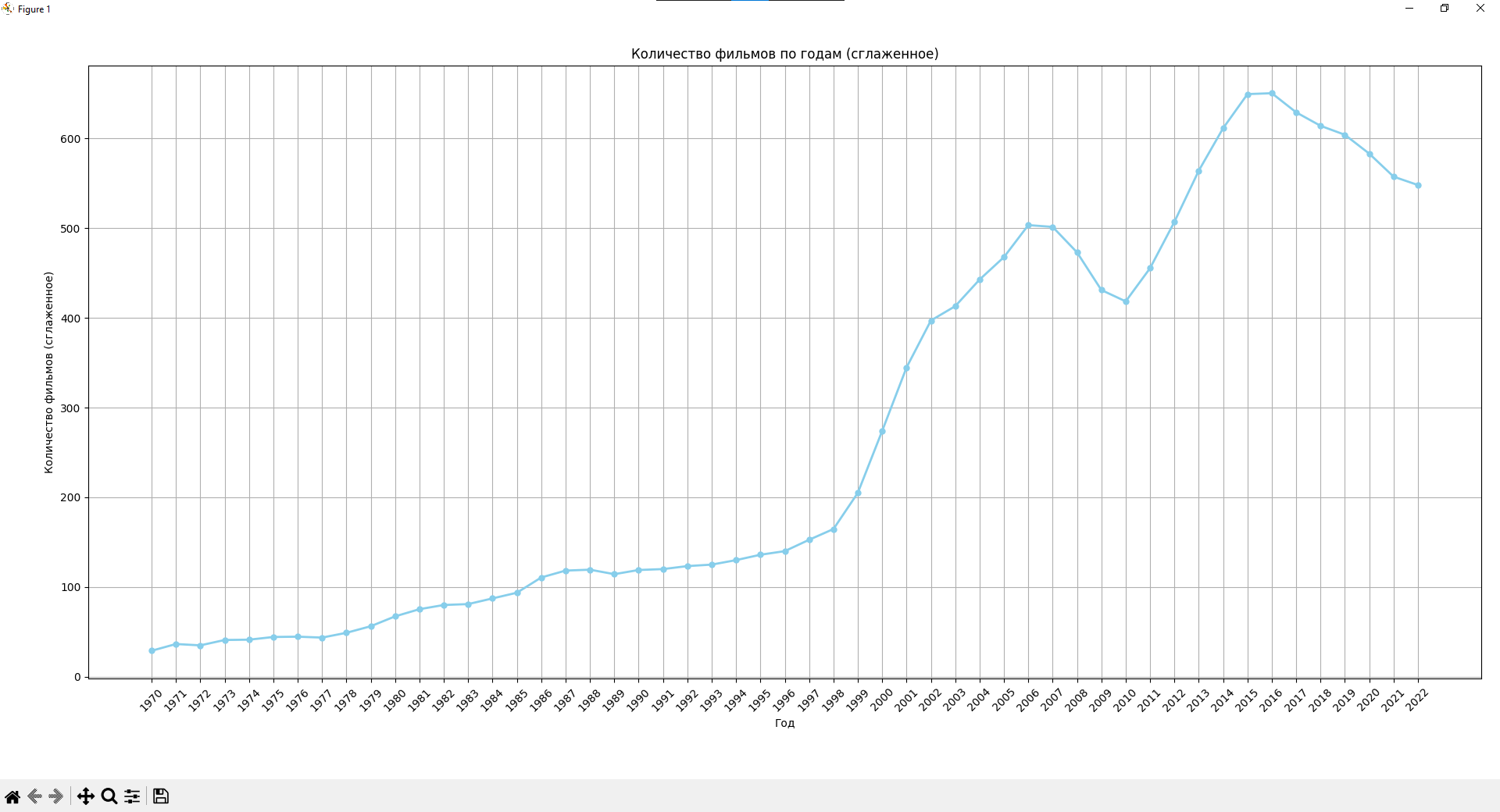
****

Рисунок 9 – линейчатая диаграмма количества выпускаемых фильмов

Листинг

# Функция для очистки и преобразования даты релиза

def extract\_year(df):

    df['Year'] = df['Release Date'].str[-4:]

    return df

# Функция для анализа количества фильмов по годам

def analyze\_movie\_count\_by\_year(df):

df = df[df['Year'].astype(int) <= 2022]

    df\_movie\_count\_by\_year = df.groupby('Year').size()

    df\_movie\_count\_by\_year\_smoothed = df\_movie\_count\_by\_year.rolling(window=3, min\_periods=1).mean()

    plt.figure(figsize=(12, 6))

    plt.plot(df\_movie\_count\_by\_year.index, df\_movie\_count\_by\_year\_smoothed.values, marker='o', color='skyblue', linestyle='-', linewidth=2, markersize=5)

    plt.title('Количество фильмов по годам (сглаженное)')

    plt.xlabel('Год')

    plt.ylabel('Количество фильмов (сглаженное)')

    plt.xticks(rotation=45)

    plt.grid(True)

    plt.tight\_layout()

    plt.show()

    print(f"Количество фильмов по годам:\n{df\_movie\_count\_by\_year}")

def main():

    df = openFile()

    cleaner = Cleaner()

    df = cleaner.clean\_duration(df)

    df = extract\_year(df)

    df = df.dropna(subset=['Year'])

    analyze\_movie\_count\_by\_year(df)

**Вывод:**

Как видим, тенденция к ежегодному росту количества снимаемых фильмов верна, временные спады 2008 – 2012гг. объясняются Мировым экономическим кризисом, а спад 2019 – 2022гг. объясняется пандемией короновируса. Тем не менее, на основании полученных данных, можно сделать вывод, что гипотеза верна.

**Анализ корреляции**

Листинг:

# Функция для анализа корреляции

def analyze\_correlation(df):

    df = transform\_text\_columns(df)

    numeric\_columns = df.select\_dtypes(include=['float64', 'int64']).columns

    corr\_matrix = df[numeric\_columns].corr()

    plt.figure(figsize=(12, 8))

    sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5, cbar=True)

    plt.title('Корреляционная матрица')

    plt.tight\_layout()

    plt.show()

    print("Корреляционная матрица:")

    print(corr\_matrix)

def main():

    df = openFile()

    cleaner = Cleaner()

    df = cleaner.clean\_duration(df)

    df = extract\_year(df)

    df = df.dropna(subset=['Year'])

    analyze\_correlation(df)

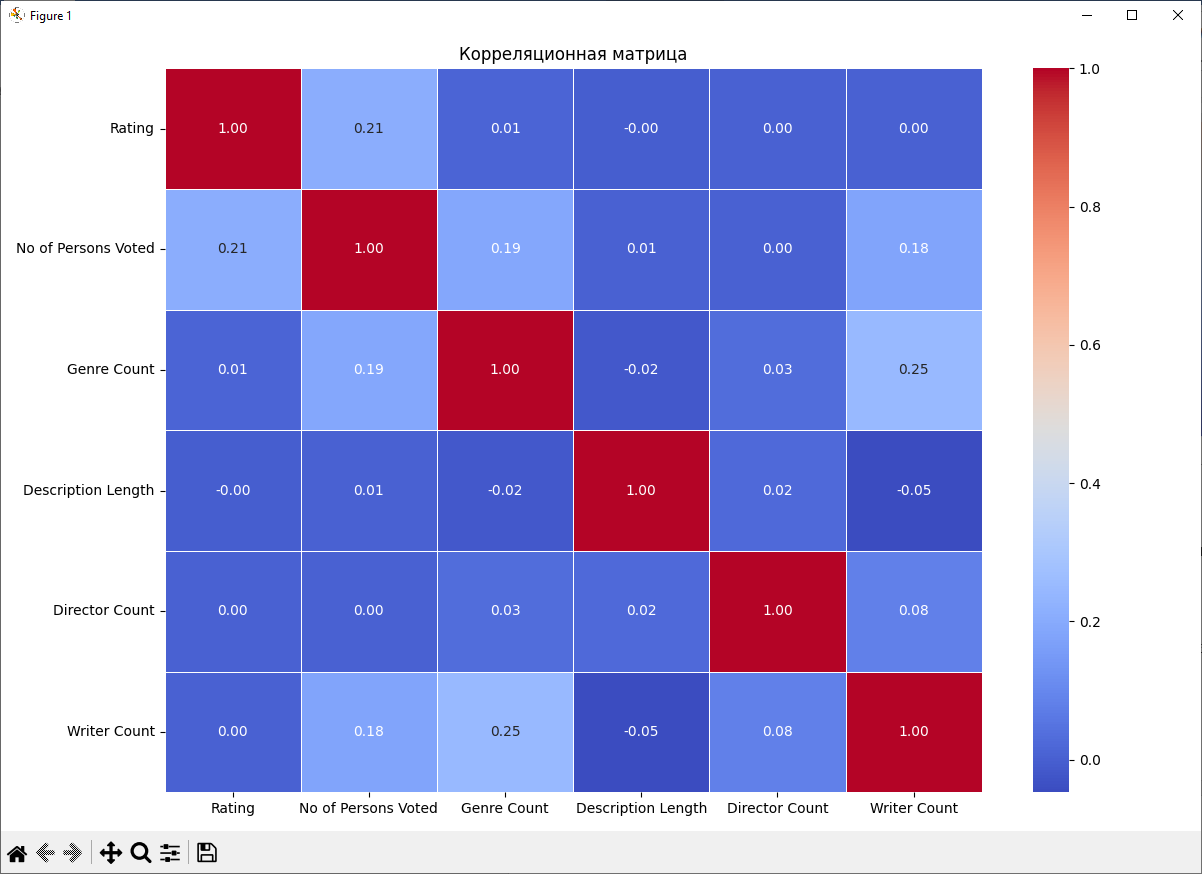
****

Рисунок 10 – корреляционная матрица

Выводы корреляционного анализа:

**ЗАКЛЮЧЕНИЕ**

В ходе научно-исследовательской работы был проведен анализ базы данных «16000+ Movies 1910-2024 (Metacritic)», выявлены важные статистические данные, которые помогли выявить тенденции современного кинематографа.

Для выполнения данной работы было изучено новое программное обеспечение: «Pandas», «Matplotlib» и «Seaborn» на базе языка программирования Python. Также мы пользовались «Visual Studio Code»: в нём мы писали код и анализировали диаграммы. В ходе работы я получил новые навыки в области анализа данных и познакомился с вышеперечисленными программами.

Была осуществлена оптимизация использования памяти, агрегирование данных, что в дальнейшем позволило провести необходимые в ходе работы исследования и получить ценные знания в предметной области.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Методические указания по программному обеспечению «Pandas»;
2. Методические указания по программному обеспечению «Seaborn»;
3. Методические указания по программному обеспечению «Matplotlib»;
4. «Pandas. Работа с данными» (2020), Автор: Абдрахманов М. И.;
5. «Python. Визуализация данных: Matplotlib, Seaborn, Mayavi»;
6. <https://devpractice.ru>;
7. <https://coderoad.wiki>;
8. <https://www.delftstack.com>.