|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Изображение выглядит как текст  Автоматически созданное описание | | | | | | |
|  | Министерство науки и высшего образования Российской Федерации | | | | | | |
|  | Федеральное государственное бюджетное образовательное учреждение  высшего образования  «МИРЭА – Российский технологический университет»  РТУ МИРЭА | | | | | | |
| **Институт** | | ИКБ | | | | |  |
|  | | | | |  | |  |
| **Специальность (направление):** | | | | Технологии разработки программного обеспечения | | |  |
|  | | | | | |  |  |
| **Кафедра:** | | КБ-3 «Безопасность программных решений» | | | | |  |
|  | | | | | |  |  |
| **Дисциплина:** | | | «Методы искусственного интеллекта» | | | |  |
|  |  |  |  |  |  |  |  |

**Практическая работа 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Студент: |  | |  | 28.09.2024 |  | A.   M. Ищенко |
|  | | *подпись* |  | *Дата* |  | *инициалы и фамилия* |
|  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа: | БСБО-01-22 |  | Шифр: | 22Б0661 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель: |  |  |  |  |  |
|  | *подпись* |  | *дата* |  | *инициалы и фамилия* |

**Москва 2024 г.**

**Задание**

Выбран датасет: [16000+ Movies 1910-2024 (Metacritic) (kaggle.com)](https://www.kaggle.com/datasets/kashifsahil/16000-movies-1910-2024-metacritic)

**1. Выбрать и обосновать метрику для измерения качества**

В рамках данного пункта необходимо подобрать наиболее релевантную метрику или набор метрик для вашей задачи, написав краткое обоснование   
(1–2 предложения).

Критерии оценки:

* Предложена метрика
* Присутствует корректное обоснование

Для данного датасета будет применена задача регрессии (предсказание рейтинга фильма) с метриками MSE (Mean Squared Error) и R². Для предсказания подобной непрерывной переменной подходит такая модель, как линейная регрессия. Оценка рейтинга будет зависеть от таких параметров, как жанр фильма, его продолжительность, количество голосов и другие характеристики.

**2. Провести EDA и предобработку данных.**

В рамках данного пункта необходимо провести предварительный разведочный анализ своего набора данных, проанализировать признаки и провести целесообразную предварительную обработку.

Критерии оценки:

* Рассмотрены базовые характеристики/статистики для набора данных
* Реализованы наиболее релевантные визуализации (не менее одной визуализации с применением plotly)
* По ходу работы присутствуют комментарии (вы, как эксперт в предметной области, можете проинтерпретировать полученные результаты анализа, сделав выводы о том, каким образом лучше работать с набором данных в будущем, какие могут возникнуть проблемы, какую предобработку целесообразно сделать и тп)
* Проведен анализ лучших практик работы с этим датасетом или с похожими данными (указать ссылки на эти работы)

Для начала импортируем все библиотеки и датасет:

import pandas as pd

import matplotlib.pyplot as plt

import plotly.express as plx

import seaborn as sns

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score, roc\_auc\_score, roc\_curve, auc

path = "16k\_Movies.csv"

df = pd.read\_csv(path)

df.head()

Вывод датасета:



Выведем количество строк и количество столбцов датасета:

df.shape



Из этого следует, что датасет состоит из 16290 строк и 10 столбцов.

Описание столбцов:

Unnamed: идентификатор

Title: название произведения

Release Date: официальная дата релиза фильма

Description: Краткий синопсис или критический обзор фильма

Rating: Средний балл на Metacritic, позволяющий судить о восприятии фильма критиками

Number of Persons Voted: Количество людей, оценивших фильм на Metacritic

Directed by: Режиссер(ы) фильма

Written by: Сценарист(ы), ответственный(ые) за сценарий фильма

Duration: Длительность фильма

Genres: Жанр(ы), к которому относится фильм, например драма, комедия или боевик

Получение информации о датасете:

#Удалим первый столбец без имени

df.drop(columns = 'Unnamed: 0', inplace=True)

print()

# покажет информацию о размерности данных и как данные индексируются, количество not-a-number элементов

print('Размерность, нумерация и не числовые значения набора данных (df.info()):\n',df.info())

print()

# показывает статистики: count,mean, std, min, 25%-50%-75% percentile, max

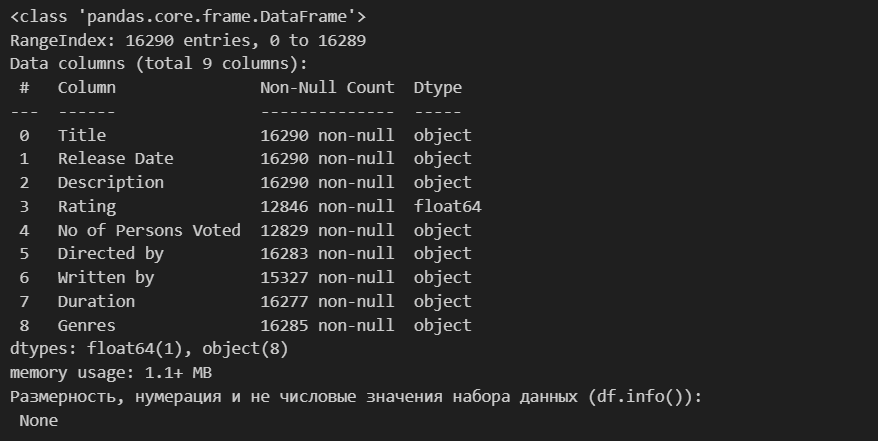
print('Статистики набора данных (df.describe()):\n',df.describe())

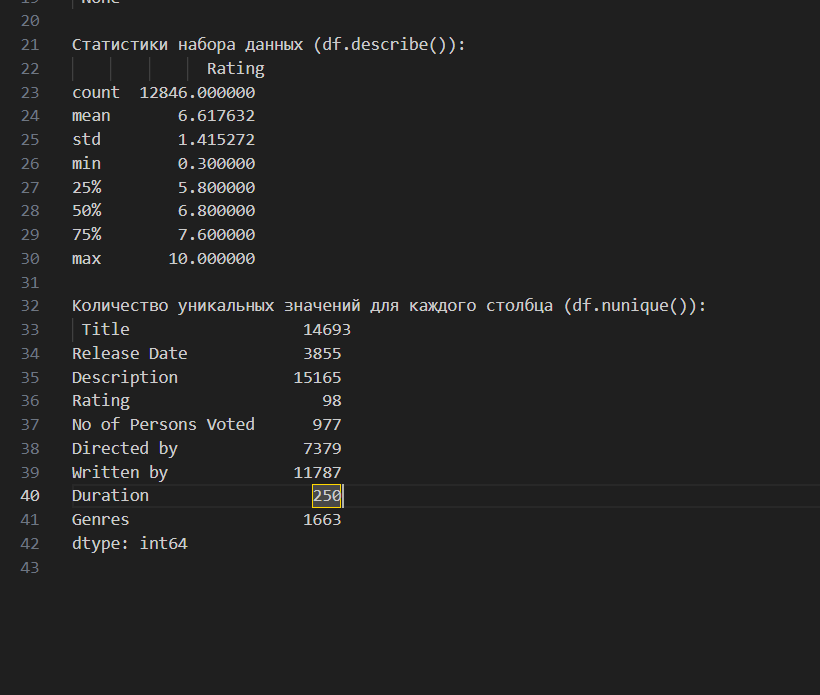
print()

# количество уникальных значений для каждого столбца.

print('Количество уникальных значений для каждого столбца (df.nunique()):\n', df.nunique())

Вывод:

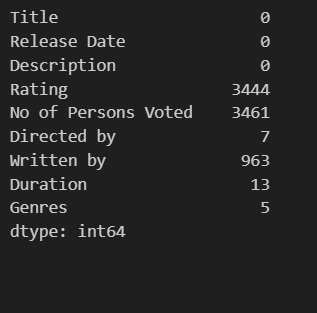




Теперь проверим датасет на наличие пустых значений:

# Проверим на пустые значения и посчитаем их для каждого столбца

df.isna().sum()



В столбцах присутствуют пустые значения:

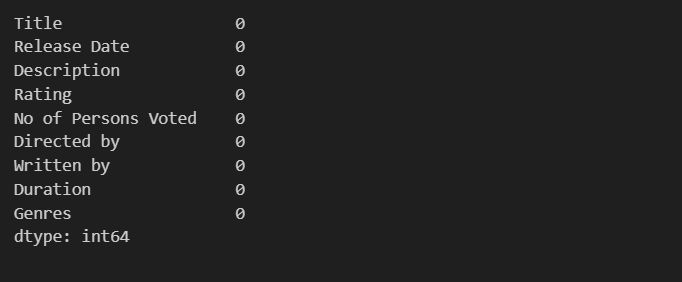
* Rating не был проставлен рейтинг
* No of Persons Voted есть фильмы, которые не оценивали
* Directed by отсутствует информация о режиссере
* Written by отсутствует информация о сценаристе
* Duration отсутствует информация о длительности
* Genres отсутствует информация о жанре

Удаление пустых значений в последних четырех столбцах незначительно скажется на результате, а вот удаление в первых двух может.

# Удаление пустых значений

df.dropna(inplace=True)

df.isna().sum()



Визуализируем распределение по всем столбцам при помощи гистограмм, кроме столбца с навыками в текстовом виде:

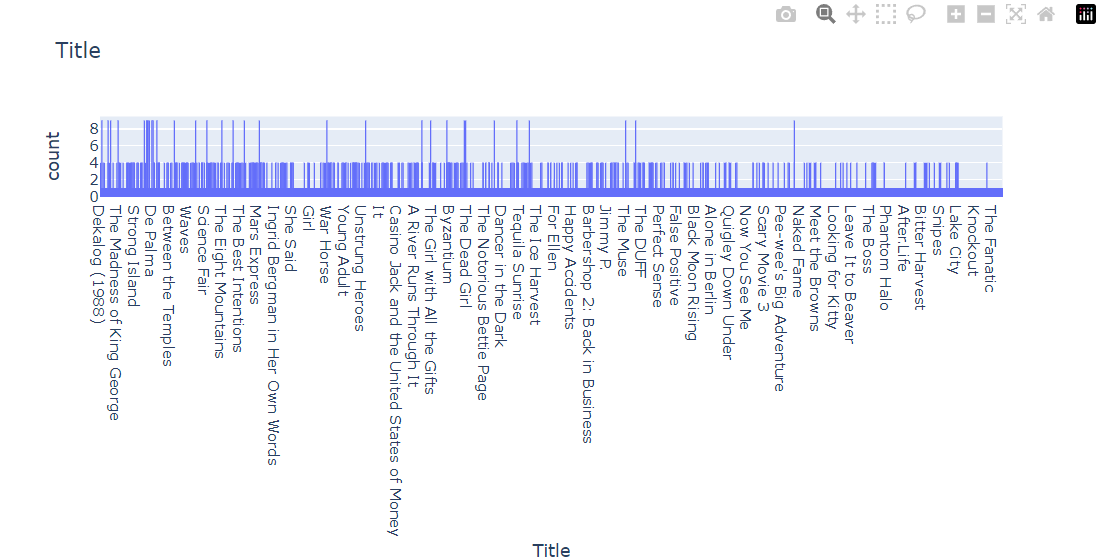
columns\_for\_figs = list(df.columns)

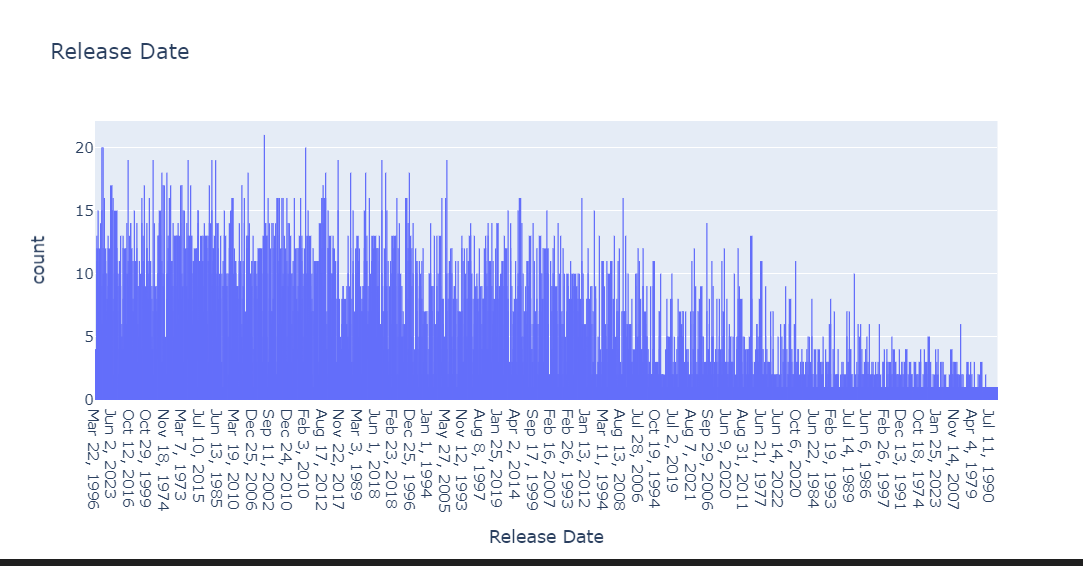
for col in columns\_for\_figs:

    if col!='Rating':

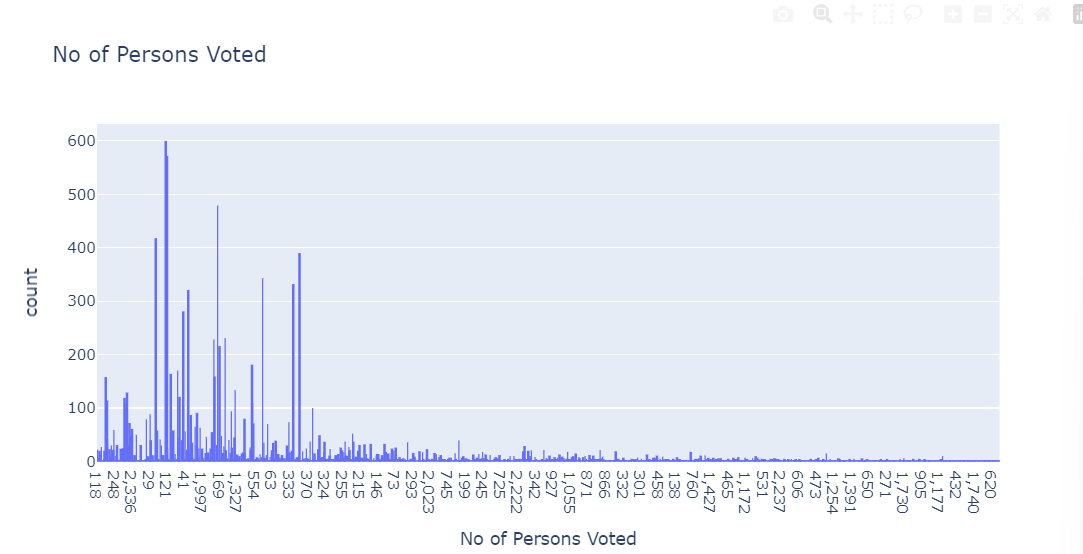
        fig = plx.histogram(df,x=col,title = col)

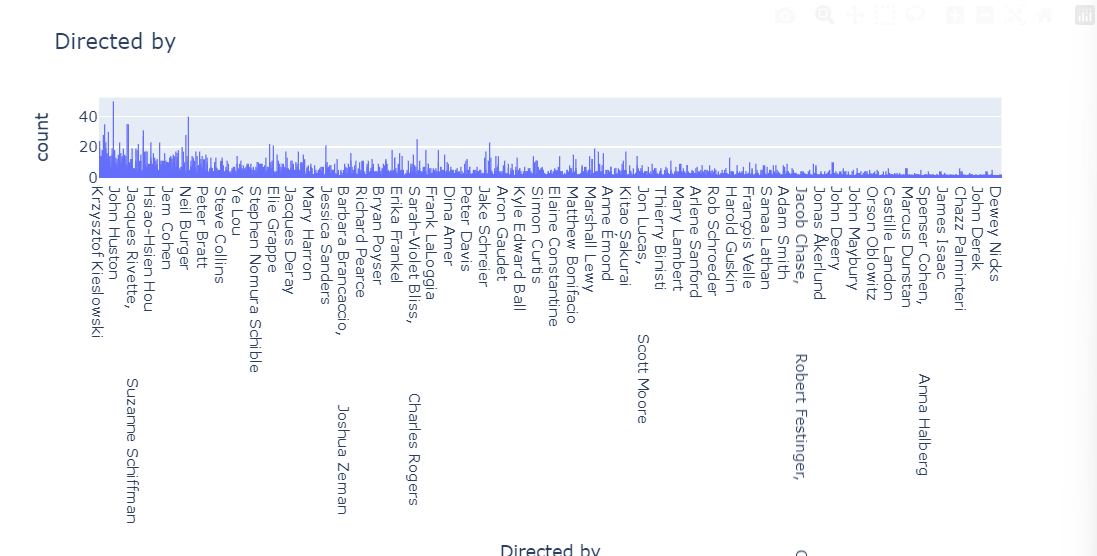
        fig.show()



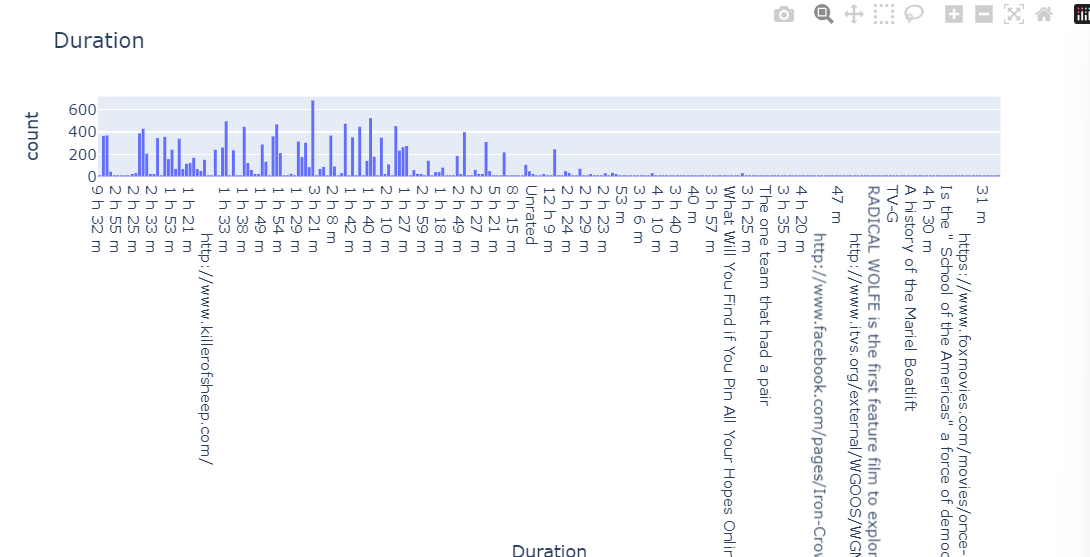














Проанализировав все гистограммы, были выявлены аномалии в столбце Duration, а именно вместо самой длительности присутствуют ссылки и прочие значения. Исправим это (переведем значения \* h \* m в минуты и отбросим оставшиеся строки):

import re

def convert\_to\_minutes(duration):

    if isinstance(duration, str):  # Проверяем, что значение строка

        # Ищем часы и минуты в строке

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

        if match:

            hours = int(match.group(1)) if match.group(1) else 0

            minutes = int(match.group(2)) if match.group(2) else 0

            total\_minutes = hours \* 60 + minutes

            return total\_minutes if total\_minutes > 0 else None  # Возвращаем None, если 0 минут

    return None  # Возвращаем None для значений, которые не являются строками

# Применим функцию к столбцу Duration

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

# Проверим результат

print("Первые строки после преобразования в минуты:\n", df['Duration'].head())

# Проверим, есть ли пропущенные значения после преобразования

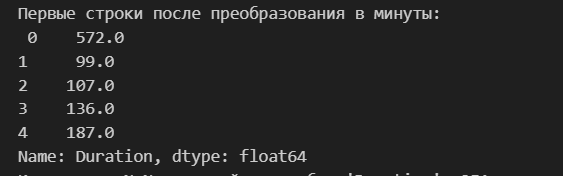
print("Количество NaN значений в столбце 'Duration':", df['Duration'].isna().sum())

# Удалим строки с NaN, если они присутствуют

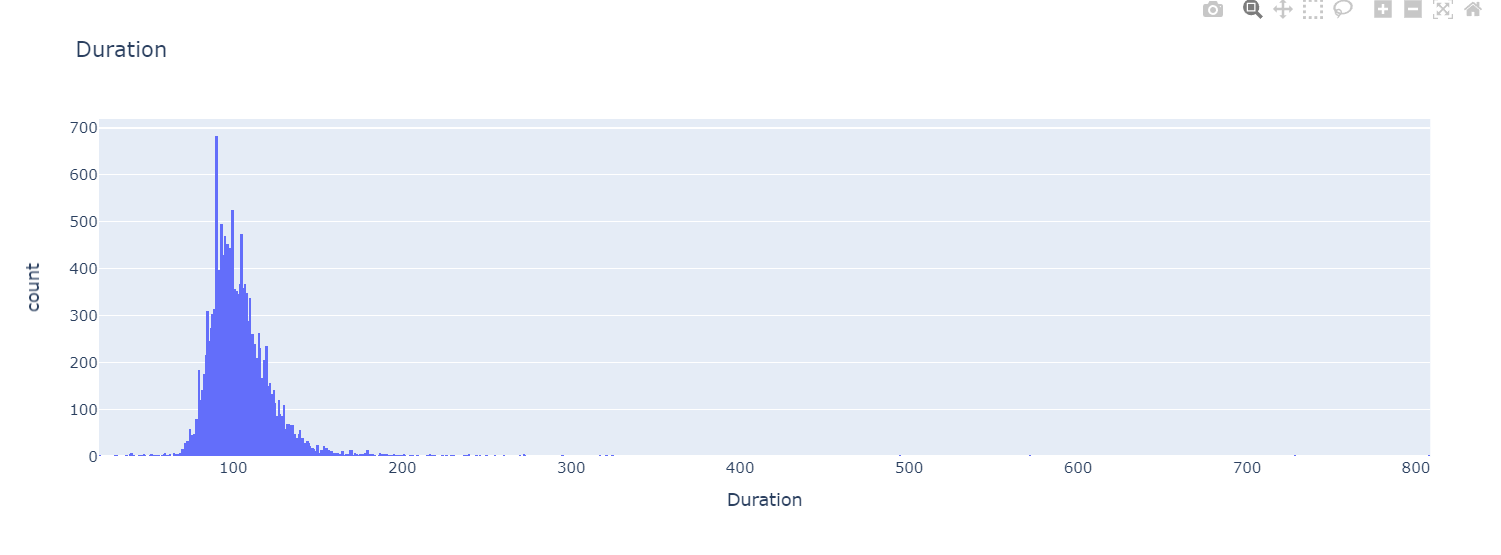
df = df.dropna(subset=['Duration'])

# Убедимся, что теперь столбец содержит только числовые значения в минутах

print("Тип данных столбца 'Duration':", df['Duration'].dtype)



Заново построим гистограмму для столбца Duration:



Получаем отличный результат без аномалий

Сделаем BoxPlot визуализации числовых значений для просмотра выбросов:

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

for column in num\_cols:

    sns.boxplot(data=df, x=column)

    plt.show()

