HTBLA Grieskirchen
4710 Grieskirchen, Parzer Schulstraße 1

Fachrichtung Informatik

Schuljahr 2024/2025

Protokoll

DSAI

Movies

Ausgeführt von:

Stefan Wiesinger, 5AHIF

Jakob Wintersteiger, 5AHIF

Grieskirchen, am 03.10.2024

Table of content

1 Numpy 1

2 Pandas 1

3 Unterschiede 4

1 Datensatz

Der folgende Datensatz wird in folgenden Auswertungen verwendet:

https://www.kaggle.com/datasets/ashishgup/netflix-rotten-tomatoes-metacritic-imdb/data

2 Auswertung

2.1 NumPy

2.1.1 Lese numerische Daten aus deinem Datensatz in ein NumPy Array

Aufgrund eines uns nicht verständlichen Errors war es uns nicht möglich mit NumPy die CSV einlesen.

```
import numpy as np

nparr = np.genfromtxt(file_path, delimiter=',', dtype=str, skip_header=1)
print(nparr)
```

```
ValueError: Some errors were detected !
    Line #3 (got 39 columns instead of 44)
    Line #4 (got 34 columns instead of 44)
```

2.1.2 Erstelle ein neues NumPy Array mit nur einer Spalte aus deinem Datensatz

```
import numpy as np

nparr = np.genfromtxt(file_path, delimiter=',', dtype=str, skip_header=1,
usecols=0)
print(nparr)
```

2.2 Pandas

2.2.1 Lese numerische Daten aus deinem Datensatz in ein Pandas DataFrame

```
import pandas as pd

# Read the CSV file into a DataFrame
file_path = './archive/netflix-rotten-tomatoes-metacritic-imdb.csv'
df = pd.read_csv(file_path)
```

2.2.2 Erstelle ein neues Pandas DataFrame mit nur einer Spalte aus deinem Datensatz

```
print(df.iloc[:, 0])
```

2.3 Observationen

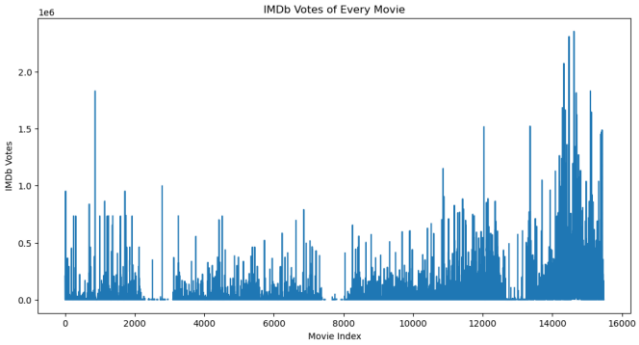
2.3.1 Erstelle eine statistische Analyse deines Datensatzes

2.3.1.1 Sind die Daten homogen verteilt? Gibt es Ausreißer?

Die Daten sind zufällig verteilt.

```
import matplotlib.pyplot as plt

# Plot IMDb Votes of every movie
plt.figure(figsize=(12, 6))
df['IMDb Votes'].plot(kind='line', title='IMDb Votes of Every Movie')
plt.xlabel('Movie Index')
plt.ylabel('IMDb Votes')
plt.show()
```

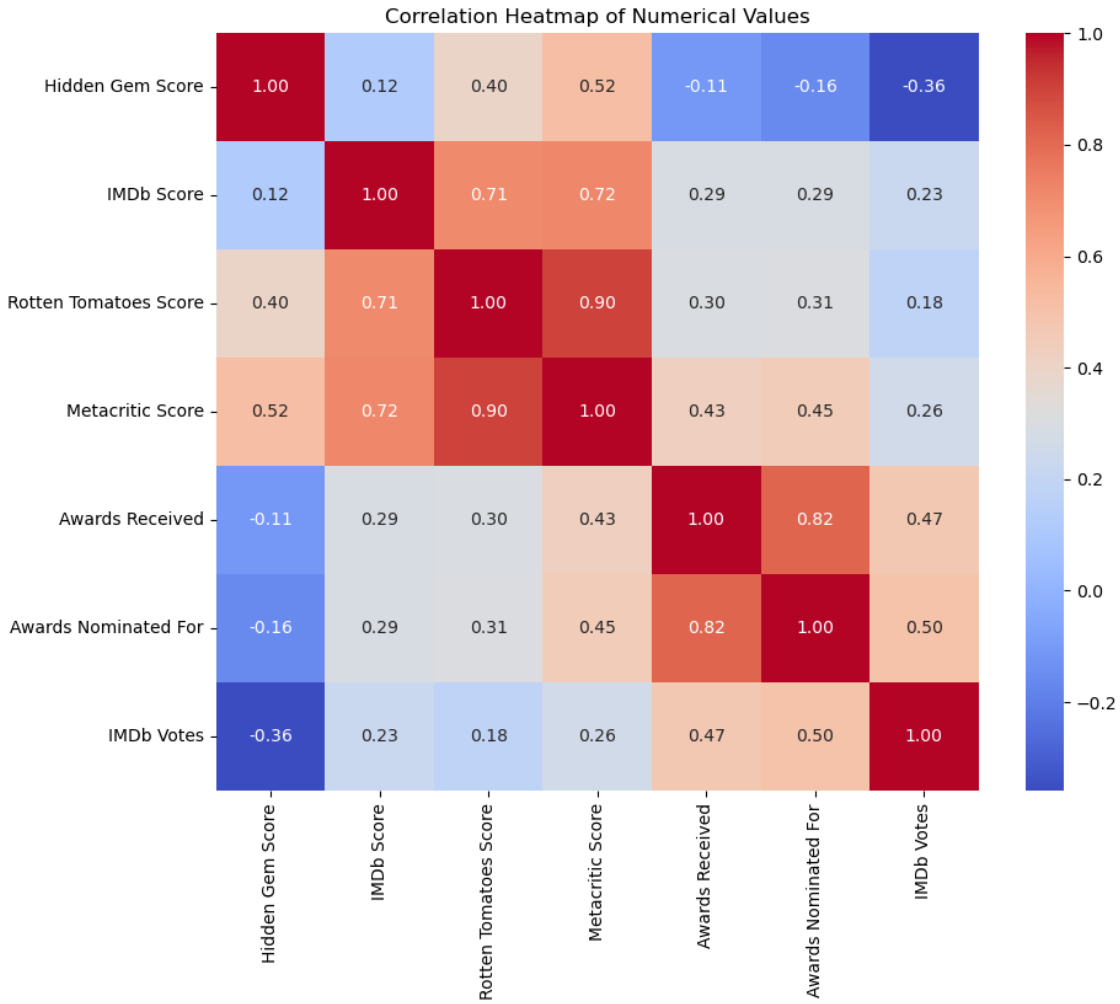


2.3.1.2 Kannst du durch die statistischen Werte schon irgendwelche Schlüsse aus deinen Daten ziehen?

```
import seaborn as sns
# Filter out non-numerical values
df_numerical = df.select_dtypes(include=['float64', 'int64'])

# Compute the correlation matrix
corr = df_numerical.corr()

# Generate a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of Numerical Values')
plt.show()
```



Wenn der Film auf einer Bewertungsseite gut bewertet ist, ist es wahrscheinlich dass er auf einer anderen auch gut bewertet ist. Auch hängt Awards Nominated wenig mit der Bewertung zusammen.

2.3.2 Suche dir eine Spalte zum Sortieren deiner Daten und gib die 3 niedrigsten und die 3 niedrigsten Werte aus

```
# Sort the DataFrame by IMDb Score in descending order and print the top
3 rows
top_3_imdb = df.sort_values(by='IMDb Score', ascending=False).head(3)
print("Top 3 movies by IMDb Score:")
print(top_3_imdb[['Title', 'IMDb Score']])

# Sort the DataFrame by IMDb Score in ascending order and print the bot-
tom 3 rows
bottom_3_imdb = df.sort_values(by='IMDb Score', ascending=True).head(3)
print("\nBottom 3 movies by IMDb Score:")
print(bottom_3_imdb[['Title', 'IMDb Score']])
```

```
Top 3 movies by IMDb Score:
               Title  IMDb Score
293      No Festival         9.7
15406   Breaking Bad         9.5
15314  Horsin Around         9.5

Bottom 3 movies by IMDb Score:
             Title  IMDb Score
10933  Be with You         1.0
608      Debt Fees         1.4
4497      Smoleńsk         1.4
```

2.4 Unterschiede

Es ist viel einfacher mit Panda zu arbeiten, Panda kann auch mit schlecht formatierten Input Dateien gut arbeiten.