National Institute of Technology, Kurukshetra
Haryana, India

Dr. Kapil Gupta
Assistant Professor
Department of Computer Application,
NIT Kurukshetra

Rhythm Deolus
MCA Student
NIT Kurukshetra

Adarsh Singh
MCA Student
NIT Kurukshetra

Ritik Sharma
MCA Student
NIT Kurukshetra

1. INTRODUCTION:

Dataset Title: TMDB 5000 Movies Dataset

Dataset Link: https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata.

Code: https://github.com/RhythmDeolus/TMDB-5000-Movies-Analysis

About the Dataset : The "tmdb 5000 movies" dataset provides comprehensive information about 5000 movies, including various columns:

budget: Production cost of the movie.
revenue: Total earnings of the movie.
genres: Comma-separated list of movie genres (e.g., "Comedy,Thriller").
original_language: Language the movie was originally filmed in.
original_title: The movie's title in its original language.
overview: A short summary of the movie's plot.
release_date: The date the movie was released.
runtime: Duration of the movie in minutes.
spoken_languages: A list of languages spoken in the movie.
title: The movie's title in English (if available).
popularity: A TMDB-defined popularity score.
vote_average: Average user rating on a scale of 1 to 10.
vote_count: Number of users who rated the movie.
production_companies: Names of the companies that produced the movie.
production_countries: Countries where the movie was filmed.

Additional Data:

keywords: A list of relevant keywords associated with the movie.
homepage: Link to the movie's official website (if available).
status: The current status of the movie (e.g., "Released").
tagline: A promotional tagline for the movie (if available).

2. DATA CLEANING:

2.1 Removing Irrelevant Columns:

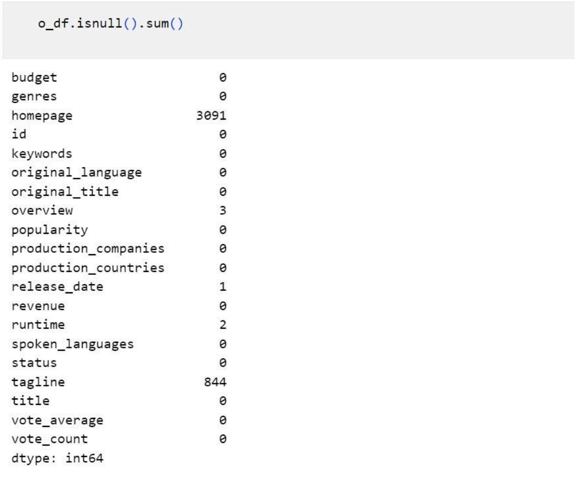
Overview: The overview column typically contains a brief description or summary of the movie. If this information is not relevant to the analysis or if it won't be used in the model, it can be removed.

Production Companies: This column lists the production companies involved in making the movie. If the analysis doesn't focus on production companies or if this information is not important, it can be removed.

Spoken Languages: This column lists the languages spoken in the movie. If language information is not relevant, it can be removed.

For now it can be hypothesized Production Companies and Spoken Languages are not relevant to our study which we will check later for this hypothesis.

2.2 Removing Null Values



```
o_df.isnull().sum()
```

```
budget                     0
genres                     0
homepage                3091
id                         0
keywords                   0
original_language          0
original_title             0
overview                   3
popularity                 0
production_companies       0
production_countries       0
release_date               1
revenue                    0
runtime                    2
spoken_languages           0
status                     0
tagline                  844
title                      0
vote_average               0
vote_count                 0
dtype: int64
```

Figure 2.1

2.2.1 Removing columns with a lot of null values:

Homepage: The homepage column contains URLs to the official websites of the movies. If many movies don't have a homepage listed or if this information is not necessary, it can be removed.

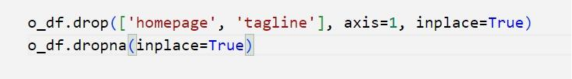
Tagline: The tagline column contains memorable phrases or taglines associated with the movies. If many movies lack a tagline or if this information is not useful, it can be removed.

2.2.2 Removing entries with null values:

Runtime: The runtime column indicates the duration of the movie. Entries with null values might indicate missing or incomplete data. If runtime is an important feature for analysis or modeling, entries with null values can be removed or imputed.

Release Date: This column indicates the date when the movie was released. Null values might indicate missing or incomplete data. Entries with null release dates might be removed if the analysis relies heavily on release date information.

Overview: As mentioned earlier, the overview column contains brief descriptions of the movies. Null values in this column might indicate missing or incomplete data. Entries with null overviews might be removed if overview information is important for the analysis.

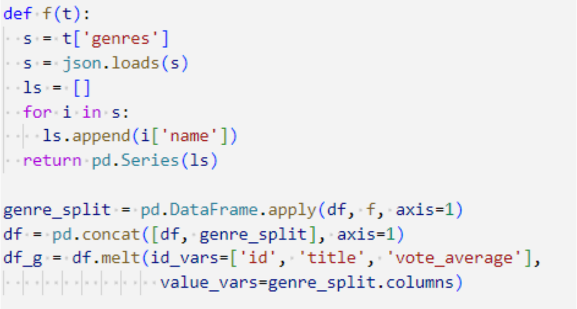


```
o_df.drop(['homepage', 'tagline'], axis=1, inplace=True)
o_df.dropna(inplace=True)
```

Figure 2.2

2.3 Using Pd.Melt To Convert Json Attributes Into Individual Rows:

Genre, Keywords, Production Countries: These columns might contain JSON-formatted data where each movie can have multiple genres, keywords, or production countries. Using pd.melt, you can convert these JSON attributes into individual rows, making it easier to analyze or model based on these attributes.

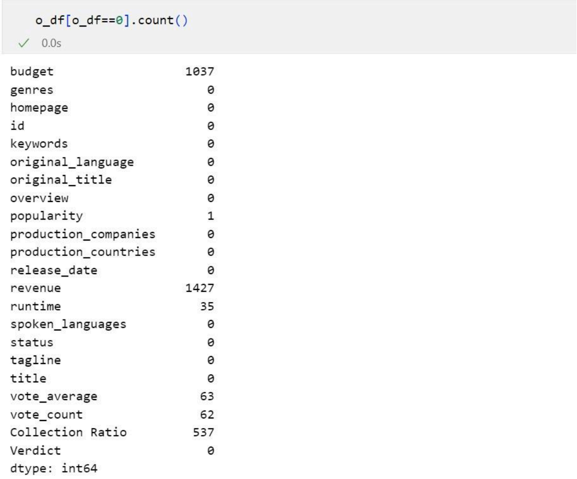


```
def f(t):
    s = t['genres']
    s = json.loads(s)
    ls = []
    for i in s:
        ls.append(i['name'])
    return pd.Series(ls)

genre_split = pd.DataFrame.apply(df, f, axis=1)
df = pd.concat([df, genre_split], axis=1)
df_g = df.melt(id_vars=['id', 'title', 'vote_average'],
                value_vars=genre_split.columns)
```

Figure 2.3

2.4 Case-Specific Removal Of Unexpected Zero Values In The Attribute Column:



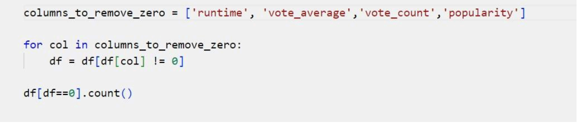
```
o_df[o_df==0].count()
✓ 0.0s

budget                  1037
genres                     0
homepage                   0
id                         0
keywords                   0
original_language          0
original_title             0
overview                   0
popularity                 1
production_companies       0
production_countries       0
release_date               0
revenue                 1427
runtime                   35
spoken_languages           0
status                     0
tagline                    0
title                      0
vote_average              63
vote_count                62
Collection Ratio         537
Verdict                    0
dtype: int64
```

Figure 2.4

Revenue, Budget: We can not remove these values since they amount to most of the dataset

Runtime, Vote Average, Vote Count, Popularity: We can remove these values as they don’t have that many rows



```
columns_to_remove_zero = ['runtime', 'vote_average','vote_count','popularity']

for col in columns_to_remove_zero:
    df = df[df[col] != 0]

df[df==0].count()
```

Figure 2.5

3. EXPLORATORY DATA ANALYSIS:

Looking at the relationship between No. of movies over the years.

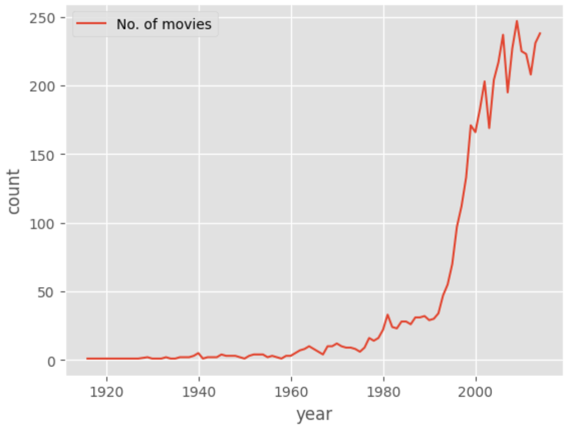


Figure 3.1

If we assume that our dataset represents the population distribution.

It can be inferred from the above graph that no. of movies are increasing over the year .

Looking at the relationship of average rating of movies over the decades.

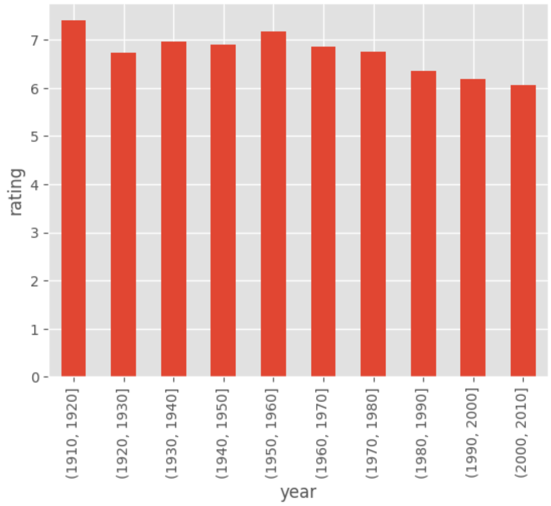


Figure 3.2

We could see a general trend of lower average rating for recent decades and there seems to be a trough at the decade 1950-60s.

It could be that the smaller sample size of older decades causes this variation.

Looking at the pie chart describing the composition of genres over movies.

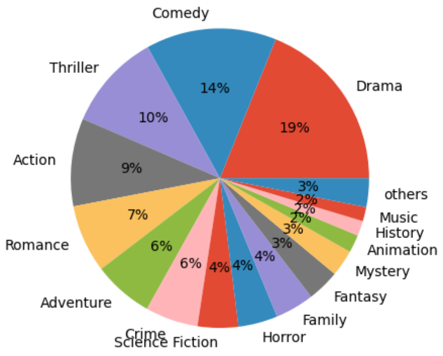


Figure 3.3

It can be inferred from this graph that the top 5 genres are:

Drama, Comedy, Thriller, Action, Romance

They correspond to 59% of the movies.

Looking at the top highest rated genres

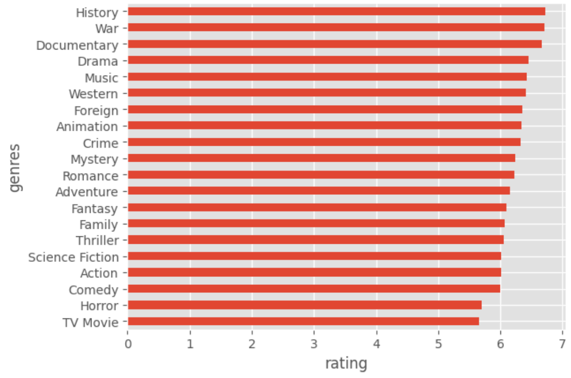


Figure 3.4

Here, It can be seen that History, War, Documentary genres seem to have significantly higher ratings than other genres.

Looking at the average rating over the year for the action genre.

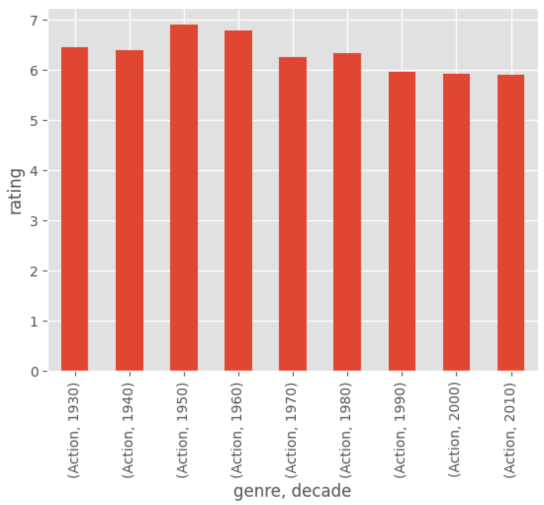


Figure 3.5

This graph also shows the similar trend of lower average rating for recent decades but with more variation.

Let's look at the histogram for average rating of movies.

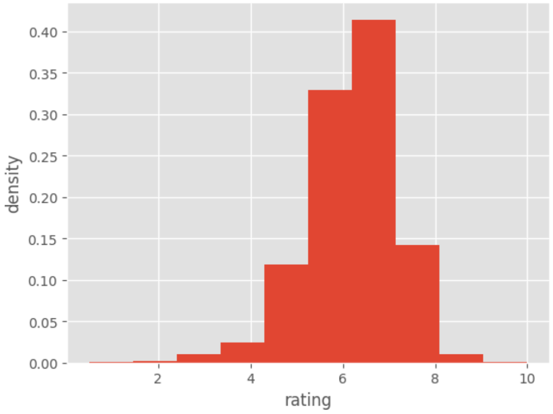


Figure 3.6

By looking at this graph it seems to be following a normal distribution which may or may not be true.

We will look at the evidence of it later in this report.

Looking at the scatterplot between vote average and release date where popularity is shown with colormap.

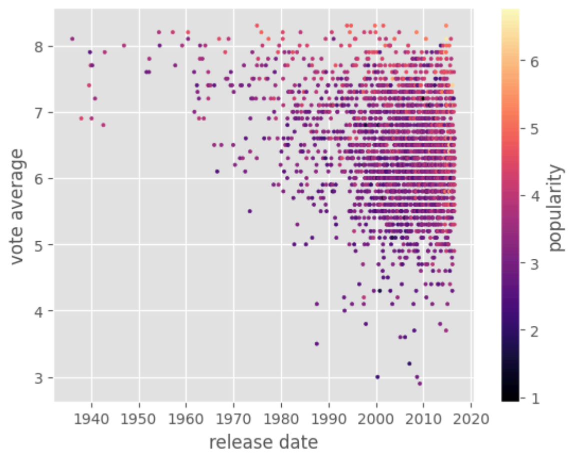


Figure 3.7

It can be inferred from the above graph that the movies that are recent and have higher vote average seems to be more popular.

Looking at a similar graph between popularity and release date.

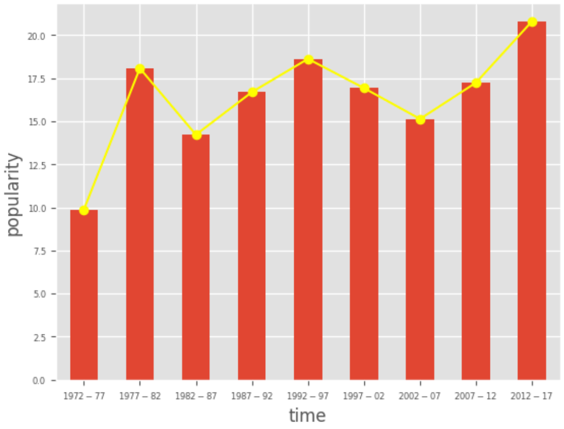


Figure 3.8

Here we can see this trend much more clearly (i.e. the movies that are more recent are more popular with some variation in this trend).

Similarly, looking at the graph between popularity and vote average:

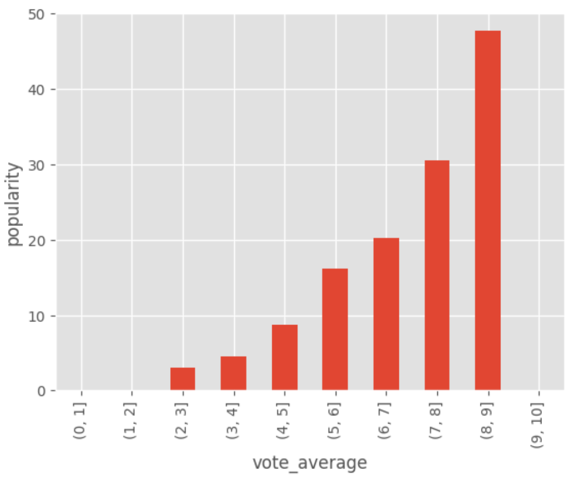


Figure 3.9

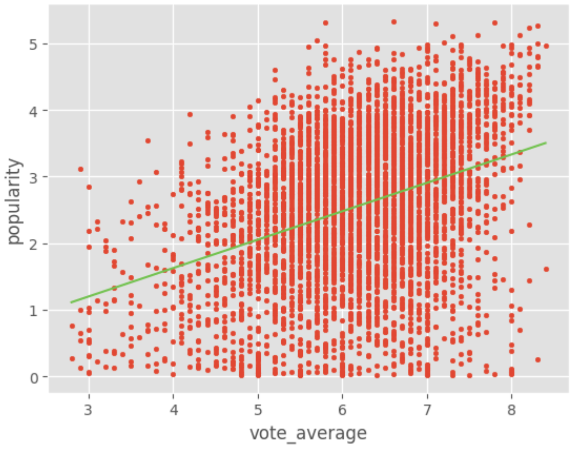


Figure 3.10

Figure 3.9 confirms the trend that higher average rating corresponds to more popularity when looked inside the window of (2, 9] average rating.

Similarly, Figure 3.10 shows a scatter plot with a linear regression line between popularity and vote average, confirming the trend of increase of popularity over vote average.

Looking at the average rating of movies with respect to their original languages in Figure 3.10 and their count in Figure 3.11 (i.e. no. of movies with that original language).

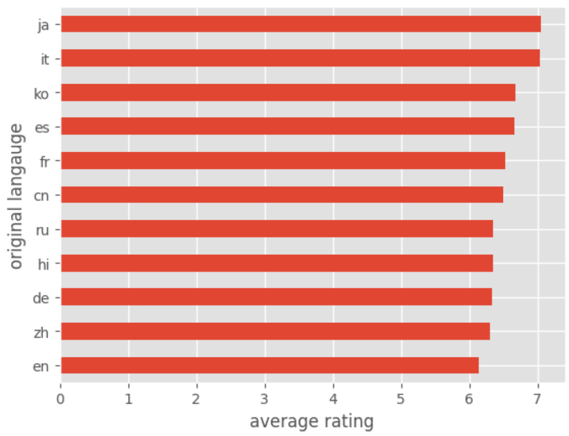


Figure 3.11

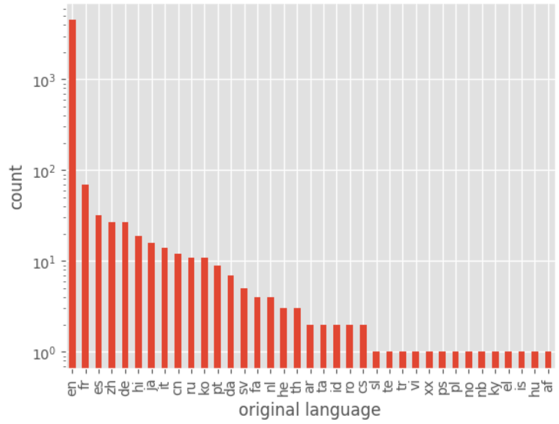


Figure 3.12

As we can see that there are some languages for which the average rating seems to be significantly high. These languages are: Japanese (ja), Italian (it).

But when inferred at the Figure 3.12 we can see that these languages have very little sample size so this inference of high average rating is not that reliable.

Looking at the vote average of keywords in Figure 3.13 and their count in Figure 3.14 (i.e.. no. of movies with that keyword).

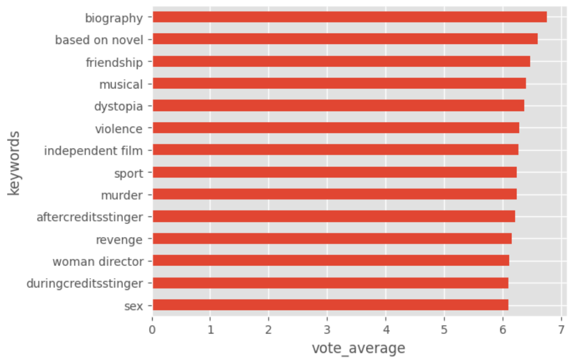


Figure 3. 13

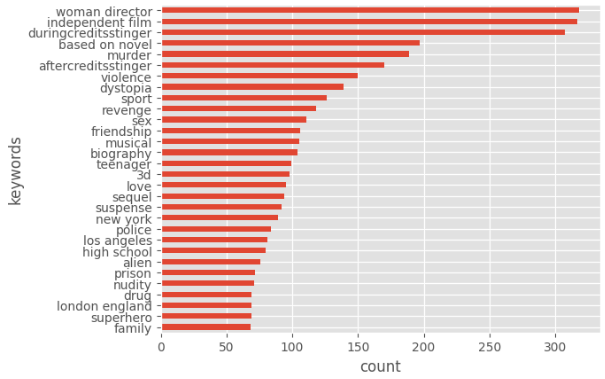


Figure 3.14

From Figure 3.13 it can be seen that these keywords: biography, based on novel, friendship correspond to high vote average.

From Figure 3.14 it can be observed that a significantly high no. of movies have keywords: Woman director, independent film, duringcreditsstinger.

Looking at the vote average for different bins of revenue generated by movies in Figure 3.15.

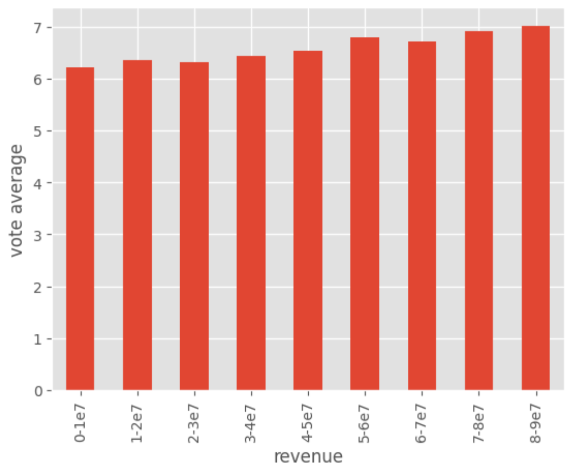


Figure 3.15

From Figure 3.15 it can be inferred that the higher revenue on average corresponds to higher vote average.

Looking at the average popularity score for different bins of budget.

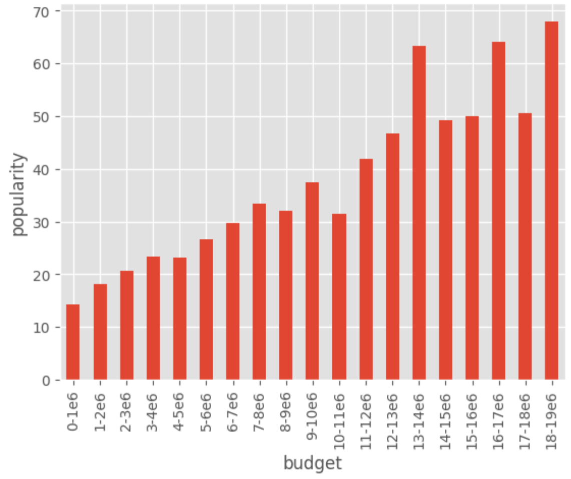


Figure 3.16

From Figure 3.16 it can be inferred that there is a general trend for increase in budget resulting in increased popularity but after increasing to a certain point there seems to be more variation in the popularity.

Looking at the scatter plot between revenue and budget with vote average as color map.

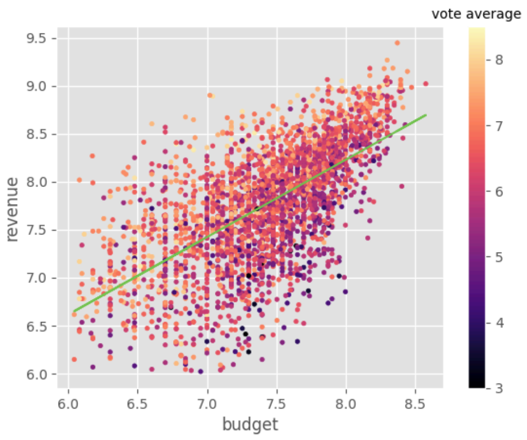


Figure 3.17

In Figure 3.17 the green line corresponds to the linear regression line between revenue and budget on a logarithmic scale.

We can see a general trend of increased revenue when we increase the budget. Also the vote averages that are high are generally above this regression line and vice versa. This could mean that on an increased budget of the movie the vote average and revenue tends to increase as well (i.e. there is a positive correlation between budget vs revenue and budget vs vote_average).

Looking at the graph between runtime vs average rating and runtime vs no. of samples

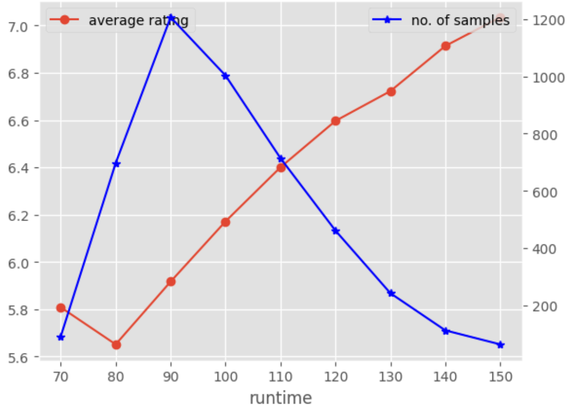


Figure 3.18

From the Figure 3.18 it can seen that under the window of 70-150 runtime the average rating seems to be decreasing at 80 and after the maximum dip it seems to be steadily increasing Similarly, we can see that No. of samples increase till 90 runtime and then start to steadily decrease as the runtime increases.

This could be due to the less sample size for high runtime movies creating increased average rating

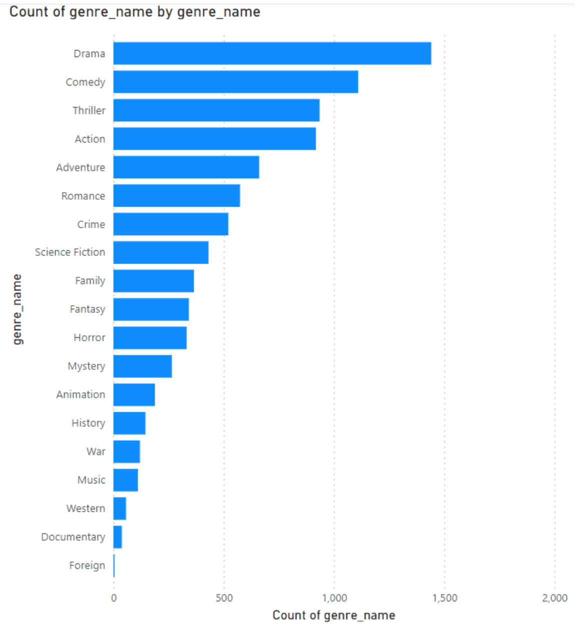


Figure 3.19

Figure 3.19 tells the number of movies present per genre.

Drama has the highest number of movies available.

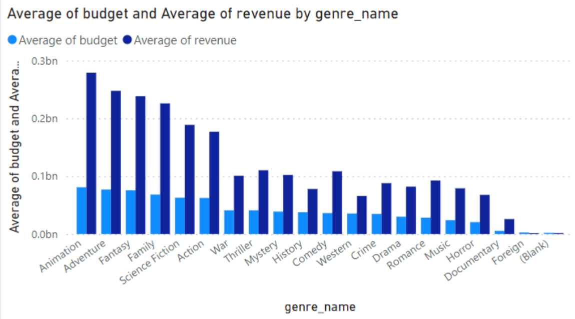


Figure 3.20

Figure 3.20 shows the average budget and revenue for each genre .
This can be beneficial , as it can provide a glimpse of what cost should be expected according to the genre , and the returns one can expect.

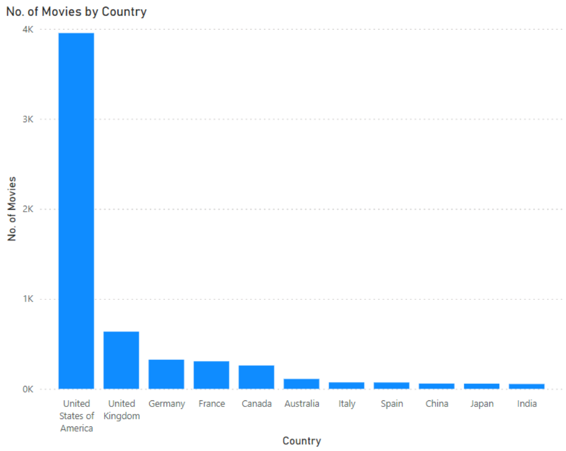


Figure 3.21

Figure 3.21 shows the number of movies available in the dataset on the basis of country .
Most of the movies are produced in the US , which is a much higher number than other countries.

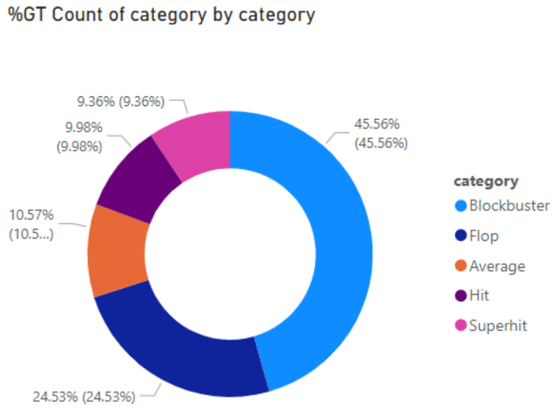


Figure 3.22

Figure 3.22 describes the classification of movies present on the basis of collection ratio.

Lack of correct standard has led to consideration of a simple parameter ‘Collection Ratio’ , which has been used to identify the categories.

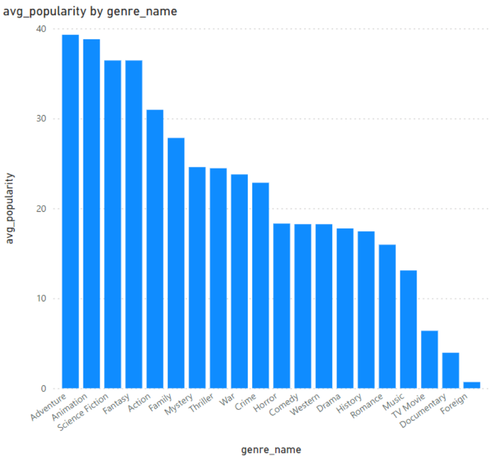


Figure 3.23

Figure 3.23 shows the average popularity of each genre.
It can be observed that the top 3 genres on this basis are ‘adventure’ , ‘animation’ and ‘science fiction’.

3.1 Verifying Claim of Normal Distribution for Vote Average

Aim: To identify if the vote averages follow a normal distribution.
Method: We will use the *Shapiro-Wilk Test* and the *Kolmogorov-Smirnov Test* to test our hypothesis.
Hypothesis:
H_0 = Vote average follows a normal distribution
H_1 = Vote average does not follow a normal distribution

Observation:

	Test Statistic	p-value
Shapiro-Wilk Test	0.862131123 2494561	5.30406031116 3094e-54

Kolmogorov-Smirnov Test	0.9809968163695512	0.0

Table 3.1.1

Result: since $p < 0.05$ for both of the tests we can say that our H_0 hypothesis is rejected (i.e. the vote average does not follow a normal distribution).

3.2 Conclusion:

1. No. of movies appears to be increasing over the years.
2. There's a general trend of lower average ratings for recent decades, possibly influenced by sample size variations.
3. The top 5 genres (Drama, Comedy, Thriller, Action, Romance) makeup 59% of the movies.
4. Genres like History, War, and Documentary tend to have higher ratings.
5. The average rating for the Action genre follows a similar trend of lower ratings for recent decades with more variation.
6. The histogram of average ratings does not follow normal distribution.
7. Recent movies with higher average ratings tend to be more popular.
8. Languages like Japanese and Italian show significantly high average ratings, but their sample sizes are small, so the inference might not be reliable.
9. Keywords like biography, based on novel, and friendship correlate with higher vote averages.
10. Keywords like Woman director, independent film, and duringcreditsstinger are common in movies.
11. Higher revenue generally corresponds to higher vote averages.
12. There's a trend of increased popularity with increased budget, but with more variation after a certain point.
13. There's a positive correlation between budget and revenue, as well as budget and vote average.
14. The average rating initially decreases around 80 minutes within the 70-150 runtime range, followed by a steady increase, while sample sizes peak around 90 minutes and decline steadily with longer runtimes.

4. CLASSIFICATION AND REGRESSION:

Correlation Matrix: This matrix contains the correlation coefficients between pairs of variables. It measures the strength and direction of the linear relationship between variables.

It's types:

Pearson Correlation Matrix: This matrix contains Pearson correlation coefficients, which measure the linear relationship between pairs of variables. It is widely used when variables are continuous and normally distributed.

Spearman Rank Correlation Matrix: This matrix contains Spearman correlation coefficients, which measure the monotonic relationship between variables. Spearman correlation is suitable for ordinal or non-normally distributed data.

Kendall's Tau Correlation Matrix: This matrix contains Kendall's tau correlation coefficients, which also measure the rank correlation between variables. Kendall's tau is robust to outliers and suitable for ordinal data.

Here we will be using Pearson Correlation Matrix

4.1 KNN Regression On Revenue

Aim: To do KNN Regression on revenue using relevant attributes

Assumption: The main assumption of the KNN Regression is that similar data points tend to have regression values. In other words, instances that are close to each other in the feature space are more likely to have similar target values.

Theory:

The formula for R-squared is:

$$R^2 = 1 - SS_{res} / SS_{tot}$$

Where:

SS_{res} is the sum of squares of residuals (also known as the sum of squared errors), which measures the total difference between the observed and predicted values of the dependent variable.

SS_{tot} is the total sum of squares, which measures the total difference between the observed values of the dependent variable and its mean.

In simpler terms, R-squared is calculated as 1 minus the ratio of the sum of squared residuals to the total sum of squares. R-squared values range from 0 to 1, where 0 indicates that the model does not explain any of the variability of the dependent variable, and 1 indicates that the model explains all of the variability.

Method:

Calculating correlation matrix and extracting columns with correlation of > 0.6 for using them to do regression on revenue.

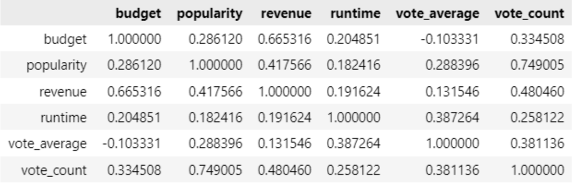
Split the dataset into two sets: train_set (80%), test_set(20%).

Training over values of k from 1 to 400 with step of 5 and taking the model with the least error value.

Plot the MSE vs k value graph and R-square vs k value graph.

Infer the Result from the obtained values.

Observation:



	budget	popularity	revenue	runtime	vote_average	vote_count
budget	1.000000	0.286120	0.665316	0.204851	-0.103331	0.334508
popularity	0.286120	1.000000	0.417566	0.182416	0.288396	0.749005
revenue	0.665316	0.417566	1.000000	0.191624	0.131546	0.480460
runtime	0.204851	0.182416	0.191624	1.000000	0.387264	0.258122
vote_average	-0.103331	0.288396	0.131546	0.387264	1.000000	0.381136
vote_count	0.334508	0.749005	0.480460	0.258122	0.381136	1.000000

Table 4.1.1

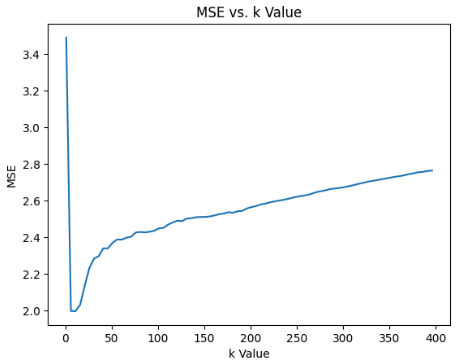


Figure 4.1.1

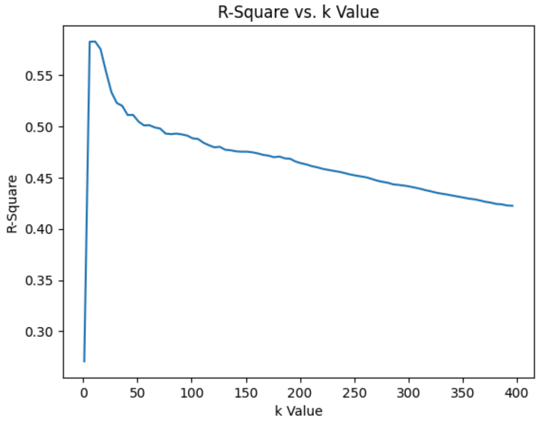


Figure 4.1.2

For minimum error point:

K-Value	26
Mean Squared Error	2.2323883586530644
R-squared:	0.5333726311975493

Table 4.1.2

Result:

From Figure 4.1.1 it can be observed that the mean square error value decreases as the k values increase in the beginning but after the maximum dip (k = 26) as you increase the value of k the mean square error starts to increase.

From Figure 4.1.2 it can be observed that the R-square value increases as the k values increase in the beginning but after the maximum point (k = 26) as you increase the value of k the R-Square value starts to decrease.

4.2 KNN Classification On Revenue

Aim: To do KNN Classification on revenue using relevant columns.

Assumption: The main assumption of the KNN Classification is that similar data points tend to have similar labels. In other words, instances that are close to each other in the feature space are more likely to belong to the same class.

Method:

Calculating correlation matrix and extracting columns with correlation of > 0.6 for using them to do regression on revenue.

Split the dataset into two sets: train_set (80%), test_set(20%).

Generate N Number of equal bins for revenue to classify.

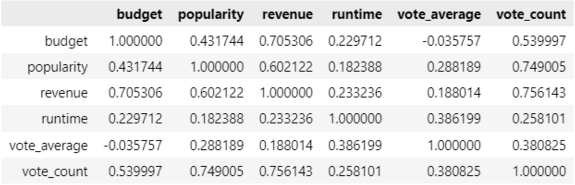
Iterating over values of N from 5 to 50 with step of 5.

Training over values of k from 1 to 100 with step of 5 and taking the model with the least error value.

Plot the N values vs K value graph, Accuracy vs K values for different values of N, Accuracy vs N values graph, F1 score vs N values, Precision Score vs N values, Recall Score vs N values.

Infer the Result from the obtained values.

Observation:



	budget	popularity	revenue	runtime	vote_average	vote_count
budget	1.000000	0.431744	0.705306	0.229712	-0.035757	0.539997
popularity	0.431744	1.000000	0.602122	0.182388	0.288189	0.749005
revenue	0.705306	0.602122	1.000000	0.233236	0.188014	0.756143
runtime	0.229712	0.182388	0.233236	1.000000	0.386199	0.258101
vote_average	-0.035757	0.288189	0.188014	0.386199	1.000000	0.380825
vote_count	0.539997	0.749005	0.756143	0.258101	0.380825	1.000000

Table 4.2.1

Hence we only consider attribute budget, popularity, and vote count for KNN classification.

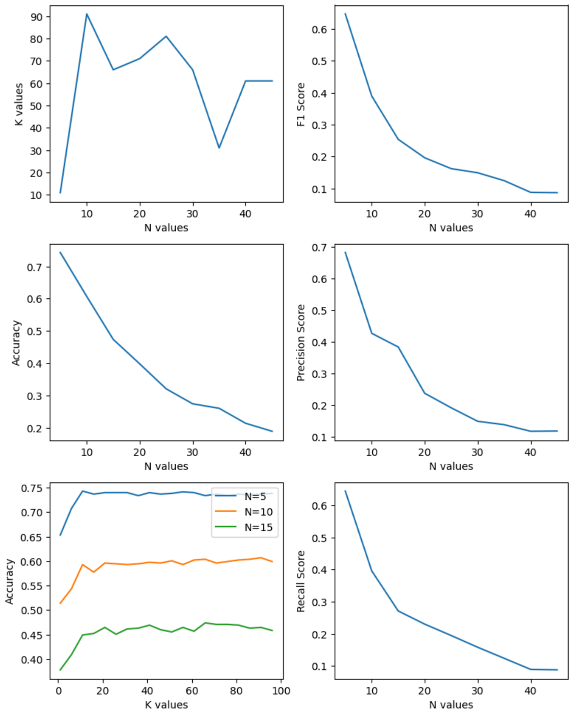


Figure 4.2.1

Result:

From the Figure 4.2.1 we can infer that as the number of bins increase the accuracy, f1 score, recall, precision score decrease.

As you increase the number of bins for revenue the prediction accuracy decreases.

It can be seen that as we increase the value of k the accuracy increases to a certain value and starts decreasing slowly (overfitting) this is true for different values of N.

4.3 Linear Regression on Revenue

Aim: To find the linear regression on revenue using relevant columns.

Assumption: The relationship between the independent variables and the dependent variable is linear. This means that the change in the dependent variable is proportional to the change in the independent variables.

Method:

Calculating correlation matrix and extracting columns with correlation of > 0.6 for using them to do regression on revenue.

Split the dataset into two sets: train_set (80%), test_set(20%).

Calculating the coefficients after linear regression for the extracted columns.

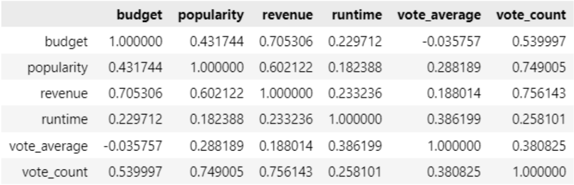
Test the model on the test_set calculating Mean Square Error and R-Square Value

Creating a new attribute of predicted revenue using the linear combination of the extracted columns.

Calculating the correlation between predicted revenue and revenue.

Infer the result from observation.

Observation:



	budget	popularity	revenue	runtime	vote_average	vote_count
budget	1.000000	0.431744	0.705306	0.229712	-0.035757	0.539997
popularity	0.431744	1.000000	0.602122	0.182388	0.288189	0.749005
revenue	0.705306	0.602122	1.000000	0.233236	0.188014	0.756143
runtime	0.229712	0.182388	0.233236	1.000000	0.386199	0.258101
vote_average	-0.035757	0.288189	0.188014	0.386199	1.000000	0.380825
vote_count	0.539997	0.749005	0.756143	0.258101	0.380825	1.000000

Table 4.3.1

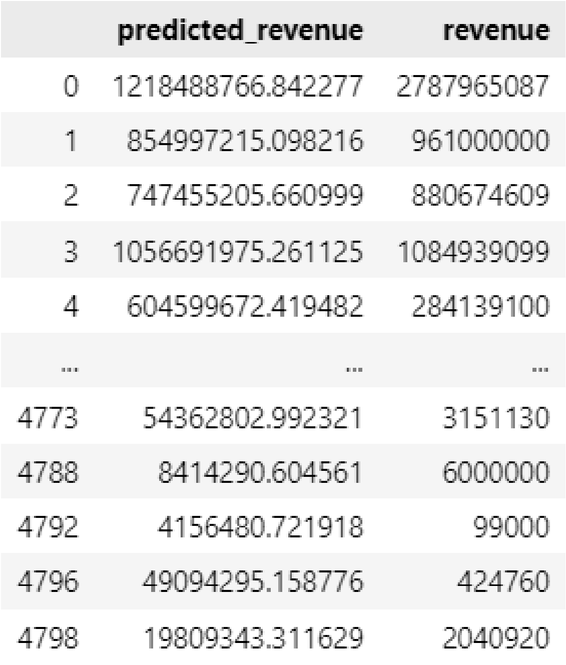
Hence we only consider attribute budget, popularity, and vote count for Linear Regression.

Coefficients after linear regression	
budget	1.7504823682143484
popularity	286193.6330347293
vote_count	64455.0990566868

Table 4.3.2

Test Result	
Mean Square Error	1.812788093343e+16
R-Square	0.6484653294499508

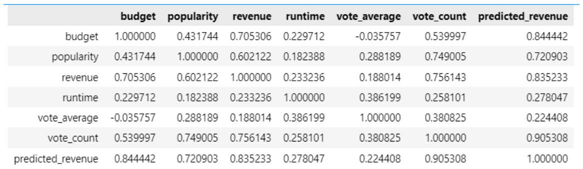
Table 4.3.3



	predicted_revenue	revenue
0	1218488766.842277	2787965087
1	854997215.098216	961000000
2	747455205.660999	880674609
3	1056691975.261125	1084939099
4	604599672.419482	284139100
...	...	...
4773	54362802.992321	3151130
4788	8414290.604561	6000000
4792	4156480.721918	99000
4796	49094295.158776	424760
4798	19809343.311629	2040920

Figure 4.3.1

Correlation between the predicted revenue and revenue:



	budget	popularity	revenue	runtime	vote_average	vote_count	predicted_revenue
budget	1.000000	0.431744	0.705306	0.229712	-0.035757	0.539997	0.844442
popularity	0.431744	1.000000	0.602122	0.182388	0.288189	0.749005	0.720903
revenue	0.705306	0.602122	1.000000	0.233236	0.188014	0.756143	0.835233
runtime	0.229712	0.182388	0.233236	1.000000	0.386199	0.258101	0.278047
vote_average	-0.035757	0.288189	0.188014	0.386199	1.000000	0.380825	0.224408
vote_count	0.539997	0.749005	0.756143	0.258101	0.380825	1.000000	0.905308
predicted_revenue	0.844442	0.720903	0.835233	0.278047	0.224408	0.905308	1.000000

Figure 4.3.2

Histogram of predicted revenue and revenue for showing their distribution:

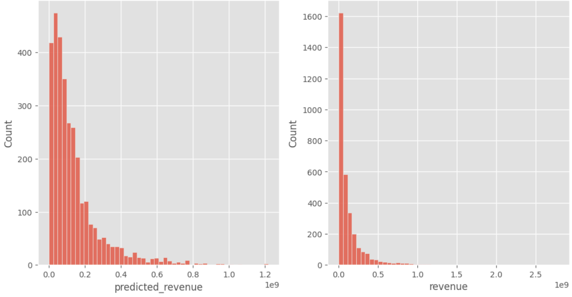


Figure 4.3.3

Result:

It can be seen that the predicted revenue has high correlation (i..e. > 0.8) with actual revenue and both form a similar distribution.
Hence we can say that budget, popularity and vote_count are good measures for predicting the revenue.

4.4 Overfitting And Underfitting

4.4.1 KNN Model

With respect to the value of k (number of nearest neighbors)

Overfitting in KNN can occur when the value of k (number of nearest neighbors) is too small, as it may lead to the model capturing noise or outliers in the training data. However, increasing the value of k tends to lead to underfitting.

With respect to the number of attributes (features) used for prediction

With too many attributes, KNN may fit the training data too closely, capturing noise instead of underlying patterns leading to *overfitting*
With too few attributes, KNN may be too simple to capture the relationships in the data, resulting in poor performance and leading to *underfitting*.

How to mitigate:

Feature Selection and Engineering:

Overfitting: Choose the most informative attributes while excluding irrelevant or noisy ones to reduce model complexity and focus on relevant patterns.
Underfitting: Create new features or increase the number of relevant attributes to enrich the data representation and capture more complex relationships.

Cross-Validation: Assess model performance using cross-validation and fine-tune hyperparameters (including both k and the number of attributes) to find the optimal balance between bias and variance, addressing both underfitting and overfitting.

Dimensionality Reduction: Use techniques like PCA or t-SNE to reduce the dimensionality of the feature space while preserving important information. This helps mitigate overfitting and avoids the curse of dimensionality, while also enriching the data representation to address underfitting.

4.5 Testing For Hypothesised Irrelevant Columns

Aim: To test if Production Company and Spoken Languages were relevant to our study or not.

Hypothesis:

H_0 = There is no correlation between Production Company vs Revenue and Spoken Language vs Revenue

H_1 = There is correlation between Production Company vs Revenue or Spoken Language vs Revenue

Method:

Convert the ordinal values into one-hot encoded values for each categories (i.e. for production companies and spoken languages)
Then use that encoding to find the spearman correlation between revenue and each category
If the correlation is significant between the any category and revenue (i.e > 0.6) then we can reject our H_0 Hypothesis

Observation:

For Production Companies:
Maximum correlation between revenue and a category = 0.2232284506897029
Average correlation = -0.0011096810650601365

For Spoken Languages:
Maximum correlation between revenue and a category = 0.09356865617796768
Average correlation = 0.005965953968371287

Result:

It can be seen that since no correlation is greater than 0.6 we can not reject our H_0 hypothesis.

5. SPATIAL ANALYSIS

5.1 Spatial Autocorrelation Of Average Rating

Aim: To find the spatial autocorrelation of Average Rating over the world map.

Prerequisites: naturalearth_lowres dataset

Hypothesis:

H_0 = No spatial autocorrelation in average rating

H_1 = Spatial Autocorrelation in average rating

Method:

Generate a new column for countries using production_countries, pd.melt, json (refer Section 2.3).

Combine this newly created column with the dataset naturalearth_lowres using countries names.

Calculate Spatial weights.

Calculate Moran's I and Moran's p-value.

Draw a graph between Spatial Lag and Average Rating.

Infer the result from observation.

Observation:

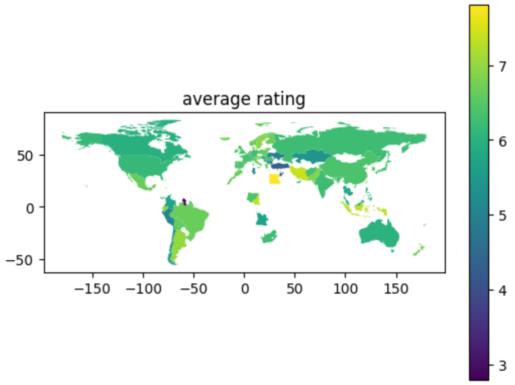


Figure 5.1.1

Moran's I	-0.040284638348214274
Moran's I p-value	0.44

Table 5.1.1

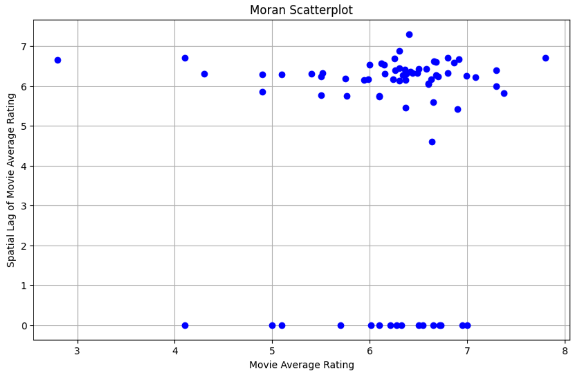


Figure 5.1.2

Result: As the p-value is >> 0.05, we can not reject the H_0 hypothesis (i.e. there is no spatial autocorrelation in average rating).

5.2 Spatial Autocorrelation Of No. Of Movies

Aim: To find the spatial autocorrelation of No. of movies over the world map.

Prerequisites: naturalearth_lowres dataset

Hypothesis:

H_0 = No spatial autocorrelation in no. of movies

H_1 = Spatial Autocorrelation in no. of movies

Method:

Generate a new column for countries using production_countries, pd.melt, json (refer Section 2.3).

Combine this newly created column with the dataset naturaleart_lowres using countries names.

Calculate Spatial weights.

Calculate Moran's I and Moran's I p-value.

Draw a graph between Spatial Lag and No. of movies.

Infer the result from observation.

Observation:

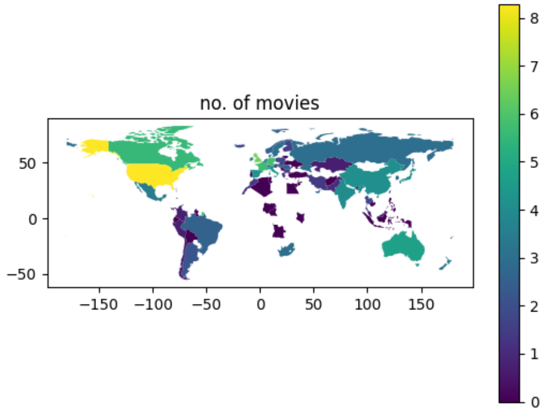


Figure 5.2.1

Moran's I	0.5277238171423252
Moran's I p-value	0.001

Table 5.2.1

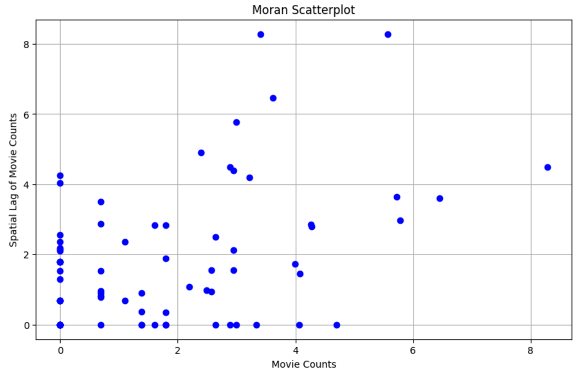


Figure 5.2.2

Result: As the p-value is < 0.05, we can reject the H_0 hypothesis (i.e. there is no spatial autocorrelation in average rating).

Hence seems to be a spatial autocorrelation between no. of movies and countries.

6. TESTING AVAILABILITY ON DIFFERENT OTT PLATFORMS

Aim: To test availability of movies on different OTT Platforms using self defined metrics

Prerequisites: amazon_prime_titles dataset[2], hotstar dataset[3], netflix_titles dataset[1]

Theory:

We are defining two metrics for measuring availability:

Simple Availability Score: we define this as the no. of movies common in our dataset and the subject OTT dataset.

Pop-Avail Score: we define this as the sum of popularity score of the common movies in our dataset and the subject OTT dataset.

Method:

Load our dataset, netflix dataset, hotstar dataset, amazon prime dataset.

Join Both dataset based on their title.

Calculate the *Simple Availability Score* and *Pop-Avail Score* for all the subject OTT datasets with respect to our dataset.

Infer from the result the most available OTT platform.

Observation:

	Netflix	Hotstar	Amazon Prime
Simple Availability	784	284	410
Pop-Avail Score	22869.797002	14048.223948	7456.656226

Table 6.1

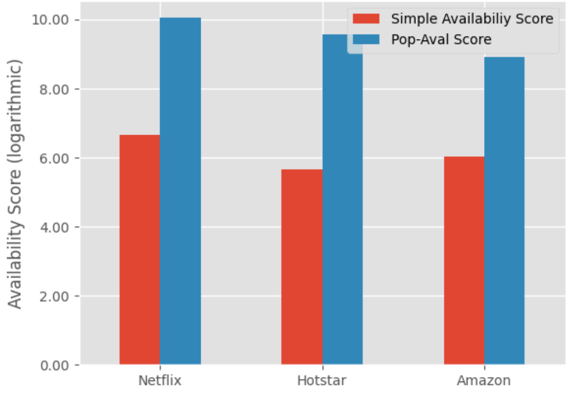


Figure 6.1

Result: Looking at the *Simple Availability Score* we can say that Netflix has the most availability, then Amazon Prime and then Hotstar.
But, looking at the *Pop-Avail Score* it can be seen that Netflix has the most availability , then Hotstar and then Amazon Prime.
In other words, Netlflix has the most availability in terms of popular movies as well as number of movies, while hotstar has the second most availability in terms of popular movies but the third in the term of no. of movies and vice versa for Amazon Prime.

REFERENCES

[1]:https://www.kaggle.com/datasets/shivamb/netflix-shows?resource=download

[2]:https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows

[3]:https://www.kaggle.com/datasets/goelyash/disney-hotstar-tv-and-movie-catalog