Prediction of Netflix Movie Rating (user Review)

1. Introduction:

Netflix, a subscription-based streaming service offers online streaming of films and television series. Starting in 1997 in USA, now it makes the service available in most of the countries in the world. It is well known for its efficient recommendation engines providing users choice. The engines work behind the scenes and provide user's choice related contents. The engines use Content-based filtering algorithm, collaborative filtering algorithm or hybrid of both.

User rating(review) plays an important role in recommendation system. The users (called subscribers/viewers/member) rate movies based on various features of a movie: genre, actor, director, title, language, country, duration, production company etc.

2. Problem statement:

Predictive models will be developed to predict movies users' rating and the best one will be selected based on the R2-score (co-efficient of determination) i.e how close the actual ratings are to the predicted values.

3. Data:

The data was collected from Kaggle and IMDB website. The links are

- https://www.kaggle.com/shivamb/netflix-shows
- https://www.imdb.com/interfaces/

4. Data Cleaning/wrangling:

Both datasets list movies and their features as columns. However, based on my requirement, I selected the features those required for the predictive model building. To achieve this, I carried out

data wrangling as follows:

- **Problem-1:** Kaggle dataset listed both movies and TV shows.
  Solution: I separated out the TV shows and only kept the movies for further data processing and model building.

- **Problem-2:** The IMDB dataset has listed titles and original titles for movies.
  Solution: I dropped the 'title' column and preserved only the 'original_title' column.

- **Problem-3:** The movie duration time column had both numbers and unit (minute).
  Solution: Unit was deleted and only numerical value was kept.

- **Problem-5:** IMDB and Kaggle dataset both contains some columns (date published, metascore, USA gross income, worldwide gross income, budget, reviews from critics) which seemed to be not useful for predictive model building.
  Solution: These columns were deleted from respective datasets.

- **Problem-6:** After the conversion of the date added column to date, year was extracted. The output obtained was float instead of integer.
  Solution: Further investigation was carried out, which showed, it was due to few null values in that column. The null values were filled with 0, and then converted the column to integer.

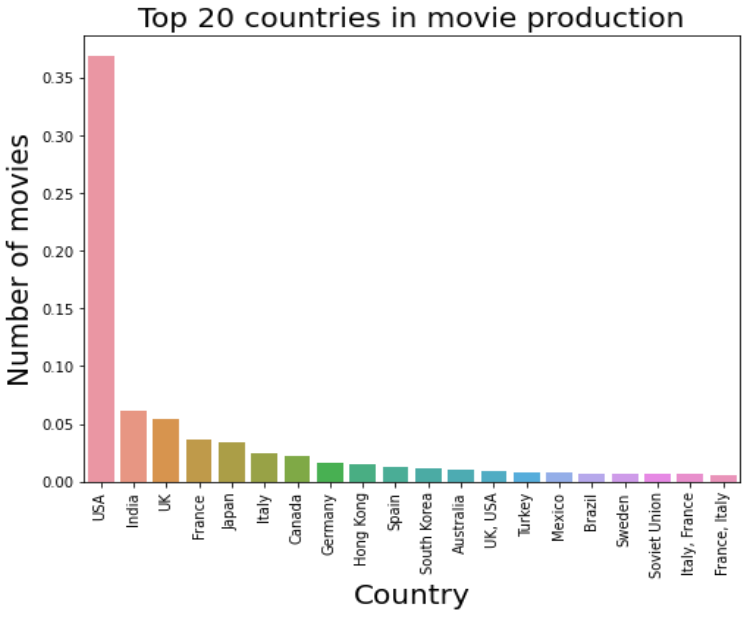
- All the rows with the null values were removed from the dataset for EDA (Exploratory Data Analysis).

The jupyter notebook for data import and data wrangling is available at:
https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netfilx/blob/main/Report/Capstone_Project_Netflix_Data_Wrangling_submission4_Report.ipynb

5. Exploratory Data Analysis (EDA):

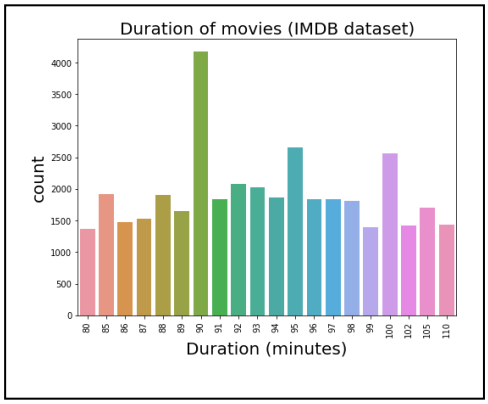
Detail investigations into each column of both datasets were carried out and following conclusions were drawn:

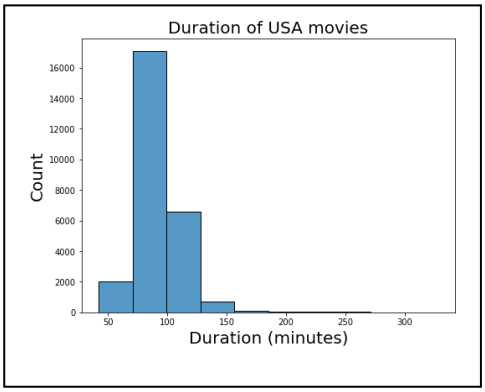
- Both datasets showed USA ranked Number One in movie production, while India ranked Second followed by United Kingdom in Third place.

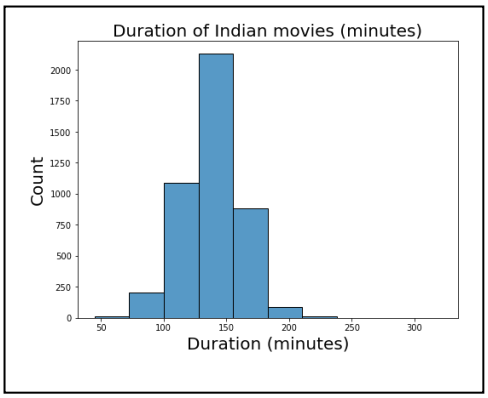


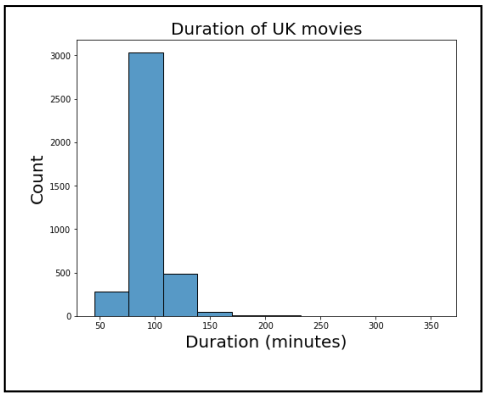
- These are the three top movie production companies:
  1.Metro-Goldwyn-Mayer
  2. Warner Bros.
  3. Columbia Pictures
  In USA, India and UK, the top movie production companies are ‘Metro-Goldwyn-May‘, ‘NH Studioz’ and ‘The Rank Organisation’ respectively.

- Both datasets show most of the movies are of duration (approximately) 90 mins. However, it varies for different countries. In USA and UK movies are of 90-100 minutes, whereas in India movies are of 120-150 minutes.









The jupyter notebook containing the EDA is available at:
https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netfilx/blob/main/Report/Netflix_EDA_submission2_for_report.ipynb

6. Feature Engineering:

As the IMDB dataset was very large as compared to Kaggle dataset, before feature engineering, I formed two new datasets out of it combining with Kaggle. Movies, which were found only in IMDB dataset were separated out and used for predictive model building, whereas movies common to both are used for testing the ML (machine Learning) model.

Feature engineering was carried out on both training and testing datasets as follows:

- Dummie variables for all the categorical columns: genre, language, actors, directors, writers, production company were created. For genre and language, the dummies for all

the unique values were carried out, because it listed few unique values (<300), however for rest of the columns, dummies for top 200 unique values in each column were obtained.

- Standard scaling was carried out for numerical columns: duration_min, votes, reviews from users to keep them in similar range. As they follow normal distribution, standard scaling was preferred among all other types of scaling.

- Few numerical columns had very high outliers, which were not mistakenly entered. However, as it might bias the ML (Machine Learning) model, hence, to avoid this, imputation was carried out, assigning 95^{th} percentile value of the respective column to the outliers.

- Finally, all the dummies columns were added to the respective main datasets and original categorical columns were dropped.

The jupyter notebook on Feature engineering is available at:

https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netflix/blob/main/Report/Netflix_data_Pre_processing_training_data_development_submission2_Report.ipynb

7. ML models

To predict the movies rating, different regression models: Simple Linear Regression, Lasso Regression, Ridge Regression, Random Forest Regressor, Gradient Boosting Regressor were developed.

- The Simple Linear regression model performance was poor, however it improved significantly when regularization was applied. Among Lasso and Ridge, the later one performed better with R2 score: 0.42.
- To have better performance, ensemble model Random Forest Regressor was developed. The highest R2 score obtained with this was 0.44.
- To achieve more higher performance, Gradient Boosting Regressor was developed. The highest R2 score obtained with this was 0.51 and this was the best model among all the models.

To improve the models' performance, PCA (Principal Component Analysis) was applied on the data and models were trained again. Though it helped Linear Regression to improve its performance and run time, it couldn't improve performance of both ensemble models.

Best model: Finally, Gradient Boosting Regressor was saved for deployment and tested on unseen data.

The two tables below list the details about the ML models.

Model	**r2_score**
Linear Regression	0.40
Lasso Regression	0.33
Ridge Regression	0.42
Random Forest Regressor	0.44
Gradient Boosting Regressor	0.51

Model	No. of Features	Hyperparameters	r2_score
Gradient Boosting Regressor	300	Learning rate: 0.1 n_estimators: 700 max_depth: 7	0.51

The Gradient Boosting Regressor model was finally applied on the unseen movies data and got r2 R2 score of 0.47.

The model optimization without and with PCA have been discussed in three separate jupyter notebooks Part-1, Part-II Part-III and available in Github in the following links:

- Part-I: https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netfilx/blob/main/Report/Netflix_data_model_development_final_Part_I_Report.ipynb
- Part-II: https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netfilx/blob/main/Report/Netflix_data_model_development_final_Part_II_Report.ipynb
- Part-III: https://github.com/damayantinaik/Springboard_Week_7_Capstone_Project_Netfilx/blob/main/Report/Netflix_data_model_development_PCAall_final_Part_III_Report.ipynb

8. Future Recommendations:

In this project, different ML predictive models have been developed to obtain the maximum possible performance with all possible hyperparameter tuning. However, the model performance can further be improved with inclusion of more features like music quality, picture quality, chorography quality, actors' ranking, users' age, etc. Hence, I'll recommend to consider these data for model building in future.

9. Acknowledgement:

I am grateful to Python developer community for providing many rich, versatile libraries to carry out all types of Data analysis and ML model building. I thank my Springboard mentor Yadunath Gupta for all his thoughtful guidance and constant encouragement to include code in advance pythonic form, which helped to me to improve myself while working on this project and complete it successfully.