Movie Recommendation System Project

By: Dennis Wainaina

Introduction

- Phoenix Incorporated wants to enter into the market of streaming services like Netflix.
- It understands the importance of recommendation systems in building a streaming service.
- It has decided to hire a company called Regex Inc a Data Science firm to build this recommendation system.
- The company is to build a system that gives 5 recommendations to the user based on what the user has watched and present their findings.

Objectives

- To build a recommendation system capable of suggesting 5 movies to users based on their past choices and popular content in the streaming service currently.

- To address the cold start problem to provide valuable recommendations to new users with limited interaction history.

- To optimise recommendation algorithms to maximise user satisfaction and platform revenue.

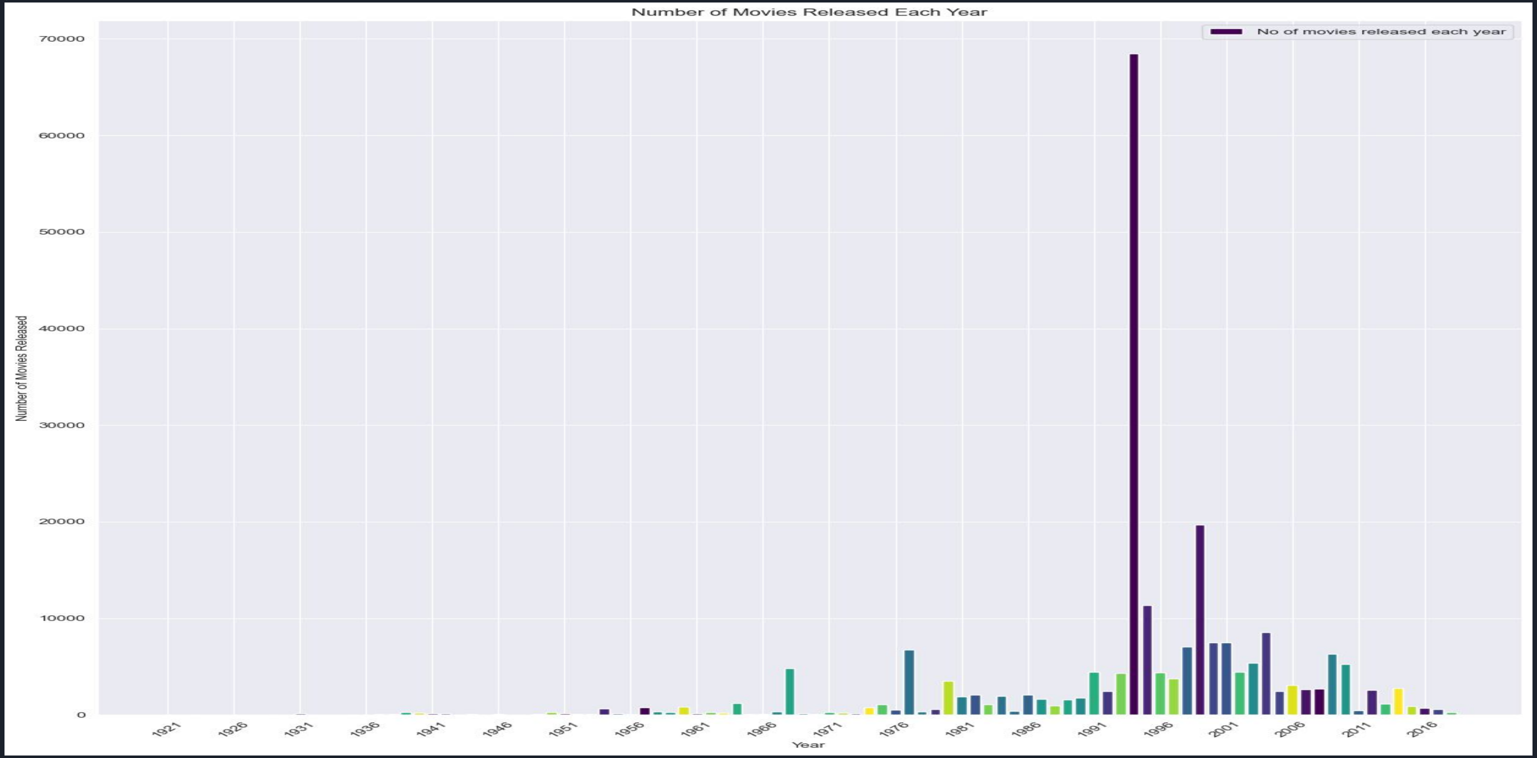
- To implement recommendation system to enhance user engagement and retention.

Methodology

- Did analysis to determine how various factors affected rating of users.
- Built model that helps users determine the movie that they will watch next based on what they have seen previously and what other like users have seen.
- Used this model to make 5 movie predictions for users based on what they and other like users have seen before.

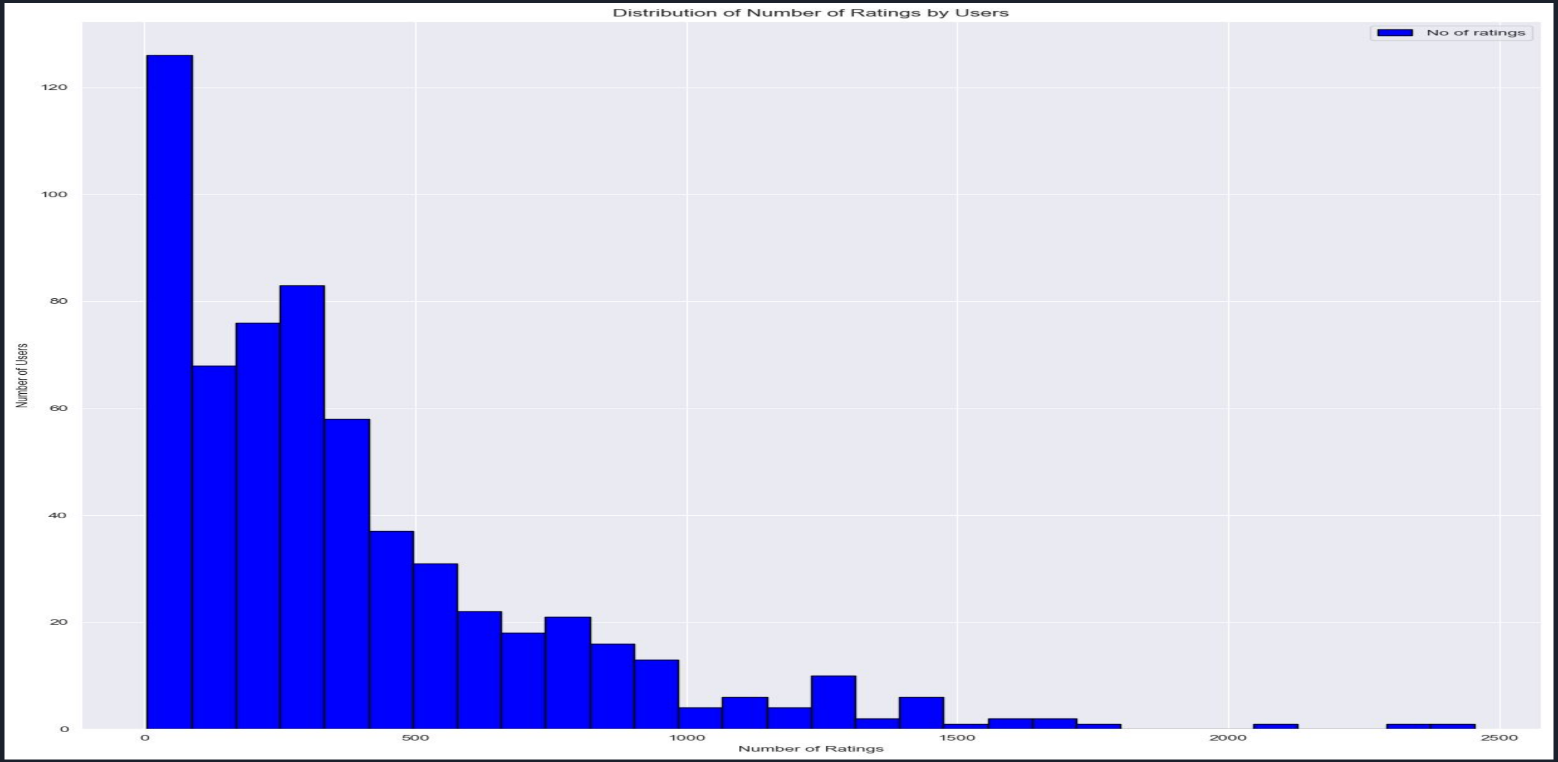
Results

- Number of movies released each year.



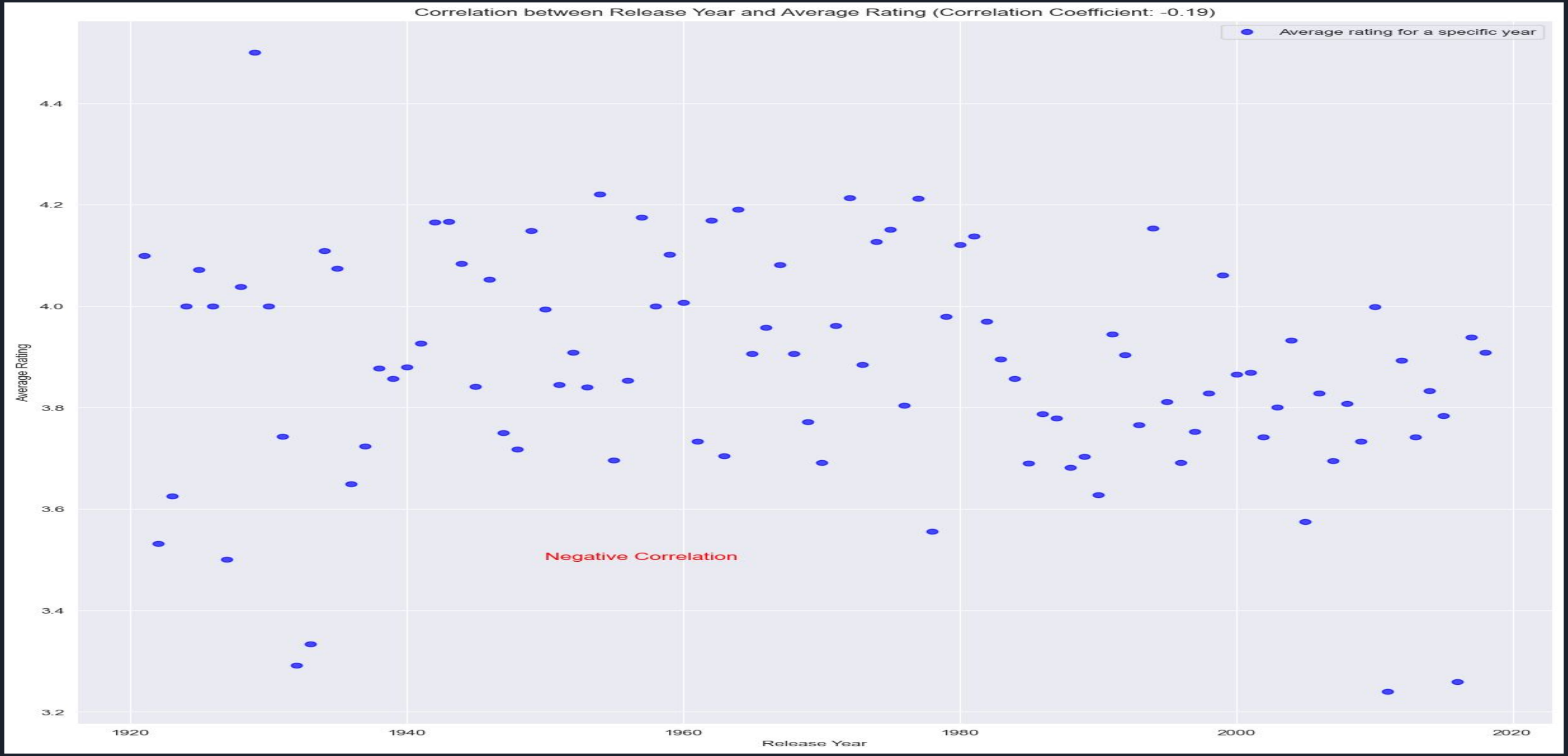
Results

- Distribution of ratings by the number of users.



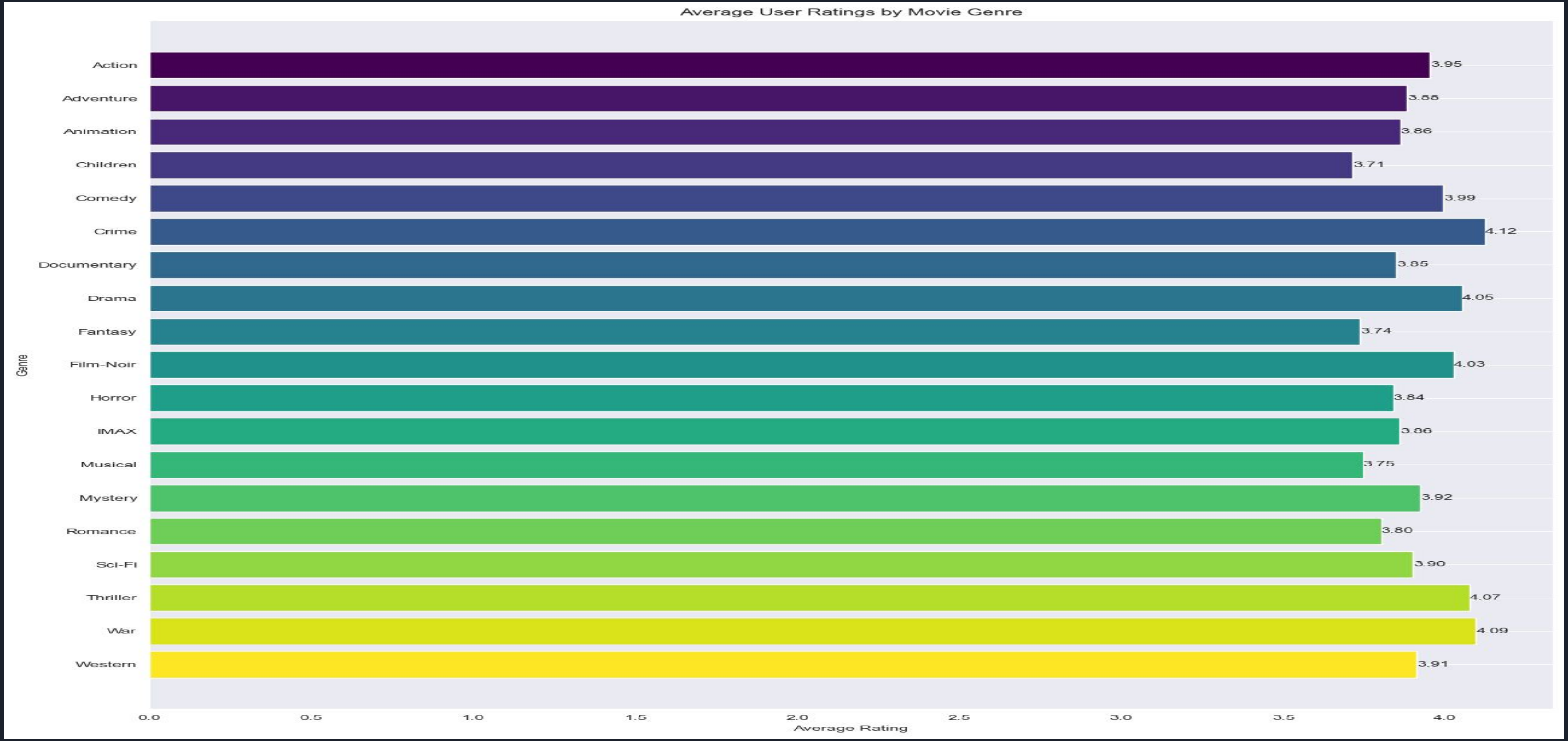
Results

- Correlation between the release year of movies and their ratings.



Results

- Average ratings of users by genre.



Model Results

1. BASE MODEL
    - Model: Singular Value Decomposition (SVD)
    - RMSE: 0.3917 (Initial Benchmark)
    - *Statement:* In this initial phase, we implemented the basic SVD model to establish a baseline for our recommendation system. This model yielded an RMSE of 0.3917, indicating room for improvement.
2. MODEL 2
    - Model: Modified SVD (SVD2) with Hyperparameter Tuning
    - RMSE: 0.7941 (Higher RMSE Despite Tuning)
    - *Statement:* Model 2 involved experimenting with a modified SVD algorithm (SVD2) and fine-tuning its hyperparameters using GridSearch. Despite efforts, this model resulted in a higher RMSE, suggesting it might not be suitable for our dataset.
3. MODEL 3
    - Model: Optimized SVD1 with Hyperparameter Tuning
    - RMSE: 0.3466 (Improved Performance)
    - *Statement:* For Model 3, we returned to the original SVD algorithm (SVD1) and optimized it by fine-tuning hyperparameters. This model achieved an improved RMSE of 0.3466, demonstrating better performance.
4. MODEL 4
    - Model: K-Nearest Neighbors (KNNBasic)
    - RMSE: 0.5570 (Computationally Intensive)
    - *Statement:* Model 4 explored the K-Nearest Neighbors (KNNBasic) algorithm. However, it proved to be computationally intensive and returned an RMSE of 0.5570, making it less feasible for large datasets.
5. MODEL 5
    - Model: Non-Negative Matrix Factorization (NMF) with Hyperparameter Tuning
    - RMSE: 0.6099 (Higher RMSE Compared to SVD1)
    - *Statement:* In Model 5, we implemented Non-Negative Matrix Factorization (NMF) and performed hyperparameter tuning. Unfortunately, this model yielded a higher RMSE of 0.6099 even after `60 mins` of hyperparameter tuning, suggesting it may not be the best fit for our data.

Conclusion

In this movie recommendation project, we designed a personalized recommendation system based on user preferences and historical data. We began with thorough exploratory data analysis, uncovering insights such as a correlation between movie release years and average ratings. Our journey started with a base model using Singular Value Decomposition (SVD), which performed well. We also explored advanced models like modified SVD and Non-Negative Matrix Factorization (NMF). Ultimately, the SVD-based model showed the most promise. Ongoing improvements, driven by user feedback and scalability, are key to enhancing the recommendation system.

Recommendations

- **Gather User Feedback**: Collect user ratings and feedback to refine recommendation algorithms.
- **Explore Algorithm Diversity**: Experiment with various recommendation algorithms and adapt to user preferences.
- **Optimize Scalability**: Ensure the system can handle growing datasets and user bases efficiently.
- **Enhance Personalization**: Improve recommendations by considering user profiles and diverse movie choices.
- **Conduct A/B Testing**: Evaluate recommendation changes to enhance user engagement.
- **Maintain Data Quality**: Monitor and clean data to prevent accuracy issues in recommendations.

Final statements

Thank you for your time if you have any questions feel free to ask.