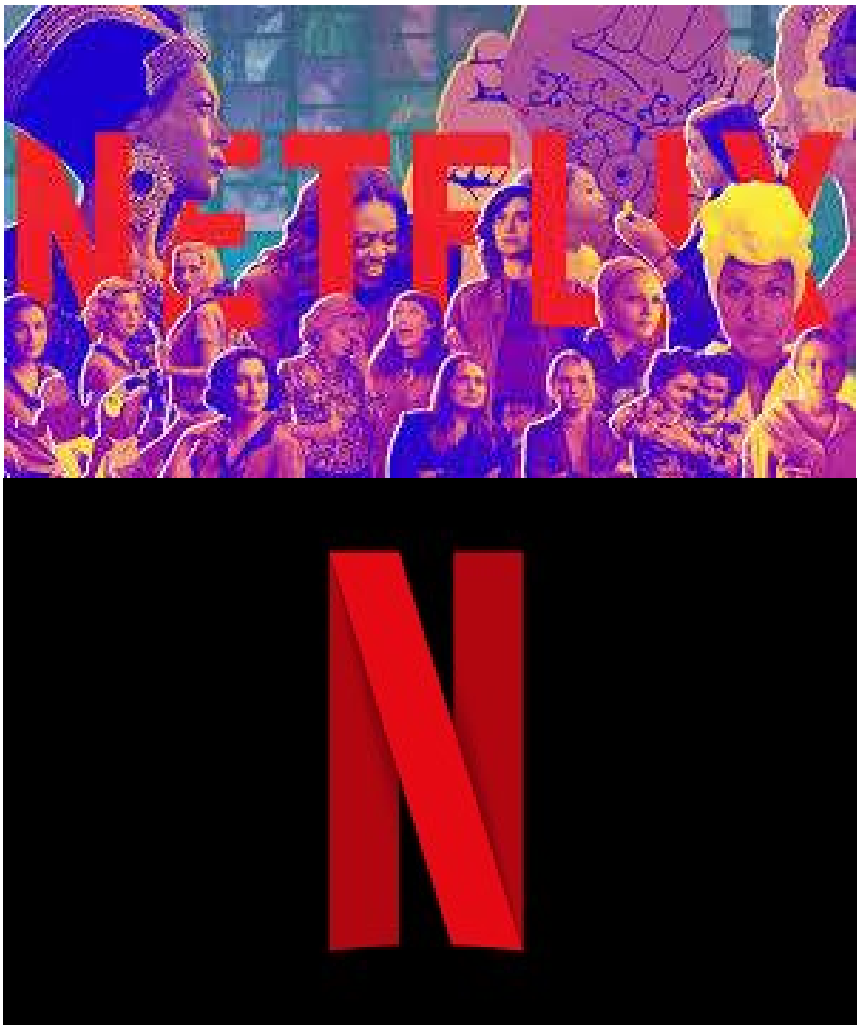


Prediction of Viewer Interest in Shows On Netflix In Particular Country

Kunal Sharma | Data Science Deliverable 3

KUNAL SHAMRA

Table of Contents

1. ABSTRACT....2
2. Business Problem....2
3. Understanding the Data....3
4. Data Mining Problem....5
5. Data Partitioning....5
6. Data Analysis....6
7. XGBoost....6
8. Surprise Baseline (Basic Algorithms)....7
9. KNN Baseline....7
10. Collaborative Filtering....8
11. Limitations....8
12. Deployment....8
13. Conclusion....9
14. References....9

1. ABSTRACT

In today's world, the growing growth of online media has resulted in the emergence of new economic activities. As a result, knowing the patron's purpose is critical to succeeding in this highly competitive climate. In this fast-paced and hypercompetitive world, understanding the desire of viewers to transact is critical. The data describes the viewing habits of different users over time, and a few data science models are used to collect knowledge about the consumers' intentions in terms of generating revenue for Netflix. We used a machine learning technique called collaborative filtering for the actual algorithm, which was designed to handle this form of user-based prediction. It considers a user's preferences as well as data from similar users to predict whether that user will like another unique object. In Python, we used the Scikit-Surprise library to accomplish this. The result indicates that viewers' goal is to forecast the future using the best fitting model, based on the efficacy of the supervised data mining models.

2. Business Problem

The business challenge here is to assist consumers in having a pleasant browsing experience so that the owners can make more money with better services. Understanding viewers' actions on the e-commerce Netflix website will aid in their decision-making process by predicting a profitable purchase and satisfied customer. The browsing session data of disti.com is used to conduct this data analysis.

This data analysis is carried out on the browsing session data of different users. It is not straightforward to anticipate as well as infer the specifics of individual customers. Based on the session info, the Data Mining Technique aids in accurately predicting revenue statistics.

Netflix generates a large amount of data to select the best suggestion for the viewer. However, due to the enormous data generated by customers from various countries using the website, there are some unique occasions during which business transactions can have a major effect on the recommendation to the audience.

The model will help to predict the recommendation to the audience that subscribers are more likely to watch from online platforms and devices to increase the user base and therefore forecast that whether any revenue will be gained.

3. Understanding the Data

This dataset contains Netflix-available movies as of. The information was gathered from Flixable, a third-party Netflix search engine. Since 2010, the number of movies available on the streaming service has decreased by over 2,000 titles, although the number of TV shows has increased. So, I will be creating models based on the Netflix Prize data set. The Netflix Prize data set will be used predict the best model for the viewer movie recommendation system. Which can be implemented with the same way on TV-Shows as well.

We cleaned the data using Python and Pandas. This involved adding movieID's to each row and removing dates. We then used Pandas and Matplotlib to explore the aspects of the dataset. Python and Pandas were used to clean the details. Each row had movieIDs added to it, as well as dates removed. We then used Pandas and Matplotlib to investigate the dataset's features.

(https://www.kaggle.com/netflix-inc/netflix-prize-data)

a. Attribute information and Data Preparation

Over 100 million reviews from 480 thousand anonymous Netflix customers were collected for over 17 thousand movie titles in the movie rating archives. The information was gathered between October 1998 and December 2005, and it reflects the current situation. All the ratings obtained during this time span have been distributed. The stars are assigned on a scale of 1 to 5 (integral). Each customer id has been replaced with a randomly assigned id to protect customer privacy. Each movie id also includes the date of each ranking, as well as the title and year of release. Each subsequent line in the file corresponds to a customer rating and the date it was given in the format:

CustomerID's, Rating, and Date are all necessary fields.

- MovieID's are numbered sequentially from 1 to 17770.

- With holes, CustomerID's range from 1 to 2649429. There are still 480189 users online.

- Ratings are on a five-star (integral) scale, with 1 being the lowest and 5 being the highest.

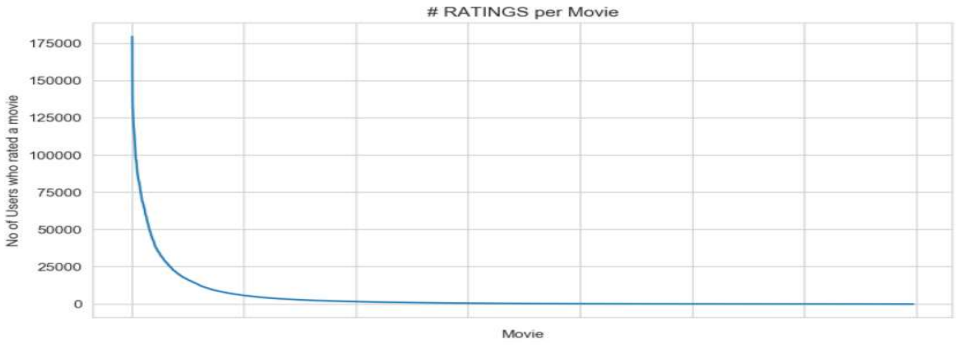
- Dates are written in the YYYY-MM-DD format.

b. Visualization of Data Variables

Checking for NaN values: No of Nan values in our dataframe.

Removing Duplicates: There are zero duplicate rating entries in the data.

Basic Statistics & Analysis



4. Data Mining Problem

Because of availability of huge data, it must be mined in an effective way using appropriate techniques to get the efficient analysis results. The problem which will be solved using data mining method is classification of Viewer's browsing data. To address the above problems, we will use classification method in data mining. The classification problem is a supervised learning method and the target variable to be predicted would be Revenue. Feature selection and modeling analysis will be done using Python on google colab.

a. Feature Selection

The most highly correlated variables are used for determining the target variable i.e., ratings which we'll focus more as we go forward.

```
Total data
-------------------------------------------------

Total no of ratings : 100480507
Total No of Users   : 480189
Total No of movies  : 17770
```

5. Data Partitioning

The data is divided randomly in 80% Training Data and 20%.

6. Data Analysis

a. Cross Validation

For analysis we will be using Machine Learning techniques on model's i.e., regression models or Surprise baseline on Knn with XGBoost to speed up the computation in a model evaluation technique. The holdout method is the simplest kind of cross validation. Here, the dataset is separated into two sets- training data set and testing data set. Then the average error i.e., RMSE is calculated across all trails (Models) is computed.

b. Supplied Test Set

The 20 % partitioned data is taken as the test data set to perform the result of trained model and calculate the RMSE values for finalizing the preferred model.

7. XGBoost

XGBoost is an open-source software library for C++, Java, Python, R, Julia, Perl, and Scala that provides a regularizing gradient boosting mechanism. It is compatible with Linux, Windows, and Mac OS X. The project aims to provide a "Scalable, Portable, and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library," according to the project description. It, as well as the distributed processing frameworks Apache Hadoop, Apache Spark, and Apache Flink, operate on a single computer.

It has recently gained a lot of recognition and success as the algorithm of choice for many winning machine learning teams.

8. Surprise Baseline (Basic Algorithms)

These are basic algorithms that do not do a lot of work but are still useful when comparing accuracies. It is an algorithm that predicts a random rating based on the training set's distribution, which is assumed to be natural.

The prediction ($\hat{r}_{ui}$) is generated from a normal distribution ($N(\hat{\mu}, \hat{\sigma}^2)$) where ($\hat{\mu}$ and $\hat{\sigma}$) are estimated from the training data using Maximum Likelihood Estimation:

$$\hat{\mu} = \frac{1}{|R_{train}|} \sum_{r_{ui} \in R_{train}} r_{ui}$$

$$\hat{\sigma} = \sqrt{\sum_{r_{ui} \in R_{train}} \frac{(r_{ui} - \hat{\mu})^2}{|R_{train}|}}$$

9. KNN Baseline

The k-nearest neighbors algorithm (k-NN) is a non-parametric classification system invented by Evelyn Fix and Joseph Hodges in 1951 and later extended by Thomas Cover in statistics. It is employed in the classification and regression of data. The input in both cases is the k closest training examples in the data set.

For KNN baseline please refer below link.

https://surprise.readthedocs.io/en/stable/knn_inspired.html#surprise.prediction_algorithms.knns.KNNBaseline

We use a shrunk Pearson-baseline correlation coefficient based on Pearson Baseline similarity (we use base line projections instead of mean user/item ratings).

Predicted rating (based on Item-Item similarity):

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_u^k(i)} \text{sim}(i,j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in N_u^k(j)} \text{sim}(i,j)}$$

10. Collaborative Filtering

The relationship between users and objects is the subject of collaborative-filtering systems. The similarity of two things is measured by the similarity of their scores by users who have rated both. A utility matrix is often used to determine collaborative processes. The recommender model's goal is to figure out a feature that predicts the utility of fit or similarity for each consumer. The utility matrix is usually sparse, large, and contains values that have been removed.

11. Limitations

Our strategy and tools, of course, have their limits. First, we agreed that the movie's release date and review were not important enough to hold. While the original winning team accounted for these in their algorithm, we believe it is fair to say that the release and evaluation dates have no impact on the user's viewing experience and subsequent ranking. While the model does a decent job of predicting tastes of existing database users, it fails miserably when it comes to "new" user submissions, always recommending the same top movies regardless of feedback. The amount of time and memory required for these calculations does not help. We intended to compare results using both Surprise and Fast.ai, but the latter took too long to measure (>1 day), so we were unable to do so. We had to ignore data for 80% of the movies in the Surprise dataset just to get the model to compute reasonably; even so, the dump file was nearly 900 MB, rendering it useless for deployment on the web (due to Heroku and Git's limitations).

12. Deployment

The model's deployment is the final step in the data mining process. The RMSE values for the models are listed below.

knn_bsl_u	1.072649932414952
knn_bsl_m	1.072758832653683
bsl_algo	1.0729400309431507
svd	1.0729585081962245
svdpp	1.0729704920881107
xgb_all_models	1.0750762820322688
first_algo	1.076373581778953
xgb_knn_bsl	1.0766603238200672
xgb_bsl	1.0766668577320093
xgb_final	1.1032957214417232

The results show that the knn_bsl_u is an excellent model for predicting the future, with an RMSE of 1.072649932414952, which is greater than any other model. The trained model can be used to interpret the test data to assess the model's effectiveness.

13. Conclusion

The aim of this project is to use Machine Learning to develop a movie recommendation engine that have minimum RMSE. My work is based on a late-2000s Netflix competition with a $1 million grand prize. The browsing trend of the visitor rating to e-commerce websites can be used to assess the online viewer's purpose. With an RMSE of 1.072649932414952, the knn_bsl_u Supervised model helps us predict the outcome using the test dataset, which is the lowest among all the models we tested. Hence, in any region with better rating prediction, the better movies can be tagged in the browser i.e.,

Better Engine Recommendation-> Better User experience-> and hence forth which will help in increasing the revenue itself for Netflix in any region of Country.

14. References

- https://www.netflixprize.com/rules.html
- https://www.kaggle.com/netflix-inc/netflix-prize-data
- surprise library: http://surpriselib.com/ (we use many models from this library)
- surprise library doc: http://surprise.readthedocs.io/en/stable/getting_started.html (we use many models from this library)
- https://github.com/manelmahroug/netflix-movie-recommendation
- installing surprise: https://github.com/NicolasHug/Surprise#installation
- Research paper: http://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/a1-koren.pdf (most of our work was inspired by this paper)
- chrome-extension://ieepebpjnkhaiioojkepfniodjmjjihl/data/pdf.js/web/viewer.html?file=https%3A%2F%2Fcran.r-project.org%2Fweb%2Fpackages%2Frecommenderlab%2Fvignettes%2Frecommenderlab.pdf