MOVIES & TV PREDICTIONS

John Jun

Abhay Kumar

Aditya Bhatnagar

11.30.2019

CSE 158 - Recommender Systems & Web Mining

◎ THE AMAZON MOVIES & TV DATASET

Context

Amazon is an eCommerce platform that sells a wide range of products to its users. There are various types of products, therefore being able to suggest users the products they are interested in is crucial to their business. A recommender system allows showcasing the “right” products to the user, i.e. recommending the customers the products they are most likely to purchase. Building a recommender system involves exploring the data collected on user activity, and applying suitable techniques to predict whether a user will purchase a product or not.

We aim to study Amazon’s ‘Movies & TV’ category of products, and will predict whether the user will purchase a given product or not.

Data Exploration

The ‘Amazon Movies & TV Dataset’ is a subset of the ‘Amazon Product Reviews’ dataset. It is a large volume of the reviews customers have written on the Amazon website.

We’re dealing with the Movies and TV products & the reviews recorded by Amazon received from over 20 million users.

The Movies & TV dataset has 8,765,568

reviews, with the following format:

```
[['0001527665', 'A3478QRKQDOPQ2',
'5.0'],

 ['0001527665', 'A2VHSG6TZHU1OB',
'5.0'],

 ['0001527665', 'A23EJWOW1TLENE',
'5.0'],

 ['0001527665', 'A1KM9FNEJ8Q171',
'5.0'],
  ....
  ....]
```

The first column represents the **ID of the movie/TV product** reviewed.
The second represents the **ID of the user** who rated this particular product.
The last column represents the **numeric rating** assigned to the product by the user.

Basic Statistics:

Dataset Size	8,765,567
Number of Users	3826085
Number of Movie/TV Products	182032
Average Rating (excludes non rated items)	4.2330104829499335

Properties

(1) all products in the dataset have each been reviewed at least 5 times and all users in the dataset have each reviewed at least 5 products (it is called 5-core)

(2) The dataset contains a total of 8765567 reviews, sorted by productID. Each review has a userID, productID, and a rating from 1 through 5, given by the user.

Interesting Findings

Popular Rating Value

The majority of the customers rate the movie 5/5. Based on a naive categorization where the rating less than 3 implies unsatisfactory purchase and the rating greater than 3 implies satisfactory purchase, the dataset clearly contains significantly more of the satisfactory purchase.

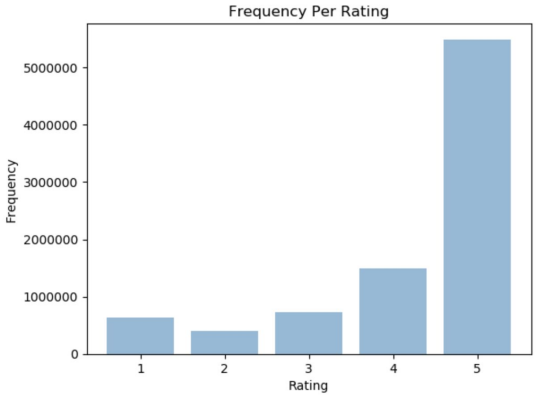
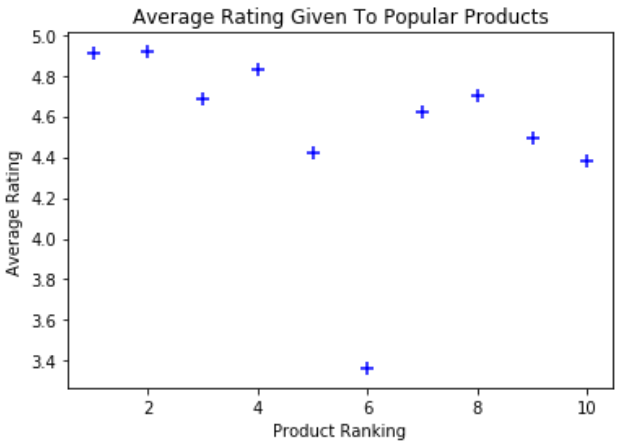
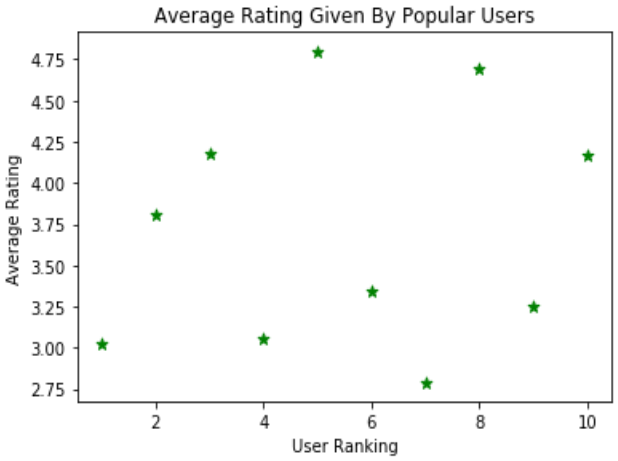


Figure 1 : Frequency of each rating in dataset

Top 10 Most Popular Products And Their Average Ratings



Top 10 Active Raters And The Average Ratings Given By Them



◎ PREDICTIVE TASK

Chosen Task

The predictive task that we've identified is to determine whether or not a user will purchase the TV show/movie or not.

Additional Data Processing

In order to make a binary prediction of the label not provided directly in the dataset, we label the reviews already present with a positive label of 1 and then employ random sampling method in order to produce userID and productID pairs with negative label of 0.

In addition to incorporating a balanced dataset that we mentioned above, we also shuffled the data so that each training, validation and test dataset would have a similar distribution of reviews per product. We did this as we found that the data was ordered by the Product ID, and thus we were required to even out the distribution.

Model Evaluation & Validity

Evaluation of the model will be performed using the accuracy of the prediction as the metric. In addition, we incorporate the Balanced Error Rate (BER), which computes the average of the sensitivity and the specificity of the predictions, and the F1 score, in order to summarize both precision and recall.

In addition, in order to assess the validity of predictions, we use validation dataset for threshold manipulation and we use test dataset for final assessment of the prediction. The dataset, therefore, is split 60% for training, 20% for validation, and 20% for test dataset.

Relevant Baseline Models

The most trivial baseline would be a random predictor. A random predictor would make prediction simply based on a random number generator (outputs 0 or 1), without incorporating any knowledge of the dataset. The expected accuracy for this baseline model is approximately 0.5.

We implemented another relevant baseline based on the popularity of a movie using a manual cut-off of 50 percentile. Intuitively, the more popular the TV series/movie, the more likely that a given user has watched the TV series/movie. It is expected that the accuracy for this baseline model to be higher than the above model using the random number generator.

◎ THE MODEL

Chosen Model

We chose Support Vector Machine (SVM) [Model 4] for the predictive task. SVM is effective in high dimensional spaces and can be versatile as different kernel functions are available and can be specified to meet the need. However, when the number of features is greater than the number of samples, it is important to caution against overfitting. The specific reasons for having chosen this model over other alternative models is explained in the sections to follow.

Features Engineered

We extracted the following features for both SVM and Logistic Regression:

(1) The average rating of a product
(2) The number of reviews by a user
(3) The number of reviews for a product.

As the dataset contained only three types of information, we were limited to use only a small number of variables for feature engineering. Using these three variables, a feature matrix was created for (1) training (2) validation and (3) test dataset.

Optimization & Issues

SVM using linear kernel produced a high accuracy rate of 0.96016. In an attempt

to explain for such high accuracy, the following 3D plot was generated. Each of the three axes represent the each of the features used for feature engineering.

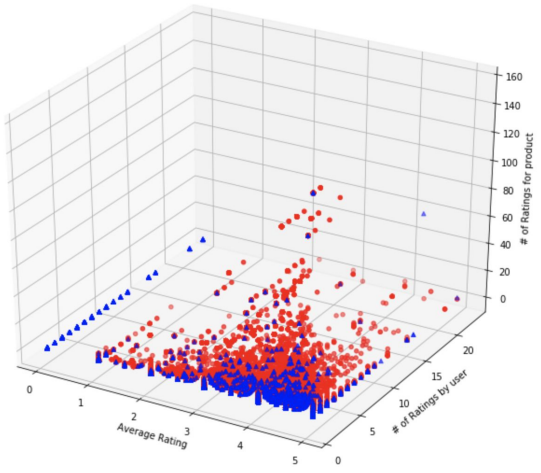


Figure 2 : 3D Plot of Features Extracted from Training Dataset

Visually, it is observed that the blue data points (label 0) tend to be concentrated in the region where the number of ratings by user is relatively low. Similarly, the red data points tend to be more loosely scattered in regions with relatively higher number of (1) ratings by user and (2) ratings for product.

From the plot, it can be observed that the linear kernel may not produce the best hyperplane to separate the labels with greatest margin. Therefore, the rbf kernel was used alternatively to make predictions based on a non-linear 3d-hyperplane. The accuracy increased to 0.98012 upon changing the kernel type.

While the SVM model with rbf kernel did produce high accuracy, the time required to train the model was significantly longer than the time required to train the logistic regression model with the same features. Provided that the logistic regression still produced a competitive accuracy of 0.94388, the logistic regression model could be an option when the computational power is an issue.

[Model 4] SVM performance:

F1 Score	0.9818938769097835
BER	0.01844
Accuracy	0.98156

Other Models

Models considered include popularity-based model, logistic regression, and a collaborative filtering model incorporating Jaccard similarity and average rating information.

1) Model that uses popularity metric uses the number of ratings per product as a measure to evaluate the product popularity. This model uses the algorithm similar to the baseline model for Assignment 1. Therefore, this model predicts 1 if the product belongs to the top 50% percentile and 0 if the product belongs to the low 50% percentile.

[Model 1] Popularity-based model performance:

F1 Score	0.78207569
BER	0.27876
Accuracy	0.72124

2) Model that uses collaborative filtering (Jaccard similarity between products p and p' and also between users u and u') and other features such as popularity and average rating information. Thresholds for the features were generated manually using a training set and the same thresholds were used for the validation and the test set. Both the training and test sets consisted of half positive and half negative samples. One key assumption made was that the average rating given to a product, in conjunction with the number of ratings given to it (a reasonably popular product with a high average ratings is likely to be purchased), makes it more likely to be purchased.

[Model 2] Collaborative filtering model performance:

F1 Score	0.9966512517939722
BER	0.00336
Accuracy	0.99664

3) **Logistic Regression**. It is a Machine Learning **classification** algorithm that is used to predict the probability of a categorical dependent variable. As our predictive task is to make binary predictions (1 = purchased / 0 = not purchased), we experimented with the logistic regression model. In addition, the features used for this model are independent variables, including the popularity of a product, the number of reviews posted by a user, and the average rating of the product.

[Model 3] Logistic Regression Performance:

F1 Score	0.9479391802221969
BER	0.05492
Accuracy	0.94508

Unsuccessful Attempts

All models beat the baseline model (using random number generator). Some of the unsuccessful attempts include the following:

(1) Due to the lack of understanding of the dataset (sorted by the productID), the popularity-based model initially failed to improve accuracy compared to the trivial baseline model using random number generator. Due to such ordering of reviews, the overlap between the products reviewed in the training dataset and the

products reviewed in the validation dataset was minimal. In order to account for such characteristics in the dataset, the entire dataset was randomly shuffled before splitting into training, validation, and test dataset.

(2) Logistic regression was worse than SVMs in terms of accuracy. Logistic regression model was 94.508% accurate while SVM (rbf kernel) was 98.156% on test sets. We attempted to incorporate Jaccard similarity between users u and u' as an additional feature in the model. The addition of these features did improve our accuracy; however, we could not devise a logical justification for its inclusion. Features such as average rating and popularity can be compared across all entries in a feature matrix as they are independent of each other. However, Jaccard only gives a value for the similarity between two particular users, so it is only relevant when comparing two users to each other.

Model Comparison (Strengths & Weaknesses)

[Model 1] Popularity-based model:

F1 Score	0.78207569
BER	0.27876
Accuracy	0.72124

Strengths: No training is required and easy to implement. Able to capture a general trend with a single feature.

Weaknesses: Does not account for factors other than the popularity and fails to consider outliers (i.e. does not account for personal preference). Hence, the accuracy and other scores are lower than those from other models.

[Model 2] Collaborative filtering model performance:

F1 Score	0.9966512517939722
BER	0.00336
Accuracy	0.99664

Strengths: Collaborative Filtering worked very well for this dataset since there was high similarity between users and products. This meant that similar users would be highly likely to purchase the same products. The high similarity can possibly be explained by the fact that we used the 5-core subset of the original dataset.

Weaknesses:
We need to find optimal thresholds by trial-and-error. It may not work as well for the original, larger dataset, which would not follow the 5-core restriction as did the dataset we used.

[Model 3] Logistic Regression Performance:

F1 Score	0.9479391802221969
BER	0.05492
Accuracy	0.94508

Strengths: Logistic regression makes it simpler to build a model as we don't have to find manual thresholds by trial-and-error. It is also less expensive to train than SVM and it performs well probabilistically

Weaknesses: Jaccard similarity didn't work well for Logistic regression as this model works better with features that aren't related to each other. Jaccard similarity is inherently a relation-oriented feature as it finds the relationship between similar users or products.
This model does not optimise number of errors.

[Model 4] SVM performance:

F1 Score	0.9818938769097835
BER	0.01844
Accuracy	0.98156

Strengths: SVM is very efficient in high dimensional spaces. It has lower BER than Logistic Regression as it optimizes the classification error rather than the likelihood (non-probabilistic). It would work well in general as there is no need to set manual thresholds for the features.

Weaknesses:
Most expensive model to train. Jaccard similarity does not work well with this model for the same reason as it did not for Logistic Regression.

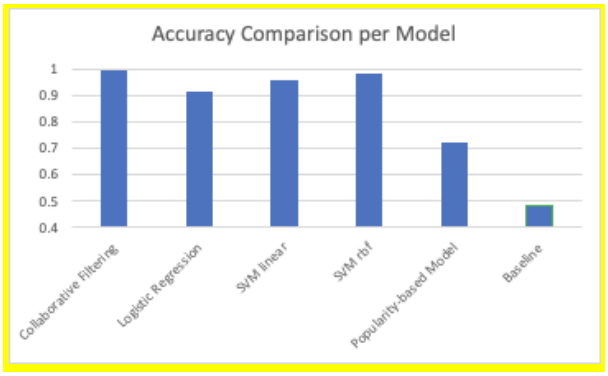


Figure 3: Accuracy Comparison between Models

◎ MOVIES & TV DATA LITERATURE

We're using Amazon Purchase Prediction dataset compiled by Prof. Julian McAuley. It's a huge crawl of product reviews from Amazon. The whole dataset contains about 82.83 million unique reviews from around 20 million users. For this project, we picked only the Movies & TV product category from the entire set.

We explored kaggle to see if there were any similar datasets people were involved with. We found a dataset provided by Netflix to improve the performance of their recommendation engine. Netflix launched an online competition called the Netflix Prize which allowed anyone to compete to improve their recommendation accuracy. This dataset is structurally similar to the one we're investigating for this report, and the competition was held for improving the accuracy of their recommendation engine. We looked at possible predictive tasks that people have worked on with the Netflix prize, and we found some people used the dataset to predict if a movie would be watched by the user or not. Since the Amazon dataset is structurally quite similar, we decided to focus on the predictive task of determining where a user will purchase a movie/TV show product or not.

Current State-of-the-Art Methods

In order to understand the data we have, we explored how Amazon actually built their own recommendation engine. We came across this article describing Amazon's approach. Amazon uses neural networks for their recommender system. Also, in order to process millions of products and their reviews in real time, they built a system called DSSTNE (Deep Scalable Sparse Tensor Neural Engine" which utilizes sparse matrices to calculate results efficiently. These techniques seemed quite intriguing, however we realized that we weren't using the entire dataset, but rather a small subset of it. Instead of focusing on multiple categories of products, we decided to explore only the Movies & TV category. Therefore, we didn't find the need to use the sparse matrices technique (DSSTNE) to make our work more efficient, as our data processing wasn't too time consuming.

We realized that what we had implemented (SVM, Logistic Regression, etc) for our recommendation engine was sufficient enough for our predictive task as our accuracy was high enough (98.156%) without implementing the state-of-the-art techniques that we found during our research.

◎ RESULTS & CONCLUSION

Best Model (= SVM)

SVM works best for our predictive task. It has a very high accuracy ~0.9815. Although, SVM does not perform as well as [Model 2] for this dataset but is likely to be more accurate in general, especially for datasets in which users and products don't have high similarity.

Features that worked well

- The average rating of a product
- The number of reviews by a user
- The number of reviews for a product
- Jaccard Similarity (only for model 2)

Features that didn't work well

For Logistic Regression And SVM:

Jaccard similarity did produce high accuracy when applied to logistic regression and SVM. However, as mentioned above in the "Unsuccessful Attempts" section, there is no proper logical justification to use Jaccard similarity. Instead, Jaccard similarity works better with features that are independent of each other. It is inherently a relation-oriented feature as it finds the relationship between similar users or products.

Why SVM worked better than other models

SVM vs Logistic Regression:

Logistic regression maximizes probability, but does not optimize the number of mistakes. This is why SVM outperforms LR in this dataset (well shuffled) as it optimizes the classification error rather than the likelihood. This can be explained by the lower BER using SVM compared to LR

SVM vs Collaborative Filtering:

SVM does not perform as well as [Model 2] for this dataset but is likely to be more accurate in general, especially for datasets in which users and products don't have high similarity. Note that the dataset used for the current task is 5-core subset of the original data.

SVM vs Popularity-based model:

The popularity-based model fails to account for personal user preferences as it simply considers the popularity of the product. On the other hand, SVM accounts for a level of ambiguosity in the dataset by considering two more additional features. In addition, as SVM calculates, in the back-end, the function for the hyperplane with highest margins from the data points and, therefore, is a more reliable and generalizable for different datasets as well.