Movie Recommendation System

Yashvi Pipaliya (AU1841092)
Information and Communication Technology-BTech. 3rd Year
Ahmedabad University
Ahmedabad. India
yashvi.p@ahduni.edu.in

Kesha Bagadia (AU1841011)
Information and Communication Technology-BTech. 3rd Year
Ahmedabad University
Ahmedabad, India
kesha.b@ahduni.edu.in

Yashvi Gandhi (AU1841033)
Information and Communication Technology-BTech. 3rd Year
Ahmedabad University
Ahmedabad, India
gandhi.p@ahduni.edu.in

Manal Shah (AU1841026)
Information and Communication Technology-BTech. 3rd Year
Ahmedabad University
Ahmedabad, India
manal.s@ahduni.edu.in

***Abstract*—Movie recommendation system helps movie enthusiasts by suggesting movies without them having to go through a process of choosing from a large set of movies thus, time consuming and confusing. The project focuses on finding a model using content based recommendation approach with the help of different algorithms.**

***Index Terms*—Data preprocessing**
Exploratory Data Analysis
Machine learning techniques
Random Forest Regressor
Feature importance
Permutation Importances
Drop Column Feature Importances

I. Introduction

A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularity of the movies. Movie recommendation systems usually predict what movies a user will like based on the attributes present in previously liked movies. Such recommendation systems are beneficial for organizations that collect data from large amount of customers,and wish to effectively provide the best suggestions possible

II. Literature survey

Movies are enjoyed by everyone, across age, gender, race, color and geography, it's a medium that connects us. But our choices, even so, remain different from others. That is where data scientists come in. They extract behavioral patterns of the audience and the movies to give required results[1]

A recommendation system suggests users a number of resources which can be anything like songs or books, with the basis of a data set. In our case, we will be working for movies. On the input of preference, it will give recommendations that the user is likely to enjoy.[2]

Amongst different types, a content based recommendation system mainly works with data provided by the user, extracted from a source, or inputted on some interface. And generally, based on the data, a profile is generated, which is then used to make suggestions to the user. With more inputs, it gets more accurate.[3]

The method to model this approach is the Vector Space Model (VSM). As the algorithm basically gives recommendations for products that are similar to the preferences of the user, it uses the computation of similarity.[1] This similarity of the movies is derived from its description, applying the concept of TF-IDF, which is Term Frequency-Inverse Document Frequency.[4]

TF is the frequency of a word in a document and IDF is the inverse of the document frequency, very much as the name suggests. TF-IDF operates in a manner such that the weighting negates the effect of high frequency words in determining the importance of an item. For example, if one was the to look up "the decline of feudalism" on Google, it is certain that "the" will occur more frequently than "feudalism", but the relative importance of the latter will be higher than the former from a search query point of view.[1]

Vectors generated from the above concepts are used to compute similarity. One way to do this is cosine similarity. It basically measures the angle of cosine between the two objects and compares them on a normalized scale. This is done by calculating the dot product of the two identities. And lesser the angle between the two vectors, more is the similarity. [4]

To make these computations, a number of features are selected from the given data. This is done using different methodologies of feature importance. Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction. [5]

But content based recommenders have their own limitations. They are not good at capturing inter-dependencies or complex behaviors. But they can still function pretty well on optimisation. [3]

III. Implementation

We first went for data gathering where we collected and merged different data-sets to obtain the required parameters

for Recommendation and Prediction. We then went for data preprocessing and cleaning where we detected and corrected the corrupt or inaccurate records from the database that may negatively impact a predictive model. Followed by Exploratory Data Analysis to identify obvious errors, and get a better idea of the patterns within the data, detected outliers or anomalous events and found interesting relations among the variables. While performing EDA we observed various trend among different parameters as shown here which gave us an more detail understanding of our finalised data set. We then split the whole dataset into test and training data as per the 20/80 ratio. We have used Label encoder to convert the string parameter to integer as most of our data was in string format. Next we used Random Forest method provided by python which is an ensemble classifier that uses multiple models of several DTs to obtain a better prediction performance. It creates many classification trees and a bootstrap sample technique is used to train each tree from the set of training data. Using the Random Forest we obtain the accuracy of 0.901638 using the rf.score which basically is R^2 value i.e is a statistic that will give some information about the goodness of fit of a model. Lastly, we have obtained feature importance of RandomForestRegressor, Permutation feature importance and Drop Column feature importance to find ideal parameters. After compering and the results obtained by these feature impotence's we removed the parameters with least importance and concluded that following the Drop Column feature importance we can increase the accuracy by 0.003 which makes the accuracy as 0.9045671.

A. Results

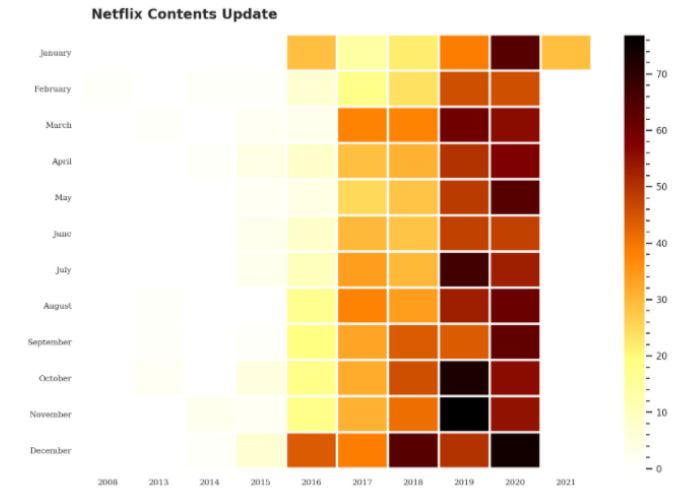


Fig. 1. Netflix Content Update.

IV. Conclusion

We have completed Exploratory Data Analysis(EDA) to identify obvious errors, as well as for a better understanding of patterns within the data, detect outliers or anomalous events, find interesting relations among the variables. We have obtained Feature importance of Random Forest Regressor, Permutation feature importance and Drop Column feature importance to find ideal parameters. We will complete the recommendation system and optimise it to the best of our ability.

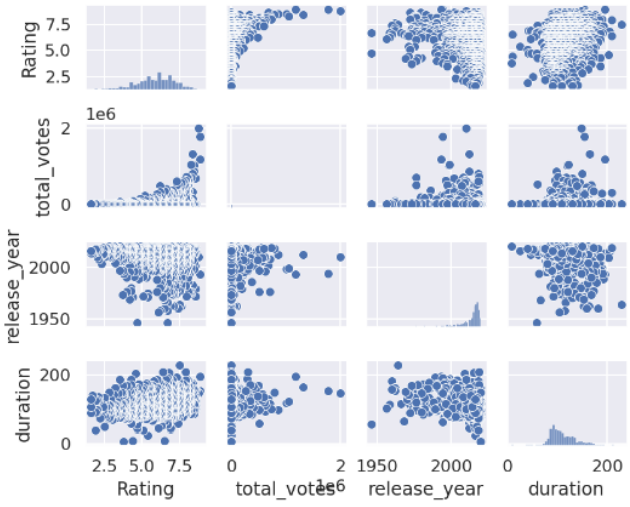
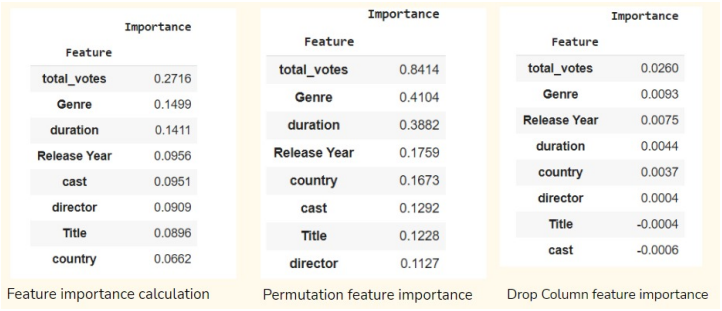


Fig. 2. sns plot.



Feature importance calculation

Feature	Importance
total_votes	0.2716
Genre	0.1499
duration	0.1411
Release Year	0.0956
cast	0.0951
director	0.0909
Title	0.0896
country	0.0662

Permutation feature importance

Feature	Importance
total_votes	0.8414
Genre	0.4104
duration	0.3882
Release Year	0.1759
country	0.1673
cast	0.1292
Title	0.1228
director	0.1127

Drop Column feature importance

Feature	Importance
total_votes	0.0260
Genre	0.0093
Release Year	0.0075
duration	0.0044
country	0.0037
director	0.0004
Title	-0.0004
cast	-0.0006

Fig. 3. Outputs of conducted feature importance methods.

V. References

[1] Das, S. (2020, November 24). Create Your Own Movie Movie Recommendation System. Analytics Vidhya.

[2] Reddy, S. (2019). Content-Based Movie Recommendation System Using Genre Correlation. SpringerLink.

[3] Das, S. (2015, September 24). Beginners Guide to learn about Content Based Recommender Engines. Analytics Vidhya.

[4] Movie Recommendation System using Cosine Similarity and KNN. (2020). International Journal of Engineering and Advanced Technology, 9(5), 556–559. https://doi.org/10.35940/ijeat.e9666.069520

[5] Brownlee, J. (2020, August 20). How to Calculate Feature Importance With Python. Machine Learning Mastery. https://machinelearningmastery.com/calculate-feature-importance-with-python/