

# Movie Recommendation System using Naive Bayes Algorithm with Collaborative Filtering

Anchal Dubey<sup>1</sup>, Raju Ranjan<sup>2</sup>

Galgotias University, Greater Noida, Uttar Pradesh, India

**Abstract:** *In recent years, the movie industry is getting more and more prosperous. There are hundreds of movies released every year. However, it is difficult to notice the releasing of every movie, not to mention actually seeing it. Therefore, movie recommendation system has become more and more popular as a research topic. Until today so many numbers of recommendation algorithms have been proposed, where collaborative filtering and content-based filtering are the two most famous and adopted recommendation techniques. Collaborative filtering recommendation systems recommend items by identifying other users with similar taste and use their opinions for recommendation whereas content-based recommendation systems recommend items based on the content information of the items. However, these systems suffer from scalability, data sparsity, overspecialization and cold-start problems resulting in poor quality recommendations and reduced coverage. Hybrid recommendation system combines the individual system to avoid certain mentioned limitations of these systems. In this project, we propose a Movie Recommendation System by combining the Naive Bayes Algorithm with Collaborative filtering.*

**Keywords:** Sentiment Analysis; Collaborative Filtering; Datasets; Android

## 1. Introduction

Recommendation System is a subclass of information filtering system that seeks to predict the 'rating' or 'preference' that user would give to an item. In this project, we have combined Naive Bayes Algorithm with Collaborative filtering for predicting which movie the user will like the best.

Collaborative filtering algorithm usually works by searching a large group of people and finds a smaller set with tastes similar to the user. The algorithm looks at other patterns and tries to form a pattern out of it and combine to create a ranked list of suggestions. Finally, it shows the suggestion to the user. Sentiment analysis is a field dedicated to extracting subjective emotions and feelings from text. One common use of sentiment analysis is to figure out if a text expresses negative or positive feelings. Written reviews are great datasets for doing sentiment analysis, because they often come with a score that can be used to train an algorithm.

New collaborative filtering algorithm are being designed to provide better performance of the algorithm to provide users with more accurate recommendations and consider the basics of naïve Bayesian algorithm for similarity.

The recommender system represents an intelligent system, which identifies the user category based on user information analysis and user interest analysis. Once such information is obtained, the analysis is performed in second stage to obtain the similarity group respective to available products and services.

## 2. Literature Survey

Recommender system has been so extensively used these days that it has become a preferable choice for researchers. First paper on recommender system was published in year 1998. Since then a significant number of papers had been published. Different factors have been explained to increase

the reliability of recommender system. In year 2005 John O'Donovan, Barry Smyth [1] have taken trust as the percentage of correct predictions that, a profile has made in general (profile-level trust) or with respect to a particular item (item-level trust). Authors have described a number of ways in which these different types of trust values might be incorporated into a standard collaborative filtering algorithm and evaluated each against a tried-and-test benchmark approach and on a standard data set. This decreases the prediction error by 22%.

In another work by [2], an approach is presented that is not limited to any specific recommendation algorithm. The intuition behind this approach comes from the assumption that multi criteria ratings represent user preferences for different components of an item, such as story, acting, direction, and visuals in the case of movies. Therefore, an item's overall rating is not just another rating that is independent of others; rather, it serves as some aggregation function  $f$  of the item's multi criteria ratings. In other words, this approach assumes that the overall rating has a certain relationship with the multi criteria ratings. For instance, in a movie recommendation application, the story criterion rating might have a very high priority—that is, movies with high story ratings are well liked overall by some users, regardless of other criteria ratings. Therefore, if a system predicts that a movie's story rating will be high, it must also predict that the overall rating will be high in order to be accurate.

In another work [3], the major discussion took place on Novel Bayes and Hidden Bayes. How they can provide an upper hand over what is being used now.

In another work [4], an approach is presented that indicates towards improvements that can be made to Naïve Bayes approach for text Classification.

In another work [5], discussions over how they largely fail to capture sentiment information and presented a model that uses a mix of unsupervised and supervised techniques to

learn word vectors capturing semantic term–document information as well as rich sentiment content.

In another work [6], an approach is presented about how the classification into classes or tags defined according to the problem statement if exceeded or exploited can affect the efficiency of the classification being done.

### 3. The Algorithm

Naive Bayes is a popular algorithm for classifying text. Although it is simple, it often performs as well as much more complicated solutions. A naive Bayes' classifier works by figuring out the probability of different attributes of the data being associated with a certain class. This is based on Bayes' theorem.

The Theorem is  $P(A|B)=P(B|A), P(A)P(B)P(A|B)=P(B|A), P(A)P(B)$ .

This states "the probability of A given that B is true equals the probability of B given that A is true times the probability of A being true, divided by the probability of B being true."

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes).

### 4. Data Flow of the Application

The data flow in this application is between three objects, which are client, backend, and our python algorithm. Once, the user submits the review for a particular movie, it gets uploaded on Firebase which is a real time cloud service, from there, a stream handler which is working continuously fetches the data and passes the review to the model through a destined function under it. From the model, there comes a response that whether the review is positive or negative. Now, that response is sent back to the Nodejs server through

which it is sent back to the client. Simultaneously, the response along with other user details gets uploaded to the NoSQL database MongoDB. All the data is collected in the MongoDB Database to provide user with other functionalities such as grouping users according to their movie taste, arranging a favorites section for the users and creating a group chat for every particular group of users.

### 5. Technical Stack

#### **XML:**

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability across the Internet.

#### **JAVA:**

Java is a general-purpose computer-programming language that is concurrent, class - based, object - oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

#### **Python:**

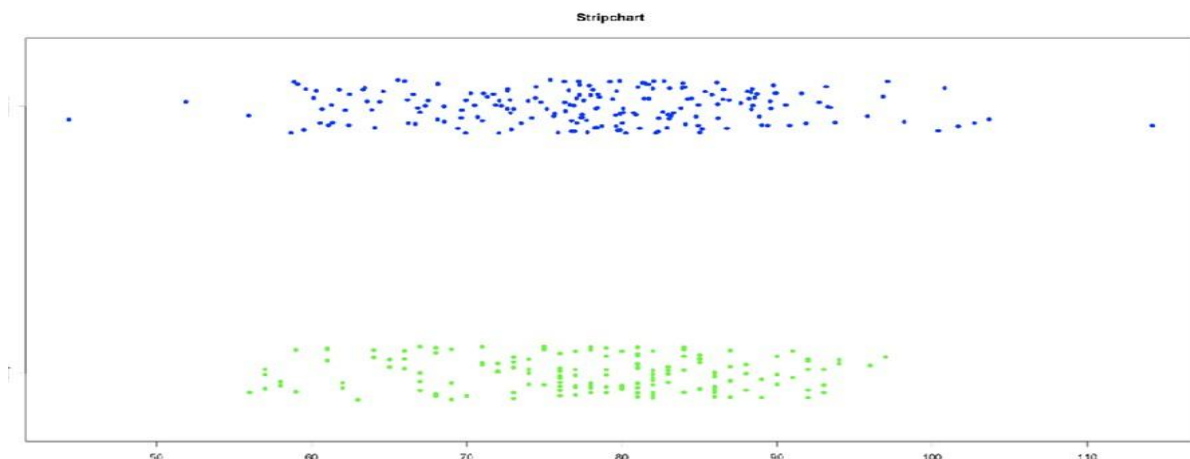
Python is an interpreted high-level programming language for general-purpose programming. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming.

#### **Android Studio:**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on Jet Brains' IntelliJ IDEA software and designed specifically for Android development.

#### **Firebase:**

Firebase is a mobile and web application development platform. Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.



**Figure 1:** A strip chart of negative & positive reviews

**Node.js**

All the REST APIs are made using Nodejs. The data is received by the POST routes from Android through Retrofit. The response generated after the database call is fetched and stored into MongoDB database. The review taken as input from the user is stored into real-time Firebase database. This is later fetched back by the python script and gives the rating based on the Naïve Bayes Algorithm, which when called by a POST request is stored back on the MongoDB database.

**MongoDB**

The NoSQL Database that we are using here is MongoDB. The database calls from Nodejs resulting into an input received from the users end is stored in MongoDB as different documents.

**6. Results**

The android app acts as a social medium in order to forge the gap between the people having interest in the same types of movies and their genre. This would even let one be informed about the recent movies and how people have actually rated them. It also allows to know what all people have rated a movie of a particular genre either 'positively' or 'negatively' and accordingly know about people showing interest in the same genre as you. We can expand it by adding a feature where people can form groups or have options to send personal messages to each other and socialize further. This can even be expanded to a bigger social medium where people can make movie plans depending on people showing interest in same genre movies, form groups, and plan it accordingly.

**7. Conclusion**

The agenda of this paper is to make a social medium through which people with same movie interest can get to know each other and can review and receive some suggestions. The app that we made is fulfilling all the agendas that our project needed to cover. The functionalities of the app are as follows:

- 1) Signup & Login of the User.
- 2) Selecting different Genres.
- 3) Giving Review.
- 4) Receiving the output i.e. whether it is positive or negative. If the result is positive, then the user can proceed to view the list in which users who have given positive reviews for the particular genre can be visible.
- 5) Along with this, people who just want to see the reviews of other people can see their votes, E.g. Which user has given positive votes and how much and same for the negative votes too.
- 6) Inviting users to a movie plan.
- 7) Adding movies to favorites Section.

**References**

- [1] John O'Donovan , Barry Smyth, Trust in recommender systems, Proceedings of the 10th international conference on Intelligent user interfaces, January 10-13, 2005, San Diego, California, USA

- [3] Adomavicius G., Kwon Y. IEEE Intelligent Systems Volume 22 Issue 3, Pages 48-55 May 2007
- [4] Liangxiao Jiang, Harry Zhang, and ZhihuaCai, —A Novel Bayes Model, Hidden Naive Bayes!, IEEE Transaction on Knowledge and Data Engineering Vol 21, No. 10, pp. 1361 – 1371
- [5] W. Zhang and F. Gao, —Procedia Engineering An Improvement to Naive Bayes for Text Classification, vol. 15, pp. 2160–2164, 2011.Sentiment Analysis of movie review comments, uat essenov, Sas a isailovic , May 17, 2009
- [6] Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.
- [7] Pang, Bo, and Lillian Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.