Recommendation System using Big Data Mining on Social Networks

Soham More[1], Prasad Karad[2], Rishi Kataria[3]

[1,2,3]*Department of Information Technology, Pillai College Of Engineering, New Panvel*

Abstract - *Recommendation System is a part of Information Retrieval, Data Mining and Machine Learning which plays a vital role in today's e-commerce industry. Recommendation systems recommend things to users and customers such as books, movies, videos, electronic products and many more. Job recommendation systems are required to achieve high accuracy while predicting job posts to users, which should be relevant to user's input. Although a lot of job recommendation systems that use different strategies for predicting job posts, this paper reflects the efforts that have been put to make the job recommendations on the basis of candidate's input which will match his/her profile. We used 3 different models to recommend the job. The algorithms we will be using are Naive Bayes, Logistic Regression and Random Forest. Along with Machine learning, Natural language Processing is used to capture important words which are extracted through user inference. Through this technique, a good level of accuracy has been achieved.*

Key Words: Naive Bayes, Logistic Regression, Random Forest, Job Recommendation, Data mining, Natural Language Processing, Tokenize, Lemmatize …

1. INTRODUCTION

In job recommendation systems, there are different customers/ candidates, who have different education level, experience and skills. Based on their respective background details, each candidate/user expects to get only those job recommendations which are highly relevant for the respective candidate. A job recommender system is expected to provide recommendations in 2 ways: firstly recommending most eligible candidates for the specified job, to the recruiters and secondly, recommending jobs to the aspiring candidates according to their matching profiles. A recommendation engine or a recommendation system helps to predict what a user is likely to be interested from the list of items. Recommendation system produces the results based on Collaborative filtering technique or Content based filtering technique. In recommendation system applications, they extract the data like users' skills, previous job history, demographic information and other necessary details. Depending upon the extracted data, the job seeker is suggested with new jobs other than what is being searched for.

Here Natural Language Processing comes in picture as every word searched is processed, analyzed, tokenized and lemmatized to base form. The current model is developed for user's skill set.

2. RECOMMENDATION SYSTEMS

In the section below, different recommendation systems are discussed in detail:

2.1 Content Based Recommender Systems

Systems which use a content based recommendation approach analyze a set of documents and descriptions of items which were previously rated by a user, and build a model or profile of user interests which are based on the features of the objects that were rated by that user. The profile built is a structured representation of user interests, which are adopted to recommend new items. The recommendation process consists of matching up the attributes of the user profile against the attributes or feature of an object.

2.2 Collaborative Filtering Recommender Systems

Collaborative filtering has two ways, the new way and the general way. In the newer way or in a narrower sense, collaborative filtering is a method of making automatic predictions about the user's interest by collecting preferences or likelihood information from many users (collaborating). The assumption which underlies about the collaborative filtering approach is that if a person A has the same opinion as a person B on a particular object, then person A is more likely to have B's opinion on a different object than that of a randomly chosen person.

2.2.1 Model Based Collaborative System

Model-Based is a sub category of Collaborative Filtering technique. It is an algorithm which provides item recommendation by first developing a model of user ratings. Algorithms in this category use probabilistic approach and inherit the Collaborative filtering Technique as it computes the expected value of a user prediction, from his/her ratings on other items/objects.

It uses different data mining techniques like Clustering, Bayesian Networks like the Naïve Bayes Model.

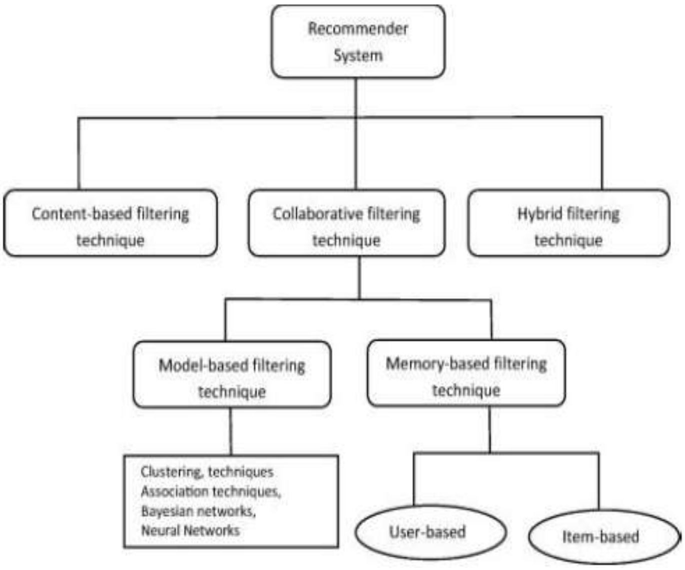
Recommender System
Content-based filtering technique
Collaborative filtering technique
Hybrid filtering technique
Model-based filtering technique
Memory-based filtering technique
Clustering, techniques
Association techniques,
Bayesian networks,
Neural Networks
User-based
Item-based

Fig 1: Recommendation Systems

3. TYPICAL RECOMMEDNATION SYSTEM

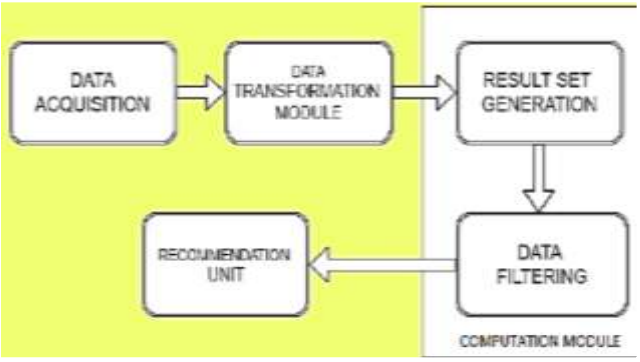

Fig 2: Typical Recommendation System

As shown in Figure 2, a typical recommendation system consists of multiple stages. In data acquisition stage, data is collected and stored from different users. These data consist of profile, past activities, browsing history. Transformation stage deals with the different processes such as sanitization of data and cluster formation. Computation model/unit works with the calculation part of the system and it mainly consists of two parts - Result set generation and Data filtering. The last stage is that the Recommendation unit where recommendations to the users are created relying upon the filtered results set from computation module. Skill recommendations are based on Naive Bayes Classifier, Logistic Regression and Random Forest which takes an account of the skills provided by the user in form of collaboration.

4. THE PROSPOSED SYSTEM

The proposed system starts with acquiring of the dataset. For dataset, we scraped social sites like LinkedIn, Indeed.com. The raw dataset consisted of 24 attributes like 'job_post', 'title', and 'company' and so on. Once the dataset was gathered, we moved towards preprocessing and cleaning of data. For recommending job profile we do not require all the attributes, instead 4-5 attributes are considered. For cleaning purpose, the null values and multiple names of single job post are combined into original job post. Example: Senior Java developer or Java trainee would mean same as Java developer. A custom tokenizer is built to process the text using NLTK, such that each word is transformed to its base form and then tokenized separately. After Preprocessing and creation of processed dataset, the next step involves model selection. Before Model selection the data is split into test and train data. Here, we selected three models as stated. Now hyper parameters are selected in order to limit expensive evaluations of objective function by choosing next input values based on those that have done well in past. By this, it will yield the lowest error on validation set so that, the results are generalize to testing set. Further, a function is created that predicts the label based on user inference. It recommends 2 job position alternatives; given a job requirement. By obtaining probability of class predictions, and picking the top N predictions in descending order (probability will be high for records obtained from downwards), N closest recommendations can be achieved. But here, only two alternatives are recommended i.e. last record or label predicted and second last record or class predicted. The prediction of class label happens through all three models I.e. Naïve Bayes, Logistic Regression and Random Forest. All the textual predictions are transformed into GUI format, for user friendliness. A web application is implemented using Flask-python where user would input data and will get recommended job titles.

4.1 Algorithms

The following are the algorithms that have been used in implementing the project:

4.1.1 Random Forest

Random forest is an algorithm which falls under the technique of random decision. This algorithm works by generating a group of decision trees at training time and outputs the class that represents the mode of classes or the mean prediction of the individual trees. Individual decision trees are generated using a random selection of attributes which act as yes or no node, which will determine split of tree. During classification, each tree casts a vote and the most popular class is returned.

Using the Random forests, the difference can be reduced by getting average of the deep decision trees that are trained with different parts of the training set.

4.1.2 Logistic Regression

Logistic Regression is a classification algorithm which is mainly used for classification problems. It is an algorithm based on predictive analysis and also based on the concept of probability. The hypothesis of logistic regression tends to achieve the cost function between 0 and 1. Therefore linear functions fail to represent it as it may achieve value greater than 1 or less than 0 which is not possible or suitable as per the hypothesis of logistic regression.

When linear regression is used, a formula of the hypothesis i.e.
$h\Theta(x) = \beta_0 + \beta_1 X$
For logistic regression we are going to modify it a little bit i.e.
$\sigma(Z) = \sigma(\beta_0 + \beta_1 X)$
We have expected that our hypothesis will give values between 0 and 1.
$Z = \beta_0 + \beta_1 X$
$h\Theta(x) = \text{sigmoid}(Z)$
I.e. $h\Theta(x) = 1/(1 + e\text{^}-(\beta_0 + \beta_1 X)$

4.1.3 Naïve Bayes

In machine learning we often select the best hypothesis (h) for given data (d). In a classification problem, our hypothesis (h) can be the class to assign for a new data instance (d). One of the easiest ways of selecting the most probable hypothesis from given data, is that we have that we can use our prior knowledge about the problem. Bayes' Theorem provides a way that we can calculate the probability of a hypothesis/happening given our prior knowledge or the probability of object has occurred.

Bayes' Theorem is stated as:
$P(h|d) = (P(d|h) * P(h)) / P(d)$

4.2 Natural Language Processing

Natural Language Processing is a process to manipulate the natural language like speech or text using software or libraries. In basic terms, it refers to how computers analyze, interact and understand the human language and large amount of natural language data. Here we used NLTK or Natural Language Tool Kit to analyze every word from input. NLTK basically is a collection of libraries and programs used to process statistical natural language written in English. We used Tokenization, Lemmatization and stop words feature.

4.2.1 Tokenization

Tokenization is a process of splitting a string or sentence into tokens. These tokens can be words or at times sentences if a paragraph is to be tokenized. Here we have used word tokenize to use each word what user enters.

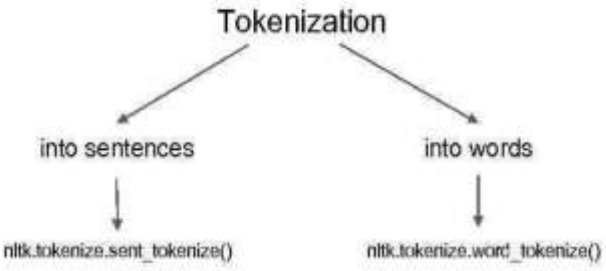

Fig 3: Tokenization chart

4.2.2 Lemmatization

Lemmatization is the process of grouping the inflected forms of same word so that it can be brought down to single root word. In basic terms, it is a process of converting any word with suffix like; 'ing' or 'ed' to basic form. Example, words like 'playing' or 'played' is transformed into 'play'.

5. RESULTS

The following results categorized into Output and Comparison of Algorithms.

5.1 Output

The following are the screenshots of web application:

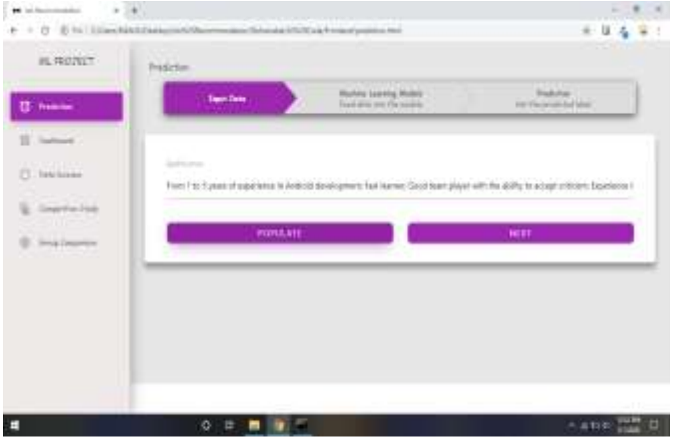

Fig 3: User input

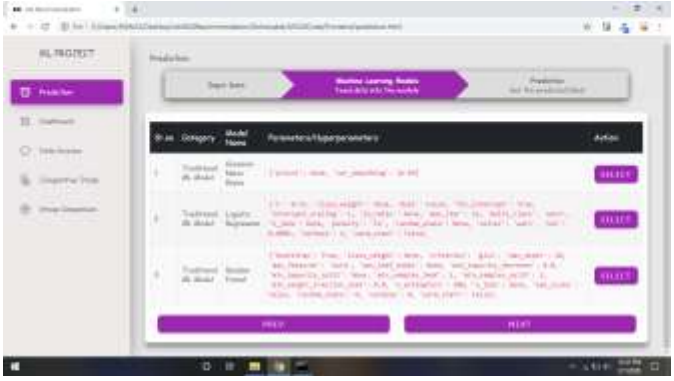

Fig 4: Selecting models to predict

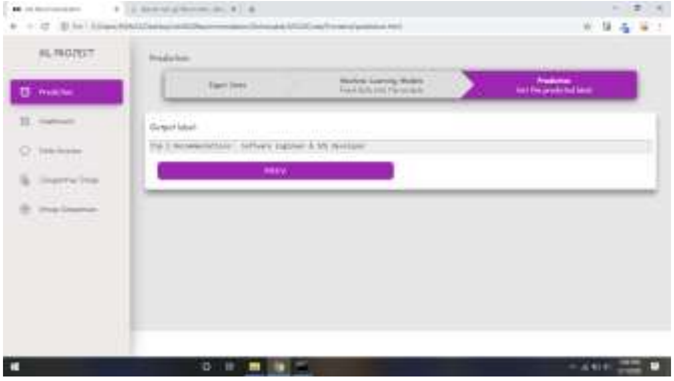

Fig 4: Prediction

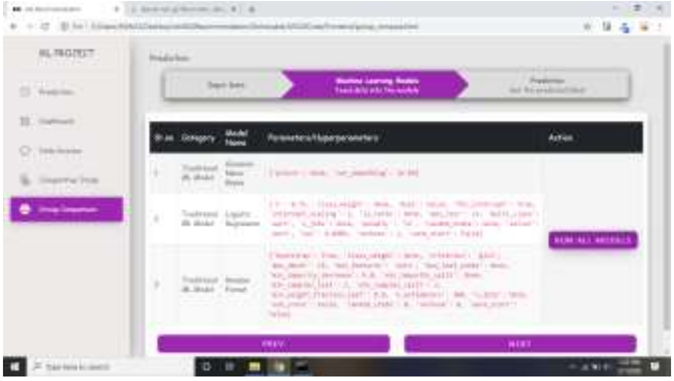

Fig 5: Executing all models

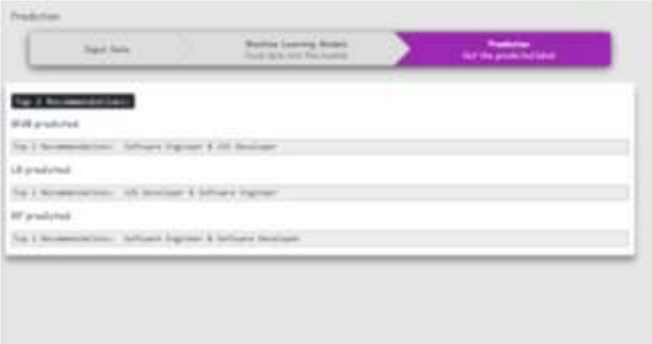

Fig 5: Prediction from all models

5.2 Comparison of Algorithms

We implemented three algorithms i.e. Naïve Bayes, Logistic Regression, Random Forest and we received different accuracy results on testing and training data. The following table represents the result:

Model Name	Training Accuracy"	Testing Accuracy
Naïve Bayes	94.56 %	69.19 %
Logistic Regression	87.58 %	73.84 %
Random Forest	92 %	70 %

6. FUTURE SCOPE

The project can be further extended by providing a way to apply for those jobs in real time by users/candidates. By integrating the system with real time data i.e. gathering data from recruitment sites and then further processing it to provide recommendations plus a feature directly apply for those post or job title. In addition, students can be provided a guide article, on how to move towards dream job (guidelines) with some profile and resume building features. Just as job portal, it will help users and mainly students who will or want to pursue jobs in future.

7. CONCLUSION

The comparative study of various techniques mentioned above is presented in this report. Different techniques are explained with their positive and negative points. The proposed system idea is put down with reference to earlier existing system. Different standard datasets or variable inputs are defined that are used in experiment for this domain system. The applications of this domain is identified and presented. In this proposed, we proposed a new method to recommend jobs to users using Machine Learning as well as Natural Language Processing. With further enhancements in datasets and optimizing the project, more accuracy can be achieved. Three different models were used. Each yields good results about 70% accuracy on testing data. We evaluated the performance of each model according to dataset.

REFERENCES

[1] Joko Sutopo and Khalid Haruna, Collaborative approach for research paper recommendation, 2017

[2] N.Suryana, Collaborative filtering implementations from recommendation system, 2017

[3] Stefan Langer, Research paper Recommendation System, 2017

[4] Classifications of recommenders systems, 2017

[5] Rupali Hande, Ajinkya Gutti, Kevin Shah, Moviemender: A movie recommendation system, 2016

[6] Achin Jain, Vanita Jain and Nidhi Kapoor, Recommendation system based on sentiment analysis, 2016

[7] Rupali Hande, Ajinkya gutti, Kevin Shah, Moviemender : a movie recommendation system, 2016