IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A CLASSIFICATION-BASED PREDICTIVE ANALYSIS FOR ONLINE JOB RECOMMENDATION SYSTEM

[1] S.Saranya [2] K. Harini, [3] J.Srinithi, [4]T.Arjun, [5]P. Senthil Kumar
[1] Assistant Professor
[2,3,4,5] Student Pursuing Bachelor of Engineering
Department of Computer Science and Engineering
Coimbatore Institute of Technology, Coimbatore, India

Abstract: Digital information has grown exponentially with lot of choices for services and products. Hence, filtering, prioritizing and efficient delivering of relevant information to tackle the problem of information overload is needed. This problem is solved in recommendation systems by searching through huge volumes of dynamically generated information to provide users with personalized contents and services. Historical data of users' preference and their purchases to predict items that might interest the users are used by recommendation systems. Recommendation engines mainly use the classifier algorithms for best prediction. So, it is noted that to develop accurate classifier algorithm to enhance the performance of recommendation engines. Random forest and decision tree popular classifier algorithms of choice that shares merits of high accuracy, high classifying speed, strong learning ability and simple construction. In this paper, we analysis a system which uses Random Forest and Decision Tree at multilevel strategies to predict the recommendations based on skills while targeting users' mindset and current trends. Compared to decision tree that provides 84.2% accuracy, random forest is better with 90.1% accuracy provided with feasible performance.

Index Terms **- Predictive data analysis, Recommender systems, classification, decision tree and random forest, Machine learning.**

I. INTRODUCTION

One of the serious social issues faced by common people today is unemployment both in developed and developing countries. Internet has been used for getting information relating to job vacancy and searching for jobs.There has been an exponential growth in the digital information, large amount of digital information available , online services and electronic origin in future years. The need for job hunting has heightened due to tremendous increase in technology. So they need proper educated and experienced person in the particular field. With this recommendation engine they can search with their profile and recruit accordingly. Job seekers can also find suitable jobs for their knowledge.

Huge information overload creates a potential problem of how to handle the search in huge data volume efficiently and to filter it efficiently to deliver the information to the user. In addition of the information needs to be processed for the user rather than filtering the information. This problem highlights a need for information extraction systems that can filter necessary information and predict the information of users' interest. This system is called recommendation systems [1]. In this, employers upload their job offers in their job portal, the recommendation engine meet the requirements for employers and job seekers. It applies machine learning and data mining techniques for filtering information and using that it can predict if user would like it or not. Many commercial application of recommendation engine can be found in many e-commerce sites such as Amazon, Flipkart. Now-a-days people search job opportunities in many online websites like linked-in, Guru99, amcat etc. They are mainly using web for recommending products and services to users. This engine provides two functionalities; it helps the users in dealing with information overload by letting them know products and services in recommendation. Next they help business make more profit i.e. by selling products with discounts thereby they are selling more products. At first, most of the job hunting websites provides all the jobs to all the seekers, the seekers have to go through all the information ,skills and requirement for the jobs and find the relevant one and have to apply for the particular job .It's a tedious process for seekers and recruiters too. Hence the modern techniques have evolved, which uses several algorithms and make it a very simple one. By the profile they have given the recommendation engine itself finds and gives the best suitable jobs for seekers. It made works of recruiters also an easy one by giving them the matched profiles of a person they want. A recommender engine mainly uses the classifier. So, it is important to develop accurate classifier [2]. There are different classification techniques like KNN, Naive Bayes classifier, Support Vector Machine, decision tree and random forest [9]. Amongst all decision tree and random forest algorithms are easy to build and fast classifiers. They produce much accurate result than other classifier in less time. So, in this paper, analysis is to check whether the decision tree or random forest provides high accuracy, better and efficient job recommendations.

II. RELATED WORK

Till now, there has been a exponential growth in the development of recommender sites. The numbers of people using the recommender engines are increasing tremendously and it is very important for these systems to generate recommendations that are close to the items of users' interest. Recommender engines are categorized into collaborative filtering, content-based or hybrid systems [3], where a content-based recommender engine recommends jobs based on content information of the jobs. Collaborative filtering recommender systems [4] taking into account the taste of user to recommend jobs. Under the assumption, the user will be interested in jobs where users' interest similar to them have rated highly and hybrid combine or unify, user and content oriented approaches and have shown to outperform their two-mode counterparts in many scenario. To improve the performance of recommender engines various classification approaches have been used for recommender engines.

In collaborative filtering recommender systems, assumptions are based on the people thoughts i.e., who agreed in the past, will agree in the future too. In [6], authors have proposed a unique switching hybrid recommendation approaching by combining a Naive Bayes classification approach with the collaborative filtering recommendation approach.

There are two categories in Collaborative filtering. Memory-based collaborative filtering is user based and it makes a prediction from the previous rated items of entire collections by the user [7]. The main advantage is the most recent information is quickly incorporated and the disadvantage is that for huge databases search of neighbours is slow. Authors proposed Model-based collaborative filtering algorithm in order to avoid inconvenience. In model-based collaborative filtering, there are wide variety of data-mining algorithms can be applied. First method is Neural networks [8]. Authors have used model-based Item-to-Item collaborative filtering in Amazon recommender engines. In online computation [10], their algorithm produces recommendations in real time, huge datasets; high quality construction does not require number of customers and number of items in product catalog. These systems results in poor quality recommendations and reduced coverage due to suffer from scalability, data sparsity, over specialization and cold-start problems. A hybrid recommendation technique that combines the best features of two recommendation techniques into one hybrid technique has been proposed to achieve higher performance and overcome the drawbacks of traditional recommendation techniques. To avoid cold-start, sparseness and/or scalability problems it is used for an attempt. Decision tree classifier and random forest classifier is trained on content information. Use of decision tree classifier reduces search time of finding neighbours but in Random forest classifier along with reduction of search time it provides high accuracy.

III. DECISION TREE CLASSIFIER

Decision tree is predominantly the most powerful and popular algorithms. Decision-tree algorithm comes under the category of supervised learning algorithms. It works for both categorical as well as continuous output variables. The contributions adopted in this work are:

- In the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain and for gini index, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or internal node as shown in Fig.1.

4.1 ALGORITHM

- Find the best attribute and place it on the root node of the tree.
- Now, split the training set of the dataset into subsets. While making the subset make sure that each subset of training dataset should have the same value for an attribute.
- Find leaf nodes in all branches by repeating 1 and 2 on each subset.
- While implementing the decision tree we will go through the following two phases:
- *Building Phase:*
  - Preprocess the dataset.
  - Split the dataset from train and test using Python sklearn package.
  - Train the classifier.
- Operational Phase
  - Make predictions.
  - Calculate the accuracy.

4.2 DATA PREPROCESSING

- Before training the model we have to split the dataset into the training and testing dataset.
- To split the dataset for training and testing we are using the sklearn module *train_test_split*
- First of all we have to separate the target variable from the attributes in the dataset.

  X=balance_data.values[:, 1:5]
  Y=balance_data.values[:,0]
- Above are the lines from the code which separate the dataset. The variable X contains the attributes while the variable Y contains the target variable of the dataset.
- Next step is to split the dataset for training and testing purpose.
- X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=100)
- Above line split the dataset for training and testing. As we are splitting the dataset in a ratio of 70:30 between training and testing so we are pass *test_size* parameter's value as 0.3.
- *random_state* variable is a pseudo-random number generator state used for random sampling.

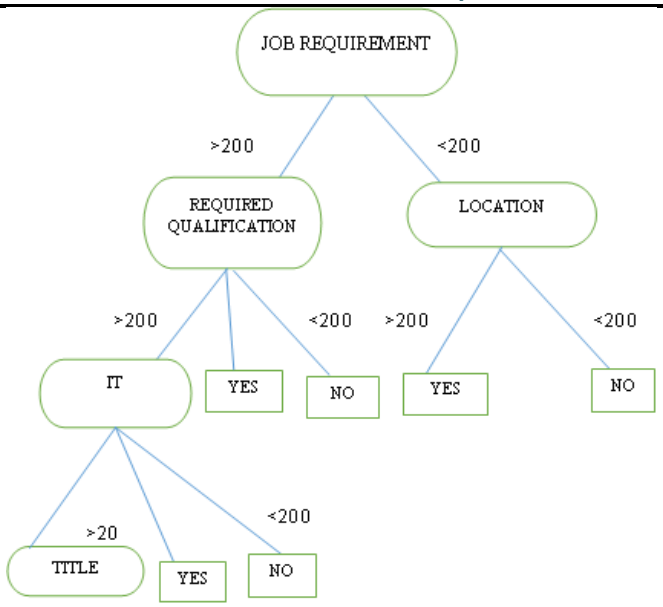


Fig.1. Decision tree classification strategy in Job Recommendation system

IV. RANDOM FOREST CLASSIFIER

Random forest is the most accurate classifier and for large number of data set it works efficiently. It can predict the missing data effectively even when large portions of data are missing and without pre-processing. The advantage is that is able to avoid over fitting of the training dataset which is not possible by other machine learning techniques. To generate trees, random subset of training data is used. After the forest has been trained, the test rows are passed through the forest. The mode in output class in each tree is considered as the output of random forest. To predict the label of users, random forest classifier is used as shown in Fig.2. The dataset is classified into train and test datasets. It is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. The idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

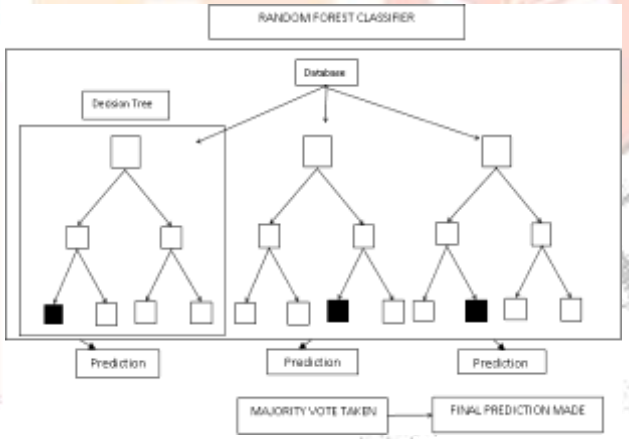


Fig.2. Random forest classification approach in Job Recommendation system

4.1 ALGORITHM

- Find a random K data points from the training set.
- Build the decision tree associated with those K data points.
- Choose the number N tree of trees you want to build and repeat step 1 & 2.
- For a new data point, make each one of your N tree trees predict the value of Y for the data point, and assign the new data point the average across all of the predicted Y values.
- Ensemble model made of many decision trees using bootstrapping, random subsets of features and average will predict the results.

V. PROPOSED SYSTEM

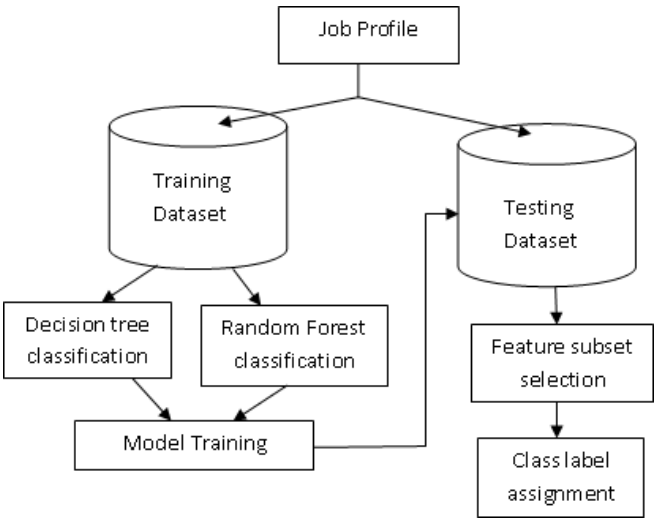


Fig.3. Classifier based Model generation in Job Recommendation system

5.1 DECISION TREE BASED STRATEGY

The Decision Tree algorithm is the most common data mining algorithm that is used in predicting the classifier. When we provide the Decision Tree algorithm with a set of attributes, it builds a tree-like model based on the various rules or decisions that it makes on the data set. Here, the leaf nodes represent the class attribute; the internal nodes represent the remaining attributes used.

For predicting the classifier and the branches represent the value of those attributes. The Decision tree algorithm with a Percentage Split method divides the data set into 70% Training data and 30% Test data and gives an accuracy rate of 84.61%. This accuracy improves drastically when we use the Random Forest.

5.2 RANDOM FOREST APPROACH

The Random Forest algorithm or in other words the Random Decision Forests algorithm are an ensemble classification algorithm which create many decision trees during the training phase and then predicts the classifier for all those decision trees and thereby gives the best possible accuracy rate from all the decision trees.

The Random Forest algorithm with a Percentage Split method divides the data set into 70% Training data and 30% Test data. The Random forest will take k random attributes and create n decision trees. And from those n decision trees, it will take the best and predict the result. The results of both the methods can be seen below; Out of 70 instances it correctly predicts 59 instances. Here, the Random Forest algorithm could only classify 63 instances correctly out of the total 70 Test instances. So, the accuracy rate achieved in the Percentage Split method was 90.01%.

VI. RESULT AND DISCUSSION

A. DECISION TREE CLASSIFIER

Entropy:

$$H(x) = -\sum_{j} p_j \log_2 p_j \qquad (1)$$

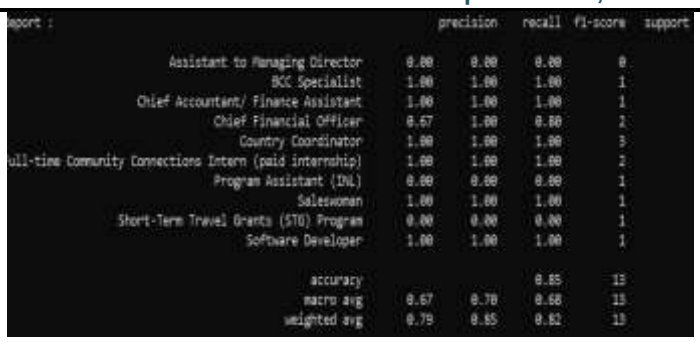
If a random variable x can take N different value,the jth value X,with probability p(Xj),we can associate with the entropy as in Eq.1.

Gini index:

Gini index and information gain both of these methods are used to select from the *n* attributes of the dataset which attribute would be placed at the root node or the internal node.

$$\textbf{Gini Index} = 1 - \sum_{j} p_{j2} \qquad (2)$$

Gini Index is a metric to measure how frequently a randomly chosen element would be incorrectly identified as in Eq.2. It means an attribute with lower gini index must be preferred. Sklearn supports “gini” criteria for Gini Index and by default, it takes “gini” value. For the given dataset, Decision Tree Classifier will produce a result of 84.61% accuracy as shown in Fig.4.

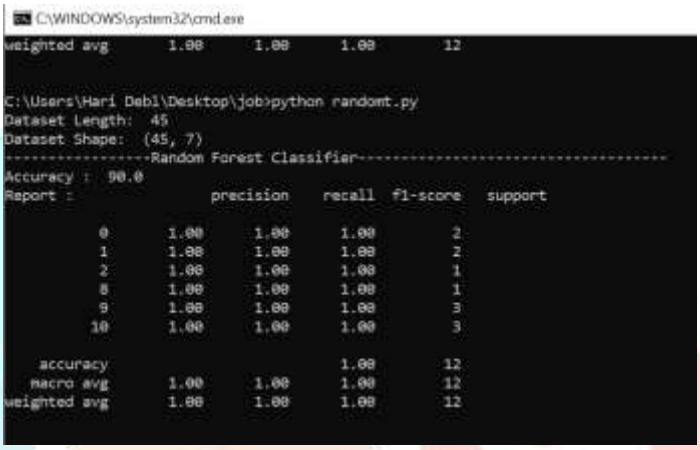


eport :	precision	recall	f1-score	support
Assistant to Managing Director	0.00	0.00	0.00	0
BCC Specialist	1.00	1.00	1.00	1
Chief Accountant/ Finance Assistant	1.00	1.00	1.00	1
Chief Financial Officer	0.67	1.00	0.80	2
Country Coordinator	1.00	1.00	1.00	3
ull-time Community Connections Intern (paid internship)	1.00	1.00	1.00	2
Program Assistant (INL)	0.00	0.00	0.00	1
Saleswoman	1.00	1.00	1.00	1
Short-Term Travel Grants (STG) Program	0.00	0.00	0.00	1
Software Developer	1.00	1.00	1.00	1
accuracy			0.85	13
macro avg	0.67	0.70	0.68	13
weighted avg	0.79	0.85	0.82	13

Fig.4. Performance evaluation of the recommendation system using Decision tree classifier

B. RANDOM FOREST CLASSIFIER

The Random Forest Classifier will give the result as shown in Fig. 5. , which shows accuracy higher than decision tree classifier. For the given dataset, Random forest classifier accuracy is 90.01%.



```
C:\WINDOWS\system32\cmd.exe
weighted avg       1.00      1.00      1.00        12


C:\Users\Hari Debl\Desktop\job>python randomf.py
Dataset Length:  45
Dataset Shape:  (45, 7)
----------------Random Forest Classifier----------------------------------
Accuracy :  90.0
Report :               precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         2
           2       1.00      1.00      1.00         1
           8       1.00      1.00      1.00         1
           9       1.00      1.00      1.00         3
          10       1.00      1.00      1.00         3

    accuracy                           1.00        12
   macro avg       1.00      1.00      1.00        12
weighted avg       1.00      1.00      1.00        12
```

Fig.5. Performance evaluation of the recommendation system using Random forest classifier

The comparative accuracy based analysis is found in Fig.6a,6b shows that when compared to decision tree, performance of random forest gives accurate true predictions.

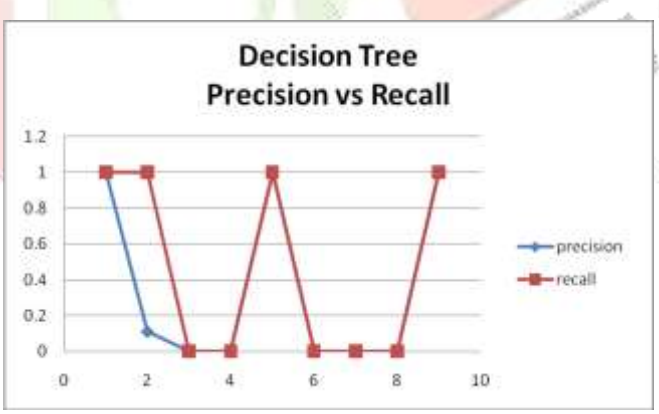


Fig.6a. Accuracy based comparative performance analysis between Decision tree classifier and Random Forest classifier

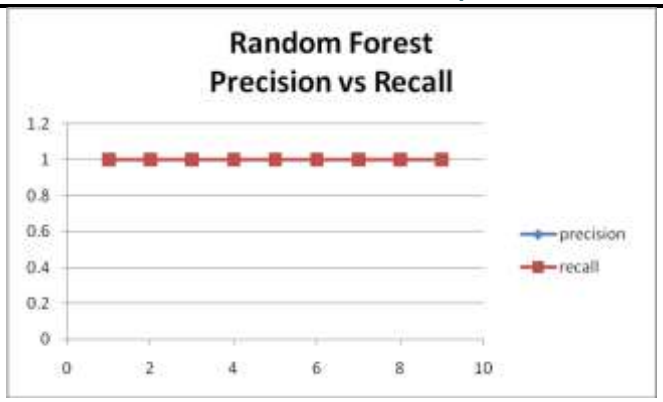


Fig.6b. Accuracy based comparative performance analysis between Decision tree classifier and Random Forest classifier

VII. CONCLUSION

The proposed strategy analyses the job application to best classify them in the nature of category assigned. From the above observations, the Decision tree classifier and Random Forest Classifier is most suitable to predict the job titles with a good precision and high accuracy. Despite the accuracy is satisfactory, several other classification, clustering and machine learning approaches can be incorporated for the betterment of the model. In future, the big data analytics tools can be adopted by choosing dataset of high dimensionality.

REFERENCES

[1] Mustansar Ali Ghazanfar and A. P. Bennett, "Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering", IAENG International Journal of Computer Science, 19 August 2010.

[2] Zhi Qiao, Peng Zhang, Yanan Cao, Chuan Zhou and Li Guo, "Improving Collaborative Recommendation through Location-based User-Item Subgroup", an 14th International Conference on Computational Science, Vol. 29, 2014.

[3] M. Balabanovic and Y. Shoham, "Content-Based, Collaborative Recommendation", Communications of the ACM, Vol. 40, No. 3, pp. 66-72, 1997.

[4] D. Goldberg, D. Nichols, B. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", Communications of the ACM, Vol. 35, No. 12, pp. 70, 1992.

[5] Tong Zhang and Vijay S. Iyengar, "Recommender Systems Using Linear Classification algorithms", The Journal of Machine Learning Research 2, 2002.

[6] Mustansar Ali Ghazanfar and Adam Prugel-Bennett, "An Improved Switching Hybrid Recommender System Using Collaborative Filtering and Naive Bayes Classifier ", International MultiConferernce of Engineers Computer Scientists, Volume 1, 2010.

[7] F.O. Isinkaye, Y.O. Folajimi and B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, 2015.

[8] Saddys Segrera and Maria N.Moreno, "Web mining based framework used for solving usual problems in recommender systems: A case study used for movies' recommendation", Neurocomputing Elsevier Journal, 2015.

[9] Saranya.S et al," A Survey on Improving the Clustering Performance in Text Mining for Efficient Information Retrieval", International Journal of Engineering Trends and Technology (IJETT), ISSN 2231-5381, Volume 8, No.5, February 2014.

[10] M.BalaAnand et al, "An Enhanced Graph based semi-supervised learning algorithm to detect fake users on Twitter", Journal of supercomputing, issue 9,2019.