Efficient Job Recommendation System Using Voting Classifier

Submitted By,

Md. Hasan Bhuiyan

Sakibul Islam

Md. Nurul Amin

Md. Shahin Shanaous

Syeda Sumiha Jahan

A thesis presented for the degree of

Bachelor of Science

in

Computer Science and Engineering

Bangladesh University of Business and Technology

November, 2021

Declaration

We hereby declare that the project entitled ‘Efficient Job Recommendation System Using Voting Classifier’ submitted for the degree of Bachelor of Science and Engineering in the faculty of Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT) is our original work and that it contains no material which has been accepted for the award to the candidates of any other degree or diploma, except where due reference is made in the next of the project to the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in this research work.

Md. Hasan Bhuiyan _____

Sakibul Islam _____

Md. Nurul Amin _____

Md. Shahin Shanaous _____

Syeda Sumiha Jahan _____

Dedication

We would like to dedicate our work to diligent researchers for whom the modern age is adequate for robust technology devices.

Approval

This research work ‘Efficient Job Recommendation System Using Voting Classifier’ report submitted by Md. Hasan Bhuiyan, Sakibul Islam, Md. Nurul Amin, Md. Shahin Shanaous, Syeda Sumiha Jahan students of Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT), under the supervision of Md. Saifur Rahman, Assistant Professor, Computer Science and Engineering has been accepted as satisfactory for the partial requirements for the degree of Bachelor of Science Engineering in Computer Science and Engineering.

_____

Md. Saifur Rahman

Assistant Professor

Department of CSE

Acknowledgement

We want to convey our deepest gratitude to The almighty God, who has shown kindness to our family and us during this journey till the completion of this research.

We want to express our sincere gratitude to Md. Saifur Rahman, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Business and Technology (BUBT). This research would not be accomplished without his supervision. We are thankful to him for his exceptional supervision and for devoting his whole attention to the development of this project. We owe him a lot for his assistance, encouragement, and direction, which eventually formed our mindset as researchers.

Finally, we are grateful to all of the faculty members of the CSE department at BUBT who have enabled us to accomplish this research work with adequate guidance and support over the last four years.

Abstract

Job recommendation is the process by which a person or individual is recommended for a suitable job according to the person or individual information. In this modern area of technology, searching job is playing a prominent role for job seekers. Nowadays, random people or fresh graduates narrowly realize which job may be appropriate to establish a better career in such a colossal job marketplace. Understanding the suitable career path for candidates requires aid from human expertise. But it is not viable to acquire the actual job track by human expertise. As a result, candidates are not able to get the perfect job. In this paper, an automated machine learning-based system is flourished that collects specific information of an individual and tame significant summons for the job seekers as candidates can straightforwardly get the preferable job through recommendation. The system gives predictions for one or multiple jobs based on the candidate's specifications. We have collected few specific data of individuals to assemble the dataset and performed them in five comparative machine learning models, respectively Random Forest, K-nearest neighbors (KNN), Support Vector Machine (SVM), Naïve Bayes, Voting Classifier. Maximum 89% accuracy has been achieved to recommend the best possible job using the Voting Classifier.

List of Figures

List of Tables

Contents

Introduction

1.1 Introduction

A recommender system is defined where the information provided by humans, and the system collects the data and gives appropriate instructions to the recipients. It can be further defined as systems that predict the product users buy the most and whose demand is at the top of the market. There are different types of recommendation systems commonly used: content-based recommender system, collaborative filtering recommender system, hybrid recommender system, etc. Content-based recommendation systems deal with user specific classification issues and user preferences. In content-based recommendation system, the individual keyword is used to describe the classifications, and a user profile is created to indicate the type of category the user prefers. A collaborative recommendation system recommends products through user collaboration.

There are two types of collaborative recommender systems: user-to-user based on user match and item-to-item based on item match. The hybrid recommendation system gathers two or more recommendations systems that benefit from their complementary benefits in various ways. In a job recommendation system, different candidates have various educational qualifications, experience, and skills; based on their academic qualifications, experience, skills, and descriptions of related subjects; each shows the suitable job. There are various types of job recommendation systems: CASPER (Case-Based Profiling for Electronic Recruitment) [1], Bilateral People-Job Recommender [2], Proactive [3], Absolventen.at. [4], LinkedIn [5], PROSPECT [6], eRecruiter, iHR [7]. Nowadays, the organization got a vast amount of Cvs for a particular job posting. Finding the appropriate applicants from the CVs is a time-consuming chore for any organization these days. The categorization of an applicant's resume is a challenging, time-consuming, and resource-intensive procedure. We designed an automated machine learning-based system that recommends appropriate applicants based on a specific job description to succeed in this issue [8].

The system takes the information from the job seekers' CVs and analyzes the Machine Learning method to recommend which job will be best for which candidate. The

system will recommend an appropriate job to the candidate, and an organization can easily find a relevant candidate for their required job post from a massive amount of applicants. Protecting data privacy is the system's principal purpose, and less dependent on personal information to generate output. The contribution of this paper includes the implemented job recommendation system that is entirely accurate and efficient for both the candidates and recruiters.

1.2 Problem Statement

The number of educated unemployed is increasing day by day. Currently, highly educated youth are not getting jobs according to their qualifications or are not getting jobs in the profession of their choice. The result is that they have to do relatively less qualified work. Many job seekers are not guaranteed employment based on merit and qualifications. There is a problem in taking and implementing effective measures to overcome unemployment through qualitative change. At present, the right person cannot get the right action despite having the desire to do it. New job seekers often do not get the desired job due to a lack of previous experience. Finding a suitable candidate out of thousands of job candidates is difficult for the company.

1.3 Problem Background

At present, in Bangladesh, you still have to fill up the forms separately for different jobs as before, which takes a lot of time and effort. Candidates do not understand which job will be good for them, and the institutions do not understand which qualification will be suitable for them. The critical challenges of job recommendation systems are,

- Job seekers will be able to use all their data all the time by submitting it only once. And the data of all job candidates will be stored in the server of our system.
- All valid candidates will be able to apply for the job of their choice.

- The system will recommend suitable jobs to candidates, and organizations can easily find the appropriate candidate for their required post from many candidates.
- With the available data of applied candidates for the specific job, our system will recommend its perfect employee by prediction.

Although researchers are constantly solving these challenges, the implementations are not fulfilled yet.

1.4 Research Objectives

The purpose of the research work is to implement a job recommendation system that solves the pasts previously mentioned vital challenges.

1.5 Motivations

The increased use of the Internet has increased the demand for online job searching. In the past year, the amount of people who searched for jobs on the Internet has been almost a billion. According to Jobvite's report 2014, 68% of online job seekers are college graduates or postgraduates. The highly competitive and dynamic nature of the job market and personal preferences and goals lead individuals to change their jobs at some point in their lives. Moving to a new job, on the other hand, is a difficult decision that can be influenced by a variety of factors such as salary, job description, and geographical location. The main issue is that most job-searching websites display recruitment information to website visitors. To find jobs to which they want to apply, job seekers must sift through a plethora of information. The entire process is time-consuming and inefficient. So, we create an automated system that can recommend jobs to people based on their previous job histories and CVs, thereby making the process of finding a new job more manageable.

1.6 Flow of the Research

The research procedure was divided into several stages. After deciding on a research topic, we first looked into the fundamentals of machine learning, which are required for our research. Following the practice, we looked at the most promising job recommendation architectures. We investigated the flaws of the proposed architectures and developed our recommendation system.

Following the completion of the plan, we presented the overall method. We gathered a common dataset and ran tests and evaluations on our implemented architecture to validate the proposed model. Finally, we completed our thesis. Figure 1.1 depicts the overall steps of the research method in a flow diagram.

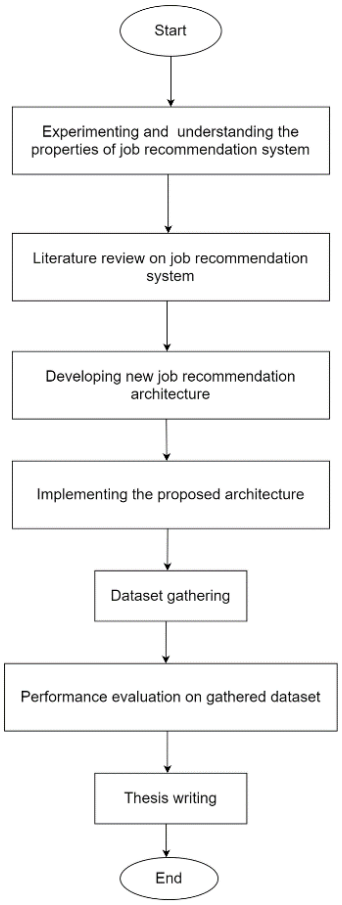


Figure 1.1. Flow of the research work.

1.7 Significance of the Research

The findings of the study will be of great convenience to the researchers that the job recommendation is possible to the probable best candidates very accurately and efficiently. The study points out the most available features of the job recommendation system. The increasingly complex problems to recruit candidates for organizations are focused on using this research in solving problems. This research will impact the researchers to explore a large number of data sets with machine learning competence. Further research will be satisfactory for the present organization's execution level of the job recommendation system, which will be much productive in recruiting candidates fairly.

1.8 Research Contribution

The overall contribution of the research work includes

- We scrutinize that most state-of-the-art architectures depend on the basic algorithm for a job recommendation, primarily on machine learning.
- We investigate the most successful method for job recommendation comparatively with other recommendations systems.
- We define machine learning techniques for job recommendations for a large number of data sets.
- We implemented a job recommendation system that is entirely accurate and efficient for both the candidates and recruiters.
- We extend our job recommendation implementation in job segmentation.

1.9 Thesis Organization

The thesis work is organized as follows.

Chapter 2 highlights the background and literature review on the field of the efficient job recommendation system.

1.10 Summary

In this chapter, we have included a comprehensive overview of that problem, specifically the themes of our thesis work alongside inspiration thesis work output. This section also illustrates the overall steps on which we have carried out our thesis work.

Background

2.1 Introduction

The demand for job is increasing along with the search for jobs by the applicants. There are many types of job, most of which are general. There are very few job sites with recommended or advanced features. The ones there are for experienced job seekers or those who want to retire and get a job again. There are few opportunities for job seekers to know which jobs are suitable for them as there is no such recommendation system. Aspirants endure the most when it comes to getting a job. This will reduce their suffering a lot. Many unemployed persons don't know which career track is most appropriate for them. The system will be beneficial for job seekers.

2.2 Literature Review

In recent times, the job recommendation system has been very much required. Therefore, many researchers have been experimenting with it extensively. Significant work is being done with the job recommendation system. Machine learning and deep learning are the two most used methods nowadays. There are different types of job recommendation systems, and different methods have been applied by analyzing one or the other. Rafter et al. implemented a parallel hybrid method that combines content-based recommender system and collaborative filtering recommender system. CASPER system seeks to minimize data overload of jobs, provides personalized recommendations for candidates. CASPER Personalized Case Retrieval (PCR) uses two processes where first the match between the candidate's question and the job is found, and then it is done server-side. In the second stage, towards the client, the relevance of the rescued job is calculated according to the target user's profile, and the tasks are finally sorted in order of significance [1].

Malinowski et al. introduced a probabilistic cv recommender and a probabilistic job recommender. The bilateral job recommendation method helps to provide two-way recommendations, where in the first step, the CV is selected. In the second step, the job is recommended [2]. Hong et al. try to group the applicants into three main groups and a different approach is applied to each group based on user clustering [10]. Singh et al. proposed an approach that combines content-based recommender system approach for recommendation [6]. Most of the writing explores coordinating with calculations that address the reciprocally of the suggestion [2, 12, 13], the test of the different client qualities that can be utilized to coordinate with job searchers with occupations [10, 13, 14], and the thought of interpersonal organizations for the coordinating with measure [2, 13, 15].

Paparrizos et al. try to anticipate an employee's upcoming job change; their algorithm uses all previous job transitions and data linked with workers and institutions. They used a massive sample of job transitions and accompanying metadata gathered from publicly available employee profiles on the internet and presented the Decision Table/Naïve Bayes hybrid classifier (DTNB) results. They didn't even have a good idea of the social aspects [11]. "Resume Classification and Matching" is an automated system that provides the time-saving ethical screening and shortlisting process. It would undoubtedly speed up applicant preference and decision-making. They used the Genism library in their proposed model, but it may lose important information using this library [16]. If important information is lost, the system may have a wrong prediction. Alternatively, there is no risk of losing the data of our proposed model because every data has given equal importance.

Appadoo et al. [17] proposed a similar concept of a machine learning-based job recommendation system into a multi-class classification problem that takes input from CVs and divides the feature into five models. The system outputs a job-fit score indicating the best candidate for the particular job. A little less accuracy has been noticed in their divided models. The authors gained 59% accuracy using Voting Classifier in the summary of job satisfaction model. In comparison, our proposed architecture works with Random Forest, KNN, SVM, Naïve Bayes, and Voting Classifier algorithm. Among these methods, the Voting Classifier performed well, and accuracy is 89%. For the recruitment of data, the proposed system used the Variable-

Order Bayesian Network (VOBN) model. According to the verdicts, the VOBN model may give HR professionals accurate and interpretable insights [18]. Appadoo et al. [17] proposed a model that gives predictions based on users' personal and professional information. Their system takes address, marital status, personality, lifestyle, and family from the users. Nowadays, secured data is essential. User information is not secure due to taking personal information. But in our proposed model, we are less dependent on personal information for prediction. Based on education, skills, and working experience, our model can maintain higher accuracy.

2.3 Problem Analysis

Although the principal impediment of deep neural networks implies that it requires vast data to outperform other strategies. The potential of content-based filtering to build on the consumers' current interests is limited. The significant impediment of user-based collaborative filtering (CF) is data sparsity and more economical recommendation quality. For our dataset, machine learning algorithms are more suitable than others.

2.4 Summary

This chapter illustrates the latest implementation of job recommendation systems with disadvantages. The thesis work aims to eliminate the error as possible and develop a job recommendation system that is appropriate and stable for fixing the current challenges.

Proposed Model

3.1 Introduction

This chapter discusses the feasibility analysis of the job recommendation system and the requirements demanded in this model. Finally, this chapter explains the model's overall architecture, which is given by a detailed walkthrough.

3.2 Feasibility Analysis

The thesis work required five researchers with one supervisor and took eight months to be executed. The research work required technical support including, hardware and software. The research work also the generation of dataset needed and evaluation process that the researchers also perform. The comprehensive data collection of the thesis work is executed, considering the legal feasibility of the dataset. Also, the thesis work did not require any financial support from the institution and supervisor.

3.3 Requirement Analysis

To conduct the proposed architecture of the overall requirements, include,

- High-performance computing device.
- Opensource software libraries for scientific computations.
- Opensource software libraries to implement the machine learning model.

3.4 Research Methodology

In this research, various machine learning algorithms have been experimented with to predict suitable jobs for job seekers. The method predicts or recommends new jobs based on job seekers education qualifications and experience information. The complete methodology is divided into few steps: (1) Data collections, (2) Data preprocessing and (3) Classification. Figure 3.1 illustrates the overall workflow of the process.

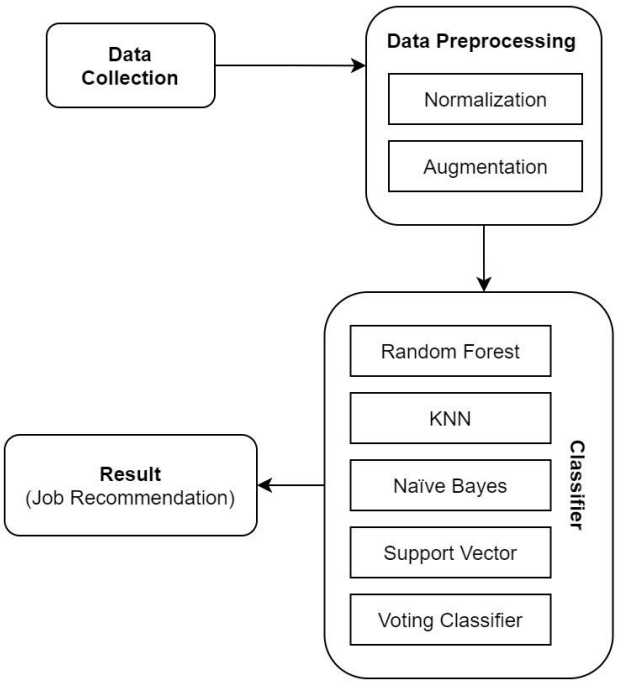


Figure 3.1: Workflow of Job Recommendation

3.4.1 Data Collection

We collected data from more than 400 profiles through an online survey. The profiles contain information about the job seekers educational and professional backgrounds. Each profile is divided into two sections. The first section contains information about job-seekers personal and educational information, such as age, gender, department and various examination results. The second section contains information about job-seekers professional backgrounds. This section includes special qualifications, previous job title, and previous job period.

In our dataset, there are thirteen descriptive features and one target feature. Descriptive features are age, gender, Secondary School Certificate (SSC) group, Higher Secondary Certificate (HSC) group, honor's department, master's department, SSC result, HSC result, honor's result, master's result, special qualification, previous job title, period of previous job and target feature is new job title. Special qualification and new job title contain single or multiple categorical data in each profile. The target feature is multioutput categorical types.

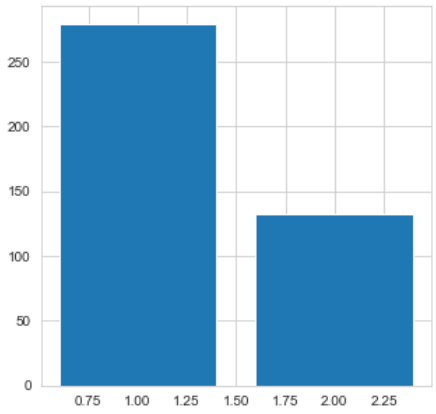


Figure 3.2: Distribution of Gender

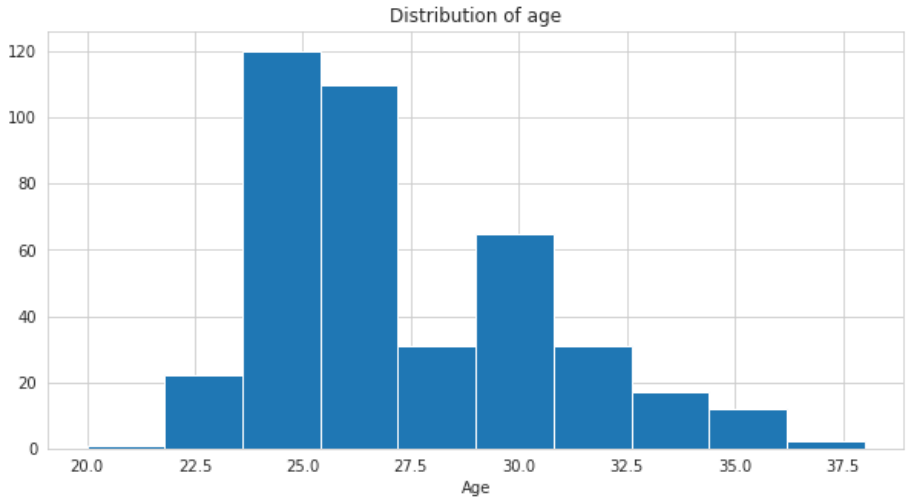


Figure 3.3: Distribution of Age

Figure 3.2 & 3.3 shows the distribution of gender and age. The amount of male profiles in the dataset is higher than female. Most profiles age in the dataset is in the range of 22 to 30 years. TABLE 3.1 shows the number of Profiles, Honor's & Master's department, Special qualification and Job title.

Description		Amount
Number of Profiles	Without Augmentation	411
	With Augmentation	1312
Department		23
Special Qualification		45
Job Title		64

Table 3.1: Dataset Statistics

3.4.2 Data Preprocessing

At the beginning of data preprocessing, all string type categorical values have been replaced with specific unique numeric values. Then the dataset has been normalized. Normalization implies rescaling the values into a range. Age and examination results values have been normalized in the range of 0 to 1. Since Special Qualifications and New Job titles have categorical type multiple values, need to handle these multiple values to apply the algorithm. To handle multiple values, the data of each profile's special qualification and new job title is first sent to a list; in this case, the length of the list is the maximum amount of data. Then each of the values in the list has been converted to a new column. As a result, the number of descriptive features changes from 13 to 22 and the column number of the target feature changes to 6. The data has been augmented based on the value of the SSC, HSC, Honor's & Master's results. The data is augmented by adding small noises to the input values.

3.4.3 Classification

We have experimented with different types of machine learning classifier to get the best output in our dataset by using scikit-learn library. This segment discusses the models used to create classifiers that perform the most suitable job recommendations.

i. Random Forest:

Random Forest operates in two stages: the first is to generate the random forest by merging N decision trees, and the second is to make predictions for each tree generated in the first state [16, 17].

Gini = $1 - \sum_{i=1}^{c} (r_i)^2$ ……………………………………………… (1)

Entropy = $\sum_{i=1}^{c} - r_i * \log_2 r_i$ ……………………………….… (2)

Here, ri is relative frequency of the class, c is the number of classes. Equation (1) calculates the Gini of each branch on a node using class and probability. Gini indicates which branch is more probable to happen. Equation (2) specifies how nodes should branch based on the probability of a specific result. To get the maximum output, we set the values of max_depth=120, min_samples_split=5, random_state=1. Here, max_depth is the most profound route between the root node and the leaf node, and min_samples_split is the least number of samples needed to split an inner node.

ii. K-nearest Neighbors

K-nearest neighbors (KNN) is a lazy learner and nonparametric machine learning classification based on the supervised learning approach [17]. It predicts the values of new data points using feature similarity, implying that the new data point will be allocated a value depending on how nearly it resembles the points in the training set. During the training phase, the dataset is saved, and when new data is received, it is classified into a category that is quite related to the new data. In this case, the value of K(the nearest data point) must be selected. Here, we select the value of K=3 (n_neighbors). To perform the KNN algorithm, Euclidean distance is used to calculate the distance between each testing data point and each training point.

Euclidean (X, Y) = $\sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$ ………………………… (3)

Equation (3) is defined as the Euclidean distance between two instances X and Y in a n-dimensional feature space.

iii. Naïve Bayes

Naïve Bayes is a classification approach based on the Bayes Theorem and the premise of predictor independence. It provides predictions based on the probability of an element. The term naïve refers to the assumption that the model's features are independent of one another. Assume that c is class variable and X is a dependent feature vector where:

$X = (x_1, x_2, x_3, \ldots, x_n)$

$P(c|X) = \frac{P(X|c)P(c)}{P(X)}$

$P(c|x_1, x_2, x_3, \ldots, x_n) = \frac{P(c)P(x_1,x_2,x_3,\ldots,x_n|c)}{P(x_1,x_2,x_3,\ldots,x_n)}$ …….. (4)

The posterior probability may therefore be expressed as follows using (4):

$P(c|x_1, x_2, x_3, \ldots, x_n) = \frac{P(c)\prod_{i=1}^{n} P(x_i|c)}{P(x_1,x_2,x_3,\ldots,x_n)}$

Since $P(x_1, x_2, x_3, \ldots, x_n)$ is constant,

$P(c|x_1, x_2, x_3, \ldots, x_n) = P(c)\prod_{i=1}^{n} P(x_i|c)$

$\hat{c} = \text{argmax}\,(P(c)\prod_{i=1}^{n} P(x_i|c))$ …… (5)

Equation (5) finds the argument that produces the highest value from a target function. The assumptions made about the distribution of P(xi |c) are what distinguishes the various naive Bayes classifiers. There are different types of Naïve Bayes classifiers. We select Gaussian Naïve Bayes for our classification because it is most suitable for our dataset than others. GaussianNB is a classification algorithm that uses the Gaussian Naive Bayes algorithm. The features' probability is considered to be Gaussian:

$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma^2_c}}\exp\left(-\frac{(x_i-\mu_c)^2}{2\sigma^2_c}\right)$

Maximum likelihood is used to estimate the parameters σ_c and μ_c.

iv. Support Vector Machine

Support Vector Machine (SVM) is a numerical classifier that determines the optimum line or decision boundary to maximize the margin between classes and divides ndimensional space into classes so that new data points can be effectively placed in the right classification [16, 17]. When training a support vector machine, the goal is to find the decision boundary or separation hyperplane that leads to the biggest margin and best divides the levels of the target feature. The samples in a training dataset that lie along the margin extents and therefore establish the margins are referred to as support vectors, and they determine the decision boundary. So, support vectors are those data points that are located near the hyperplane.

Assume that,

Training set T = $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$

T = $\{x_i, y_i\}_i^n$; $1 \leq i \leq n$

Here, x_i is the input vector y_i is class label for i^{th} training data and n is the number of training data. The point above or on the hyperplane is class +1, whereas the point below the hyperplane is class -1.

$$h(x_i) = \begin{cases} +1, & \text{if } w.x + b \geq 0 \\ -1, & \text{if } w.x + b < 0 \end{cases}$$

$$y = \text{sign}\left(\sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + \beta\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

$$\min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{j=1}^{n} \alpha_j \quad \ldots\ldots\ldots\ldots \quad (7)$$

SVM follow the decision function of equation (6). Here, α and β are parameters where α = {α1 , α2 , … , αn } and K is kernel function. The kernel function receives data as input and converts it into the appropriate form for processing. It is implemented to each data instance in order to map the original non-linear state to a higher-dimensional space where they are divisible. The SVM objective function (apart from being convex) is quadratic. In equation (7) α minimize the objective function.

v. Voting Classifier

A voting classifier is one of the most powerful ensemble methods trained in various machine learning models and predicts an output as class based on the maximum probability of their elected class [17]. Voting classifier is an effective approach that can be useful when a single method exhibits bias towards a certain factor. When we have doubts about a certain machine learning model, the voting classifier is an excellent choice.

There are two ways to vote in a voting classifier: Hard Voting & Soft Voting. The predicted output class in hard voting is the class with the largest majority of votes — in other words, the class with the highest chance of being predicted by each classifier. The output class in soft voting is the prediction based on the average of the probabilities assigned to that class.

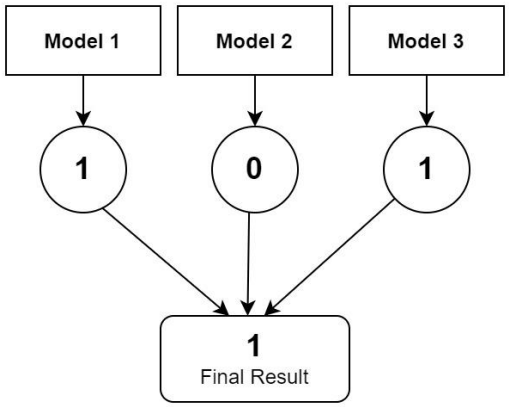


Figure 3.4: Hard Voting

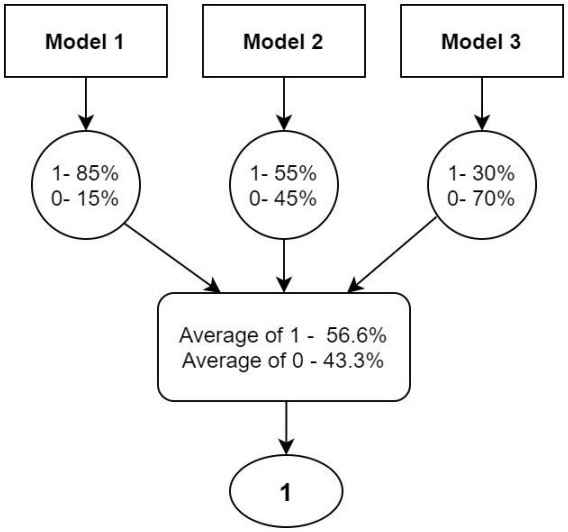


Figure 3.5: Soft Voting

3.5 Design, Implementation, and Simulation

The overall workflow of the proposed architecture is illustrated in Figure 3.1. All the mentioned steps of the prototype are implemented using Python. The random forest classifier is implemented using scikit-learn [21]. Also, for additional calculation, implementation, and support, Numpy [19] is used. The visual evaluation reports are generated using Matplotlib The dataset used to test the architecture is directly inserted, and no variations or selections were made while testing the architecture.

3.6 Summary

This section explains the architecture of the proposed machine learning-based job recommendation system. The overall architecture uses the voting classifier as the base classifier of the features as well.

Implementation, Testing, and Result Analysis

4.1 Introduction

In this chapter, the proposed architecture is tested and analyzed. This section includes the system configuration that was completed. This section outlines the assessment criteria used to measure the correctness of the result while also providing a full analysis of the outcome.

4.2 System Setup

The procedures are carried out with the assistance of Python. The NumPy [19], Pandas [20], and Scikit-Learn [21] libraries are used to conduct calculations and build machine learning models. The dataset contains information about CVs data, and our proposed architecture extracts jobs information by directly using the raw data.

Classifier	**Parameters**
Random Forest	n_estimators = default max_depth = 120 min_samples_split = 5
Gaussian Naïve Bayes	default
KNN	n_neighbors = 3
SVC	gamma = ‘auto’
Voting Classifier	estimators = [SVC, RandomForestClassifier, KNeighborsClassifier, GaussianNB], voting = ‘hard’

Table 4.1: All Classifier Parameter

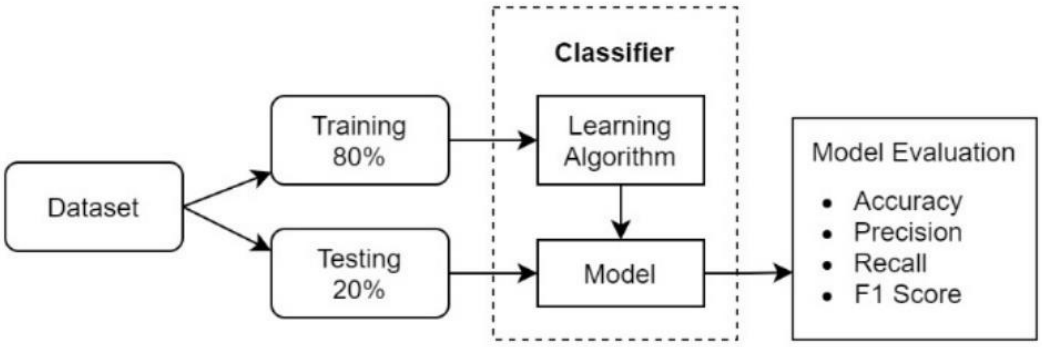


Figure 4.1: Classification Workflow

Table 4.1 represents the parameters of all the classifiers which we used to create the model. Figure 4.1 shows the architecture of the classification workflow. The dataset is divided into two parts, Training, and Testing. 80% data is set for Training and 20% for Testing. The test data is never augmented in the experiment for keeping the testing data authentic. Training data has been augmented.

4.3 Evaluation

To determine how better an algorithm or technique is, relative and shareable performance indicators are necessary. Model evaluation is the process of quantifying the accuracy of a system's predictions. To do so, we assess the newly trained model's performance on a new and unrelated dataset. Most of the performance metrics are based upon the confusion matrix, which consists of true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) values. When our model makes classification predictions, these four potential outcomes are possible. The importance of these aspects varies depending on how the performance review is conducted. Model evaluation performance indicators show us:

- How well our model performs
- Is our model precise enough to go into production?
- Will a larger training set increase the performance of our model?

4.3.1 Accuracy

The accuracy of a recommendation system may be defined as the number of right predictions the model calculates from the overall estimations of the model. Accuracy is one metric for measuring classification models; it is the percentage of correct predictions made by our model. The accuracy is measured as,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

4.3.2 Precision

Precision is defined as the ratio of correct outcomes to all positive results. The capacity of a classification model to identify just about the relevant data points is referred to as precision. The accuracy is measured as,

$$\text{Precision} = \frac{TP}{TP+FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

4.3.3 Recall

Recall is defined as the proportion of positive instances in relation to the total number of positive examples. As a result, the denominator (TP + FN) here represents the actual number of positive cases in the collection.

$$\text{Recall} = \frac{TP}{TP+FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

4.3.4 F1-score

The F1-score combines a classifier's accuracy and recall into a single metric by calculating their harmonic mean. It is the weighted average of accuracy and recall values ranging from 0 to 1, with 1 being the optimal F-score value.

$$\text{F1-score} = \frac{2*Precision*Recall}{Precision+Recall} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ldots (11)$$

4.4 Results and Discussion

The classification model's accuracy is defined as the percentage of accurate predictions it makes. Before and after data augmentation, we have found different accuracy in different classifier models. In this case, some models performed well in our dataset, and some performed comparatively less. Random Forest, KNN, SVM, and Voting Classifiers performed well but Naïve Bayes performed less than that.

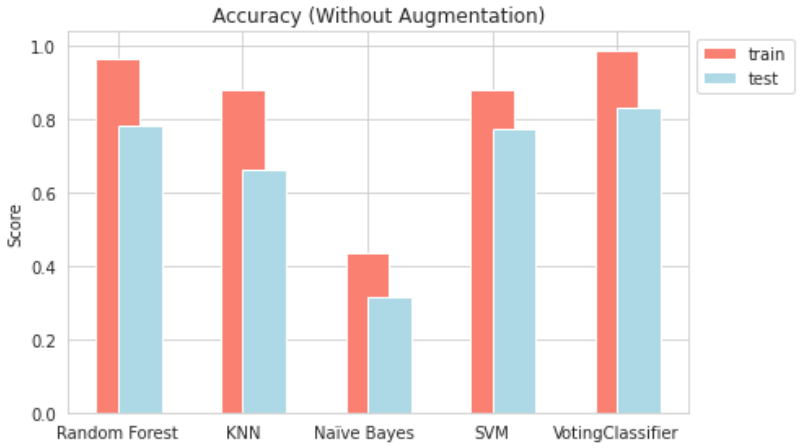


Figure 4.2: Accuracy without Augmentation

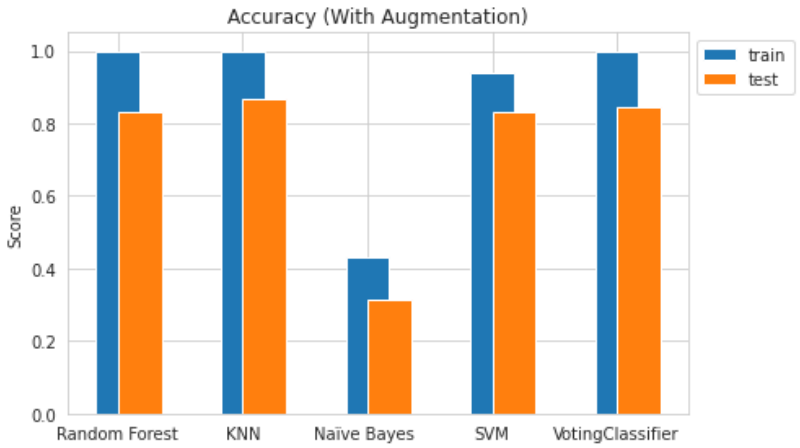


Figure 4.3: Accuracy with Augmentation

Figure 4.2 & 4.3 compare training and testing accuracy of different classifier models before and after data augmentation. Before data augmentation, the testing accuracy of Random Forest, KNN, Naïve Bayes, SVM, and Voting Classifiers respectively 78%, 66%, 31%, 77%, and 81%. After data augmentation, the testing accuracy is respectively 86%, 89%, 31%, 82%, and 89%. Here we can see that the Naïve Bayes classifier did not perform well. Naïve Bayes classifier heavily relies on data. It makes the assumption that all attributes are independent of one another. In practice, achieving a collection of predictors that are entirely independent of one another is almost impossible. It performs well in text classification. This is why the Naïve Bayes classifier did not perform well on our dataset.

We received the highest accuracy in the voting classifier both before and after data augmentation. Because voting classifier is a model that is trained in a combination of various classifier models and predicts an output based on the maximum probability of their chosen class, so even if one model gives a wrong prediction, there is a high probability of getting the correct output through the voting system, resulting in greater accuracy than other models. We have applied voting classifier using Random Forest, KNN, SVM, and Naïve Bayes classifier.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Random Forest	0.86	0.73	0.86	0.73
KNN	0.89	0.79	0.83	0.81
Naïve Bayes	0.31	0.45	0.45	0.43
SVM	0.83	0.76	0.86	0.75
Voting Classifiers	0.89	0.84	0.83	0.83

Table 4.2: Performance Evaluation

TABLE 4.2 shows the comparative performance evaluation after data augmentation of all classifiers. If we pay attention to the accuracy & F1-score, we can conclude that

Random Forest and Voting Classifier work better than the other classifier. Though, the accuracy of all classifiers can be increased by increasing more data.

4.5 Summary

According to the assessment reports, this architecture performs the most pleasant job recommendation tasks.

Standards, Constraints, and Milestones

This chapter illustrates the thesis work's Standards, Impacts, Ethics, and Challenges. Next, the Constraints and Alternatives are shown. Finally, the proposed work's Schedules, Tasks, and Milestones are given.

5.1 Standards (Sustainability)

We assure you that our thesis work will last for many years. The job recommendation system has recently been a significant study area. Job recommendation systems can benefit society and the government. Furthermore, the Machine Learning method that we employed for implementation is a cutting-edge deep learning approach. Our utilized resources will be available for more extended periods, so our thesis work will be sustainable.

5.2 Impacts on Society

The job recommendation system has a wide area of impact on the usage of the system. In the job recommendation system, different candidates have various educational qualifications, experience, and skills; each shows a suitable job based on their academic qualifications, experience, skills, and descriptions of related subjects. As a result, this can be implemented at a national level government job post and private organization job post. Implementing a job recommendation system that recommends appropriate jobs to the applicants based on a specific qualification to succeed in this issue.

5.3 Ethics

The job recommendation system has a significant level of use, depending on the data applied to prepare the model. The usage of job recommendation systems must maintain individuals' privacy concerns and should not be used for any purpose that

raises a national or social security threat. The usage, along with the dataset gathering, must be performed under the code of ethical principles.

5.4 Challenges

Even though current job recommendation system technologies are continuously expanding, the organizations creating such systems continue to encounter information security issues. This thesis study clearly shows that the current job recommendation systems are on the verge of being secured. Job recommendation are protected against hacker assaults, including data privacy.

5.5 Summary

Nevertheless, it should be noted that a job recommendation system can be a well-suited supplement for HR. This system will not waste the time of both the candidate and the organization. The contribution of this study includes the implemented job recommendation system that is entirely accurate and efficient for both the candidates and recruiters.

5.6 Design Constraints

The proposed architecture of the overall structure may be executed based on ML, which has two types of constraints: soft constraints, which are imposed by adding extra penalties to the loss function, and hard constraints, which are requirements that must be met while building the model. The model requires devices with high processing capability to perform predictions. The model does not require any GPU support.

5.7 Component Constraints

The component requirements of the proposed architecture include,

- Minimum Processor Requirement: Intel i3 (7th Gen, 3GHz)
- Minimum Memory Requirement: 4GB (DDR3, 1600 bus)

5.8 Budget Constraints

The anticipated budget is to be determined by the current market price of the required components.

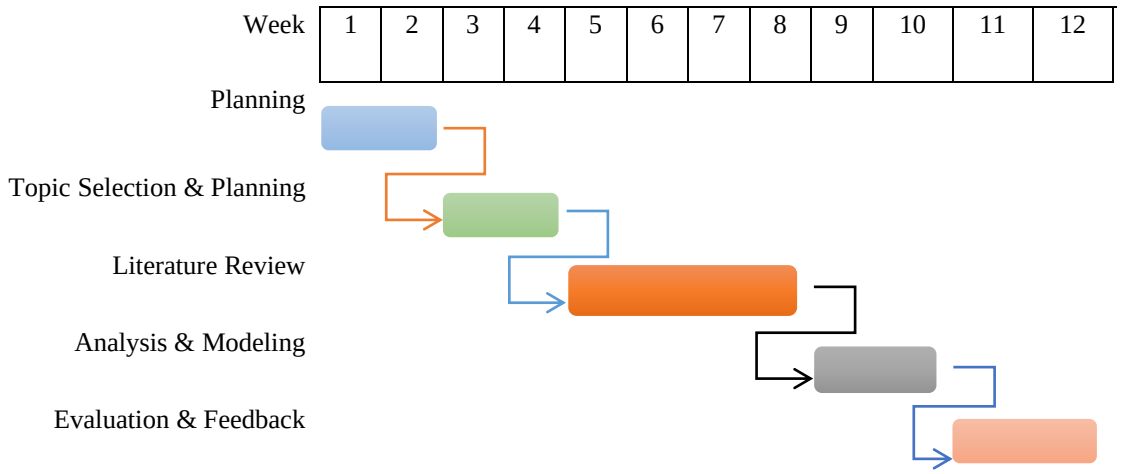
5.9 Timeline

The whole timeline of the thesis work can be divided into three sections based on our supervisor's work execution procedure's three semesters. The planning and assessment of thesis-related works is part of the first-semester work process. The second-semester work process involves collaborative effort on prototype design and prototype analysis. We developed and tested the overall design in the third semester and reported on the overall workflow.

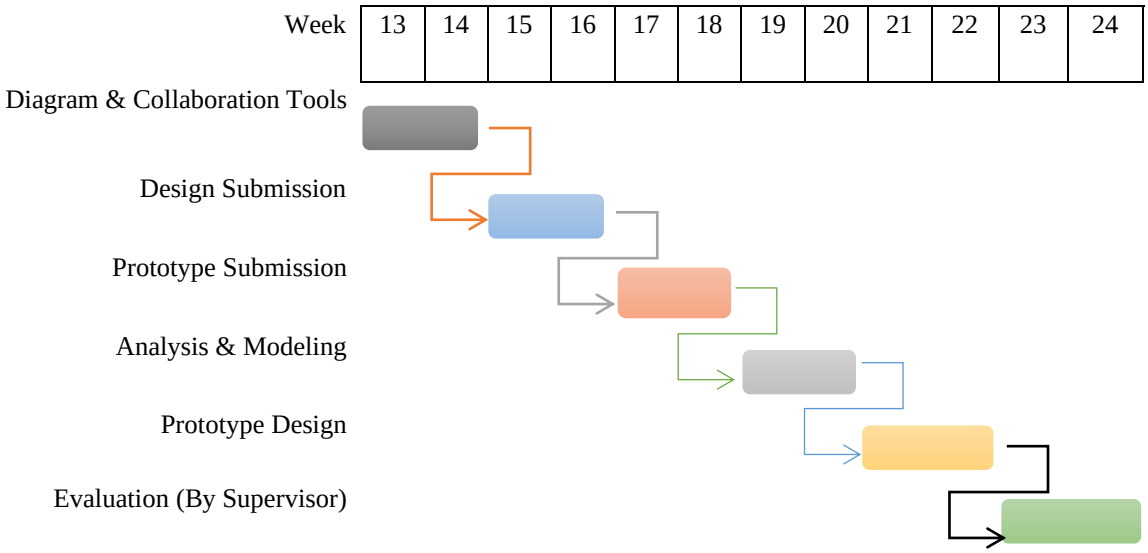
5.10 Gantt Chart

Figure 7.1 is a Gantt chart detailing the thesis work execution process. The entire duration of the thesis work is three semesters, with each semester lasting twelve weeks.

1st Semester



2nd Semester



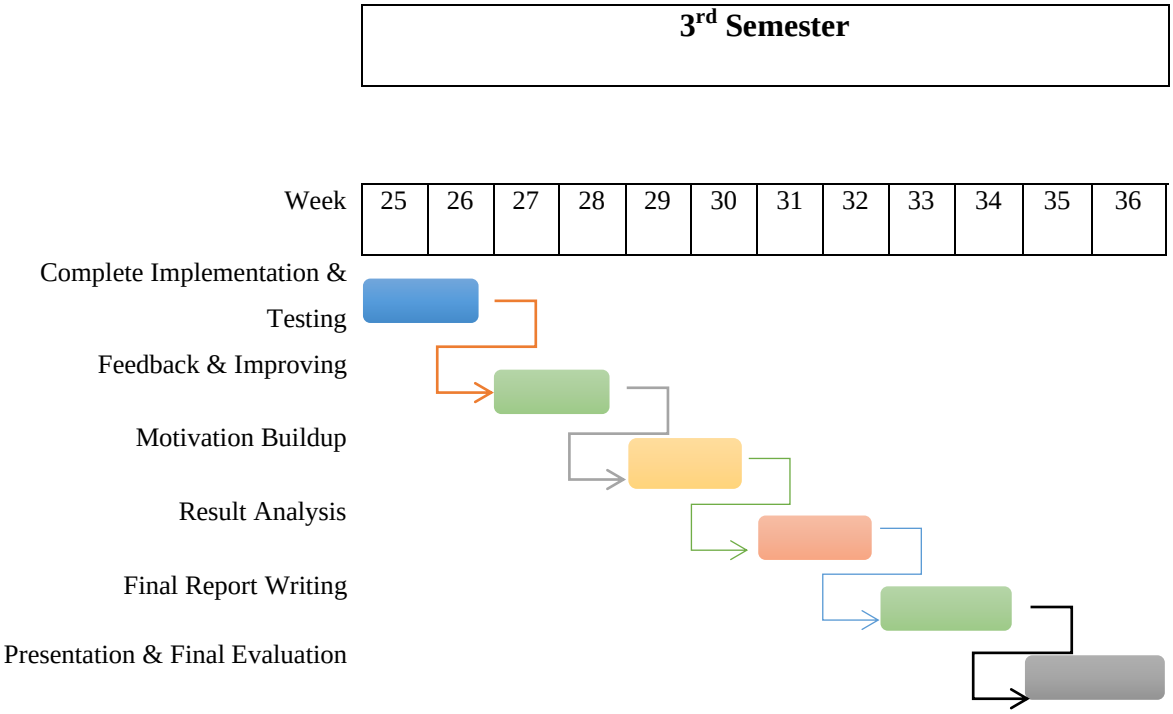


Figure 7.1: Gantt chart of the work execution process.

5.11 Summary

The avoidance of machine learning in the proposed job recommendation system can be implemented in lightweight devices that are cost-efficient and available.

Conclusion

6.1 Introduction

The paper inaugurates an automated job recommendation system that can assist both fresher and experienced candidates in getting the appropriate occupation. Data have been collected to testify various machine learning models. The research endeavor embraces preprocessing tasks, including normalization and augmentation with the collected dataset, harnessing processed data output using few classifiers. Machine learning methods have been conducted in this research work. Comparing the outcomes of the models, by acquiring the Voting Classifier model as the possible, leading consequence to recommend an acceptable profession with an accuracy rate of 89% superior performance. However, this accuracy can be escalated furthermore. The system dispenses secured, and privacy safeguarded as we deploy less information of personage. We are also optimistic that our contribution will succor future researchers to adopt new tactics and accomplish a robust system in the swathe of job recommendation research function.

6.2 Future Works and Limitations

As the number of users grows, the algorithms suffer scalability issues. If we have 100 million resumes and 100,000 jobs, the system will create a sparse matrix with one trillion elements. These are called scalability issues. We tend to solve this particular challenge of the proposed architecture. We are also forethought of testing our approaches with a much bigger dataset and comparing the affluent performance between Machine Learning and Deep Learning in our subsequent research.

References

[1] Rafter, Rachael, Keith Bradley, and Barry Smyth. "Personalised retrieval for online recruitment services." In The BCS/IRSG 22nd Annual Colloquium on Information Retrieval (IRSG 2000), Cambridge, UK, 5-7 April, 2000. 2000.

[2] Malinowski, Jochen, Tobias Keim, Oliver Wendt, and Tim Weitzel. "Matching people and jobs: A bilateral recommendation approach." In Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), vol. 6, pp. 137c-137c. IEEE, 2006.

[3] Lee, Danielle H., and Peter Brusilovsky. "Fighting information overflow with personalized comprehensive information access: A proactive job recommender." In Third International Conference on Autonomic and Autonomous Systems (ICAS'07), pp. 21-21. IEEE, 2007.

[4] Hutterer, Matthias. "Enhancing a job recommender with implicit user feedback." PhD diss., 2011

[5] Kenthapadi, Krishnaram, Benjamin Le, and Ganesh Venkataraman. "Personalized job recommendation system at linkedin: Practical challenges and lessons learned." In Proceedings of the eleventh ACM conference on recommender systems, pp. 346-347. 2017.

[6] Singh, Amit, Catherine Rose, Karthik Visweswariah, Vijil Chenthamarakshan, and Nandakishore Kambhatla. "PROSPECT: a system for screening candidates for recruitment." In Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 659-668. 2010.

[7] Hong, Wenxing, Lei Li, Tao Li, and Wenfu Pan. "iHR: an online recruiting system for Xiamen Talent Service Center." In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1177-1185. 2013.

[8] Rajath V , Riza Tanaz Fareed , Sharadadevi Kaganurmath, 2021, Resume Classification and Ranking using KNN and Cosine Similarity, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 08 (August 2021).

[9] Hu, Biyun, Zhoujun Li, and Wenhan Chao. "Data sparsity: a key disadvantage of user-based collaborative filtering?." In Asia-Pacific Web Conference, pp. 602-609. Springer, Berlin, Heidelberg, 2012.

[10] Hong, Wenxing, Siting Zheng, Huan Wang, and Jianchao Shi. "A job recommender system based on user clustering." J. Comput. 8, no. 8 (2013): 1960-1967.

[11] Paparrizos, Ioannis, B. Barla Cambazoglu, and Aristides Gionis. "Machine learned job recommendation." In Proceedings of the fifth ACM Conference on Recommender Systems, pp. 325-328. 2011.

[12] Yu, Hongtao, Chaoran Liu, and Fuzhi Zhang. "Reciprocal recommendation algorithm for the field of recruitment." Journal of Information & Computational Science 8, no. 16 (2011): 4061-4068.

[13] Buettner, Ricardo. "A framework for recommender systems in online social network recruiting: An interdisciplinary call to arms." In 2014 47th Hawaii International Conference on System Sciences, pp. 1415-1424. IEEE, 2014.

[14] Gupta, Anika, and Deepak Garg. "Applying data mining techniques in job recommender system for considering candidate job preferences." In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1458-1465. IEEE, 2014.

[15] Malinowski, Jochen, Tim Weitzel, and Tobias Keim. "Decision support for team staffing: An automated relational recommendation approach." Decision Support Systems 45, no. 3 (2008): 429-447.

[16] Roy, Pradeep Kumar, Sarabjeet Singh Chowdhary, and Rocky Bhatia. "A Machine Learning approach for automation of Resume Recommendation system." Procedia Computer Science 167 (2020): 2318-2327.

[17] Appadoo, Kevin, Muhammad Bilaal Soonnoo, and Zahra Mungloo-Dilmohamud. "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing." In 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1-6. IEEE, 2020.

[18] Pessach, Dana, Gonen Singer, Dan Avrahami, Hila Chalutz Ben-Gal, Erez Shmueli, and Irad Ben-Gal. "Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming." Decision Support Systems 134 (2020): 113290.

[19] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011).

[20] McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." Python for high performance and scientific computing 14, no. 9 (2011): 1-9.

[21] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.