ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

Automated Resume Screening System Using Machine Learning

Kumarisravel S.[1]; Dr. J. Lysa Eben[2]

[1]Student, [2]Assistant Professor

[1,2]Department of Data Science, Madras Christian College, Chennai, India

Publication Date: 2026/04/08

Abstract: The manual process in recruitment is too hard when the data is huge and complex. This increase in data can bring inaccuracy in selecting a candidate based on the job description. This project presents the involvement of Machine learning techniques in the recruitment process such as ranking of candidate's resumes, recommendation of JD (Job description) to candidates and recommendation about resumes to recruiters and recommendation about candidate's resumes to recruiters. This system increases accuracy, consistency, and speed in the recruitment process. Most of the common processes in recruitment got automated in this system. Techniques like TF-IDF, Cosine similarity and Classification models are used to achieve the features of classification, recommendation and ranking of resumes. TF-IDF vectorization usually converts textual data like resumes or JD (Job description) into numerical vectors, then Cosine similarity is used for providing ranking and recommendation to recruiter and candidates. Models that have higher performance in accuracy, precision, recall and F1-Score are used for resume classification features. This project carries a Dataset from Kaggle, which has suitable and required data for the process like resumes with candidate skills, communication address and experience. Later, using the taken Dataset, we train models like Logistic regression, Support vector machine, Random forest and Naive Bayes to find out which mode fits best in the classification feature.

Keywords: *Machine Learning, Job Description, Resumes, Kaggle, Random Forest, TF-IDF, Cosine-Similarity, Recommendation, Ranking and Classification.*

How to Cite: Kumarisravel S.; Dr. J. Lysa Eben (2026) Automated Resume Screening System Using Machine Learning. *International Journal of Innovative Science and Research Technology*, 11(4), 39-44. https://doi.org/10.38124/ijisrt/26apr374

I. INTRODUCTION

The process of recruitment will be hard if a huge amount of improper data is present in job descriptions and resumes. This irrelevant data presence makes the recruiter not to be sure about "is it the candidate whom he is expecting?.

This can be avoided by involving Machine Learning techniques; this also increases efficiency in performance. Techniques from this Machine learning technology like TF-IDF, Cosine similarity finds the similarity score in between Job description and Candidate's resume; this matching score in between these two topics can be used in Ranking and Recommendation of resumes to recruiter and Job description to Candidates. Also, approaches like Random forest models help in categorizing resumes based on posted jobs or matching skills or based on provided keywords.

➢ *Proposed System*

The proposing system has the feature of recommendation of Job description and resumes, ranking of resumes for recruiters, classification of resumes based on matching key from Job description and resumes which will avoid the manual process and human interaction in the system which brings bias in system.

For this feature, the system uses the techniques from Machine Learning such as TF-IDF for making textual data to numerical vectors, cosine-similarity for finding similarity in between Job description and resumes from candidates. Also, the model such as Random Forest is used for resume classification. For this classification process, the model gets training from a dataset collected from Kaggle. The dataset will have details of resumes and job descriptions.

➢ *System Feature*

The Automated Resume Screening System has the feature of below:

- Recommending of resumes to Recruiters based on posted Job description (JD)
- Recommending Job description (JD) to a candidate based on their matching Skills, experience and qualification from their resumes.
- Ranking of candidate resume based on available keyword in resume, which is matching with Job description.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

- Classification of resumes to recruiters based on matching skills found in Job description.

II. LITERATURE REVIEW

There are many researchers who worked on automating the process of recruitment using machine learning techniques.

Kavita Ganesan (2019) explained in what way Natural Language processing technology can be applied and used in extracting important information from textual data like resumes. The explained topic also had concepts such as tokenization and TF-IDF for text representation [9].

For Candidate's resume classification, S. K. Lakshmanan et al (2020). also proposed a machine learning-based approach using models such as Naive Bayes and Support Vector Machine; this proposed work improved higher efficiency in selecting a candidate [4].

A. K. Sharma and R. Gupta (2021) proposed and developed a system that uses TF-IDF and Cosine similarity for matching candidate's resumes with recruiters' Job description (JD). The proposed system brings the topic about similarity-based concepts that can improve candidate-job matching [3].

P. K. Singh et al. (2022) implemented and presented a recruitment system with the help of Random Forest and Logistic Regression for classification. The results of their presented system bring a topic of this ensemble models can perform better accuracy compared to traditional methods [5].

But most of the mentioned studies and presentations only focus on resume classification and depend only on keyword-based matching. They do not provide a complete system that includes ranking and recommendation of candidates.

The mentioned topics bring a requirement for an automated resume screening system that has classification, similarity-based ranking and recommendation to improve the overall recruitment process [1][2].

III. METHODOLOGY

Some of the important problems prevail nowadays in the recruitment process:

- Classifying resumes for recruiters like whether it is suitable, not suitable and moderate.
- How well his/her resumes fit.
- Suggestion / recommending of resumes for recruiters based on given JD.
- Suggestion of JD to candidate based on the matching skills.

The mentioned problems drive the recruitment process in choosing unfit candidates, loss in revenue, higher time consumption in retaining candidates and lack of results. Machine learning approaches are used in this system to solve the mentioned issues. The project uses methods like TF-IDF vectorization for text representation, Cosine similarity for finding how similarity resumes and JD gets matched and classification algorithms for categorizing Candidate based on suitability [8].

➢ *Dataset Description*

Kaggle dataset is used for this resume screening process. The taken dataset has details like JD (Job details) and candidate resumes. It is a synthetic dataset. The taken dataset is then used in training the classification models such as Logistic regression, Naive Bayes to classify candidates resumes into suitable, not suitable and moderate. Also, the system uses the approach of TF-IDF vectorization to convert textual data such as JD (Job description) and Resumes to numerical vectors for finding the similarity between them using Cosine-similarity and then deals with ranking and recommendation.

➢ *Data Preprocessing*

The taken dataset will have the unstructured textual data with noise like special characters, irrelevant words, inconsistent formatting. So, we perform preprocessing, which standardizes the data for extracting the feature and machine learning model training.

Preprocessing generally has this below process.

- *Text cleaning:*

Here, the resumes are cleaned by removing special characters, punctuation marks, numbers and extra spaces to reduce noise in the dataset.

- *Lowercase:*

All data of resumes will be converted into lowercase to maintain uniformity in this process. Example : Name, experience, address generally will get converted to lowercase.

- *Tokenization:*

It mentions maintaining necessary words instead of keeping all words. Example: maintaining relevance of objectives of the candidate only with important words rather than maintaining all content he mentions.

- *Stop word removal:*

This process removes the words that do not provide any meaning.

- *Lemmatization / Stemming:*

This process makes the words to be reduced to their root form. Example: Lemmatization of the word 'running' will be 'run'.

- *Handling missing data:*

Removing missing data from dataset like address of candidate or skills of candidate. This removal process increases the performance of the model.

➢ *Feature Extraction*

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

This process usually converts text into numerical forms. TF-IDF techniques are the most commonly used methods for feature extraction. Here, Resumes and Job description (JD) from the taken Kaggle dataset get converted into numerical vectors [9].

➢ *Model Building*

After the process of TF-IDF conversion, the dataset split into training and testing sets in an 80:20 ratio. Machine Learning models like logistic regression, Naive Bayes, and Random forest. etc, are used to classify resumes into categories like suitable, not suitable, and moderate.

The models are trained using the trained dataset and tested using the tested dataset to check its performance. Once the training gets done for the model, the model will be used for classification's prediction of new resumes.

- *Example:*

Training of java developer resumes with labels of suitable, not suitable and moderate for a classification model, letting the model predict by providing new resumes for suitability of the resumes.

➢ *Model Training Process*

In the learning process, the model learns the relationship between resume content and their appropriate job description. Every resume is given a label like suitable, not suitable or moderate.

- *Example:*

A resume that has all required skills for a python developer will be labeled as suitable. The model usually gets TF-IDF vectors as input, while the output will be pre-defined categories such as suitable, not suitable and moderate.

Once the training process gets done, the chosen model can classify new resumes into these categories: suitable, not suitable and moderate. The trained model is further used in the system for ranking, recommendation and resume screening. In this system, for classification purposes, the model like logistic regression, Naive Bayes, Random forest etc. is trained using different resumes and labels like suitable, not suitable and moderate.

After the training completion, the taken model will be tested using test data, and then it will predict the real time data.

➢ *Resume Classification*

The models such as Random Forest, Logistic regression, and Naive Bayes are used to perform classification processes. The taken model gets training by labels such as suitable, not suitable and moderate to classify the resumes. During this training process, the model will know about the pattern between resumes and its appropriate label.

When a new resume is uploaded, that will be transformed into TF-IDF vectors and passed to a trained model, then the model predicts proper labels for that resume and shows them to the recruiter.

➢ *Similarity Matching*

This process compares the two vector data like Resume vector and Job description (JD) vectors and finds the similarity score between them. For this process, generally cosine similarity is used.

Based on found matching scores, the system functions for Ranking and recommendation for resumes and resumes [9].

➢ *Ranking System*

Using similarity scores found in between job description vector and Resumes vector, this process for Ranking. Cosine similarity technique is applied to find the score of similarity. Similarity score high is meant for the best match.

➢ *Recommendation System*

- *Resume recommendation to Recruiters:*

Based on similarity score, the resume that has higher similarity score will be recommended to Recruiters automatically. For this process, Cosine similarity used to find its similarity score between Resume vectors and Job description vectors.

- *Recommendation of resume to Candidate:*

Based on similarity score, the Job description (JD) which has higher matching score will get recommended to Candidate.

➢ *Evaluation Metrics*

The evaluation metrics like Accuracy, Precision, Recall and F1-Score are usually used to test the performance of the using model, which also helps in selecting correct and proper classification models [8].

IV. SYSTEM DESIGN

The proposed system is generally used to automate the recruitment process by annually using candidate resumes and job descriptions using the technology of machine learning. This reduces major effort and improves accuracy of candidate selection through classification, ranking and recommendation.

Figure 1 shows the Architecture of the proposed system.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

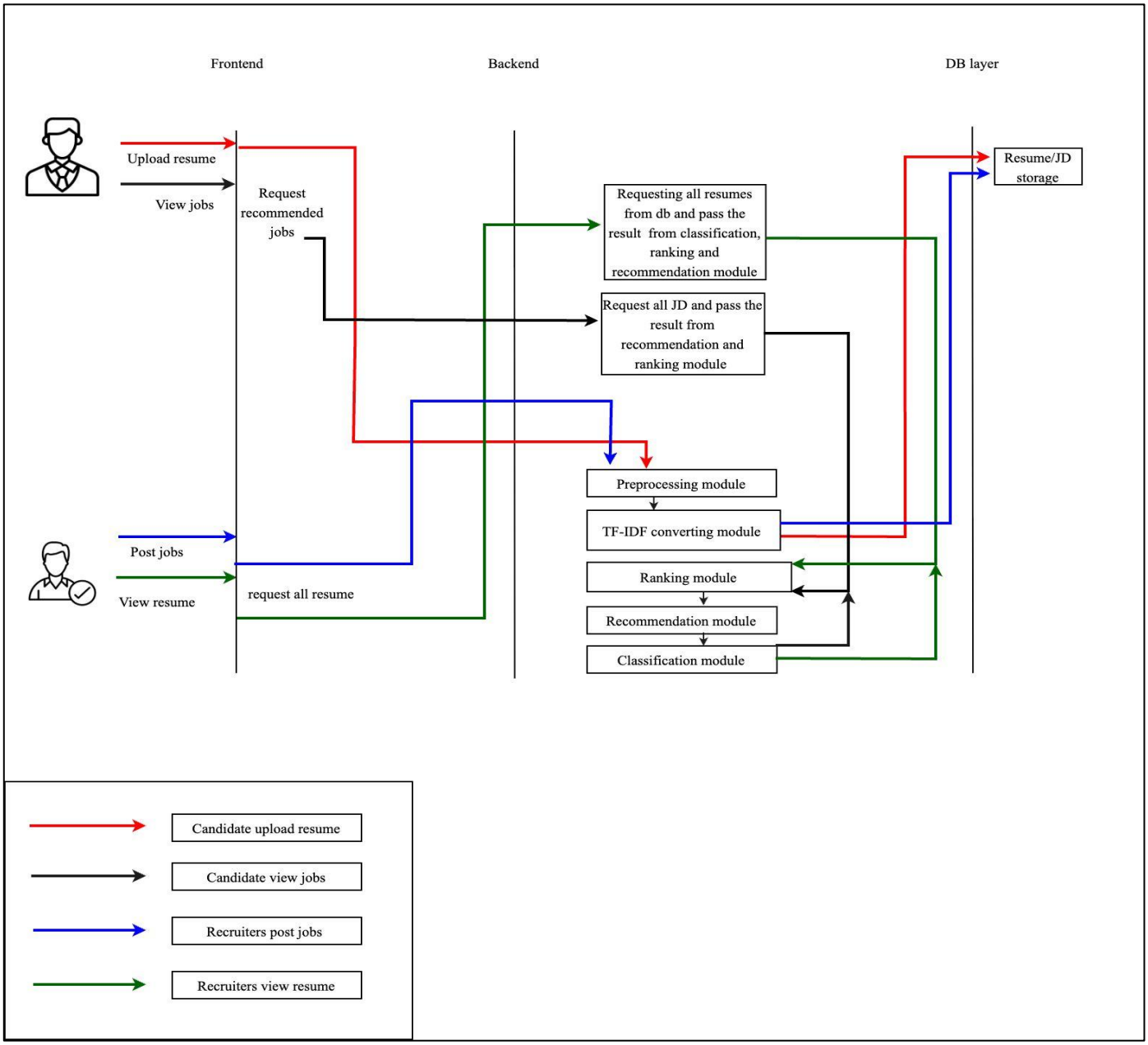

Fig 1 System Architecture

V. IMPLEMENTATION DETAILS

Technologies such as Django, SQLite, HTML, CSS, and JS in this app designing process. It is designed like a Web based application. There are two roles involved in this system such as Recruiters and Candidates.

The recruiter has the options of login, sign up, post JD (Job description), view applied Candidate, ranking of candidate's resume, classification and recommendation of them. They also have the option of logout.

Likewise, the candidate has the option like login, sign up, view available jobs, and apply for the available jobs. The system will show tags like suitable, not suitable and moderate, which means the candidate has the option where he can see the jobs that are suitable for his/her skills automatically.

The recommendation of resumes to recruiter, recommendation of JD to candidate is done using Cosine similarity. Rank score for each resume at the recruiter side is done using Cosine similarity. Showing of classification of resume like suitable, not suitable and moderate is generally works by classification model that performs best in evaluation metrics.

For all processes like ranking, classification and recommendation, the textual data first does the process of preprocessing and then happens numerical vector conversation using the technique of TF-IDF.

SQLite database is used in case of any data storage in this system.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

VI. RESULTS AND DISCUSSION

➢ *Introduction*

Four machine learning models are involved in the evaluation process of this system: Logistic regression, Naive Bayes, Support Vector Machine (SVM) and Random Forest to classify resumes into the categories of suitable, not suitable and moderate. The performance of the system is measured using Accuracy, Precision, Recall and F1-score.

➢ *Model Performance Result*

The taken classification models are : Logistic regression, Naive Bayes, Support Vector Machine (SVM) and Random Forest. The TF-IDF vectors of resume data are used in the training process of taking models.

The taken models have performed as below:

- Logistic Regression performed an accuracy of 64.58%, precision of 67.79%, recall of 64.58%, and F1-score of 64.15%.
- Naive Bayes performed an accuracy of 56.94%, precision of 62.12%, recall of 56.94%, and F1-score of 54.80%.
- SVM performed an accuracy of 59.95%, precision of 65.05%, recall of 59.95%, and F1-score of 60.40%.
- Random Forest performed the highest performance with an accuracy of 70.02%, precision of 69.65%, recall of 70.02%, and F1-score of 68.53%.

Table 1 shows the difference performance's comparison for taken models.

Table 1 Model Comparison

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	64.58%	67.79%	64.58%	64.15%
Naive Bayes	56.94%	62.12%	56.94%	54.80%
SVM	59.95%	65.05%	59.95%	60.40%
Random Forest	**70.02%**	**69.65%**	**70.02%**	**68.53%**

➢ *Model Comparison*

The model Random Forest performs best from taking all models across all evaluation metrics. Logistic Regression performs moderate performance, while SVM performs a little bit lower. Naive Bayes performs lower than other models.

The accuracy comparison graph below makes sure the Random Forest model performs the best among the taken models for this system.

Figure 2 shows accuracy comparison of taken models.

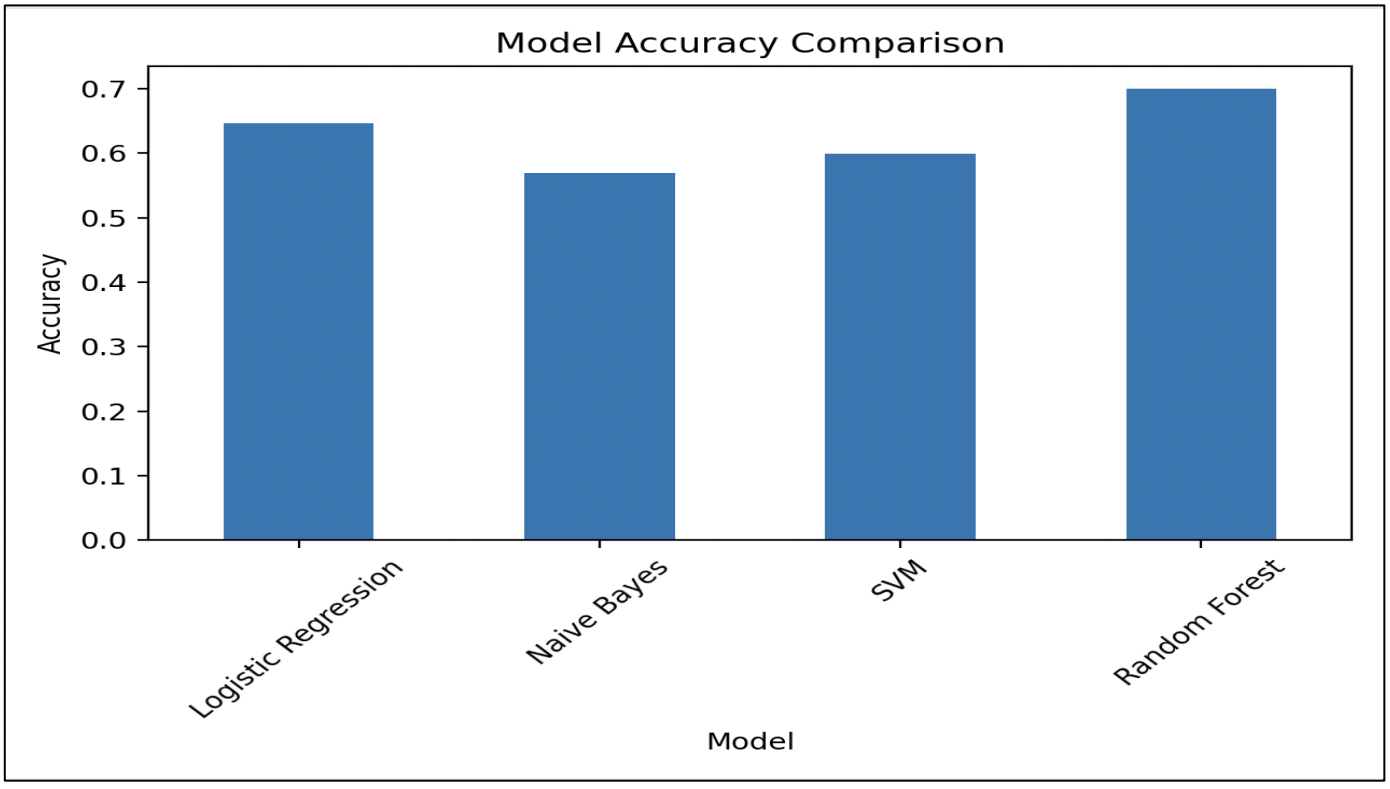

Fig 2 Accuracy Comparison Graph

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr374

➢ *Evaluation Metrics Analysis*

Evaluation measures like Accuracy, Precision, Recall and F1-score are used to estimate the model in general.

- Accuracy mentions whether all predictions by the system are correct or not.
- Precision mentions whether the system chooses all resumes are relevant or not.
- Recall mentions the measure about how many relevant resumes are correctly identified.
- Precision and recall are balancing the current system using F1-score.

The performance of Random Forest is best in all the mentioned metrics, mentioning again it is a better classification model and brings more reliable predictions compared to other models.

➢ *Ranking of Resumes*

After the classification process of candidate resumes, the ranking process can be done based on found keywords in the resume that are related to posted jobs by Recruiters. The system performs the feature of Ranking using Cosine similarity. Candidate's resumes with higher similarity scores are ranked higher; it enables recruiters to find the candidate resumes easily.

➢ *Recommendation of Resumes*

The system is designed with recommendation of Resumes to candidate for an easy selection process based on the posted jobs; also, it has the feature of recommendation of Job post to candidate based on their matching skills. This feature brings greater ease of communication between jobseekers and recruiters.

VII. CONCLUSION

The results show that the proposed system effectively automates resume screening. During the training process of the model, the chosen model avoids overfitting and also increases accuracy.

The approach of TF-IDF makes the textual process of Resume to be more effective. Classification is implemented and evaluated in this system; to make the system have higher processing, we have implemented Ranking and Recommendations.

FUTURE WORK

The advanced deep learning techniques can be integrated in the future, to improve the performance of accuracy and resume classification in the best way.

The system can be designed with semantic analysis with the ability to understand the context and meaning, so the system will not rely on keywords.

The other similarity techniques that increase the ability of ranking a candidate resume for a job description will be added to this system for improvement. The advanced approaches will be integrated for improving recommendation of resumes to recruiters and job description to candidates.

In addition, the system will be deployed into a real-time web application for public usage, which improves its practical usage.

REFERENCES

[1]. S. Malinowski, T. Keim, O. Wendt, and T. Weitzel, “Matching People and Jobs: A Bilateral Recommendation Approach,” Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006.

[2]. M. Paparrizos, B. B. Cambazoglu, and A. Gionis, “Machine Learned Job Recommendation,” Proceedings of the 5th ACM Conference on Recommender Systems, 2011.

[3]. J. Yi, J. Allan, and W. B. Croft, “Matching resumes and jobs based on relevance models,” Proceedings of the 33rd International ACM SIGIR Conference, 2010.

[4]. S. G. Raj, S. S. S. Kumar, and M. S. Reddy, “Resume Parsing and Classification Using Machine Learning,” International Journal of Engineering and Advanced Technology, 2019.

[5]. A. Sarkar, S. K. Saha, and S. Mitra, “Resume Classification Using Text Mining Techniques,” International Journal of Computer Applications, 2018.

[6]. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2011.

[7]. C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[8]. F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9]. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, 1988.

[10]. Kaggle Dataset, “Resume Dataset for Classification,” Available: https://www.kaggle.com