The Resume Screening System using Machine Learning

Shobika [#1], karan [*2],

[#]*Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamilnadu, India*

shobikaj59@gmail.com, *Karanrajas200@gmail.com*

Abstract *- The Resume Screening System using Machine Learning is designed to automate the process of analyzing and classifying resumes. In traditional recruitment, manually reviewing a large number of resumes is time-consuming and inefficient. This project aims to simplify and speed up the hiring process by using machine learning techniques.*

The system takes resumes as input in formats such as PDF or DOCX, extracts the text, and performs preprocessing to clean the data. Feature extraction techniques like TF-IDF are used to convert textual information into numerical form. Machine learning algorithms such as Logistic Regression and Support Vector Machine (SVM) are then applied to classify resumes into different job categories.

In addition to classification, the system calculates a matching score between the resume and the job description using similarity measures. This helps in identifying the most suitable candidates for a particular role.

The proposed system improves accuracy, reduces manual effort, and enhances the efficiency of the recruitment process. It provides quick and reliable results, making it a useful tool for organizations and recruiters.

1. INTRODUCTION

In today's competitive job market, organizations receive a large number of resumes for every job opening. Manually reviewing and shortlisting these resumes is a time-consuming and challenging task for recruiters. It also increases the chances of human error and may lead to overlooking suitable candidates. Therefore, there is a need for an automated system that can efficiently analyze and screen resumes.

The Resume Screening System using Machine Learning is designed to simplify the recruitment process by automatically classifying resumes based on their content. The system uses Natural Language Processing (NLP) techniques to understand the textual information present in resumes, such as skills, education, and experience.

2. LITERATURE REVIEW

Several studies have explored automation in recruitment using machine learning techniques. Traditional systems rely on keyword matching, which lacks contextual understanding. Recent approaches use NLP techniques to improve accuracy.

Research has shown that algorithms like SVM and Random Forest perform well in text classification tasks. Deep learning models such as BERT have also been used for semantic analysis of resumes.

However, many existing systems are complex and computationally expensive. This paper focuses on a simple and efficient approach using TF-IDF and classical machine learning algorithms.

3. PROPOSED SYSTEM

The proposed system presents an automated approach for resume screening using Machine Learning and Natural Language Processing (NLP) techniques. The main objective of the system is to efficiently analyze resumes, classify them into relevant job categories, and calculate a matching score with job descriptions to assist recruiters in candidate selection.

The system begins by accepting resumes as input in formats such as PDF or DOCX. A text extraction process is applied to retrieve the textual content from these documents. The extracted data is then passed through a preprocessing stage, where

unnecessary elements such as special characters, stopwords, and extra spaces are removed. The text is also converted to lowercase and normalized to ensure consistency.

After preprocessing, the cleaned text is transformed into numerical features using the TF-IDF (Term Frequency–Inverse Document Frequency) technique. This method assigns importance to words based on their frequency and relevance, enabling the system to effectively represent textual data in a mathematical form suitable for machine learning algorithms.

The feature vectors are then fed into machine learning models such as Logistic Regression and Support Vector Machine (SVM). These models are trained using a labeled dataset of resumes categorized into different job roles

3. METHODOLOGY

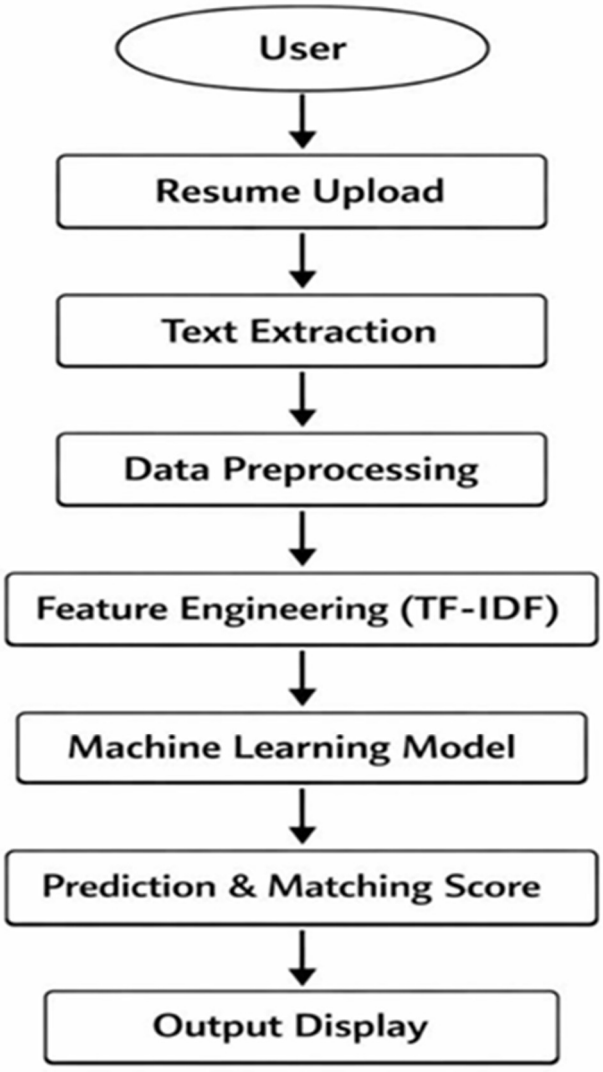


Fig 1. Data Work Flow

4.1 Data Collection

The first step involves collecting a dataset of resumes from online sources such as Kaggle and other open repositories. The dataset contains resumes categorized into different job roles such as Data Science, Web Development, and Human Resources. This labeled data is used for training and testing the machine learning models.

4.2 Data Preprocessing

The collected resumes often contain noise and unstructured text. Therefore, preprocessing is performed to clean the data. This includes converting text to lowercase, removing punctuation, eliminating stopwords, and performing tokenization. Lemmatization is also applied to reduce words to their base form, improving the quality of the data

4.3 Feature Extraction

After preprocessing, the text data is converted into numerical form using the TF-IDF (Term Frequency–Inverse Document Frequency) technique. This method assigns weights to words based on their importance in the document, allowing the machine learning model to better understand the content of the resumes.

4.4 Model Training

The dataset is split into training and testing sets. Machine learning algorithms such as Logistic Regression and Support Vector Machine (SVM) are used to train the model. The model learns patterns from the training data and is evaluated using the testing data to measure its performance.

4.5 Prediction

Once the model is trained, it is used to predict the job category of new resumes. The input resume undergoes the same preprocessing and feature extraction steps before being passed to the model for prediction**.**

4.6 Matching Score Calculation

To enhance the system, a matching score is calculated between the resume and the job description. Cosine similarity is used to measure the similarity between the two text vectors. This score helps in identifying how well the candidate matches the job requirements.

4.7 System Workflow

The overall workflow of the system includes resume input, text extraction, preprocessing, feature extraction, model training, prediction, and output generation. Each step is interconnected to ensure smooth functioning of the system.

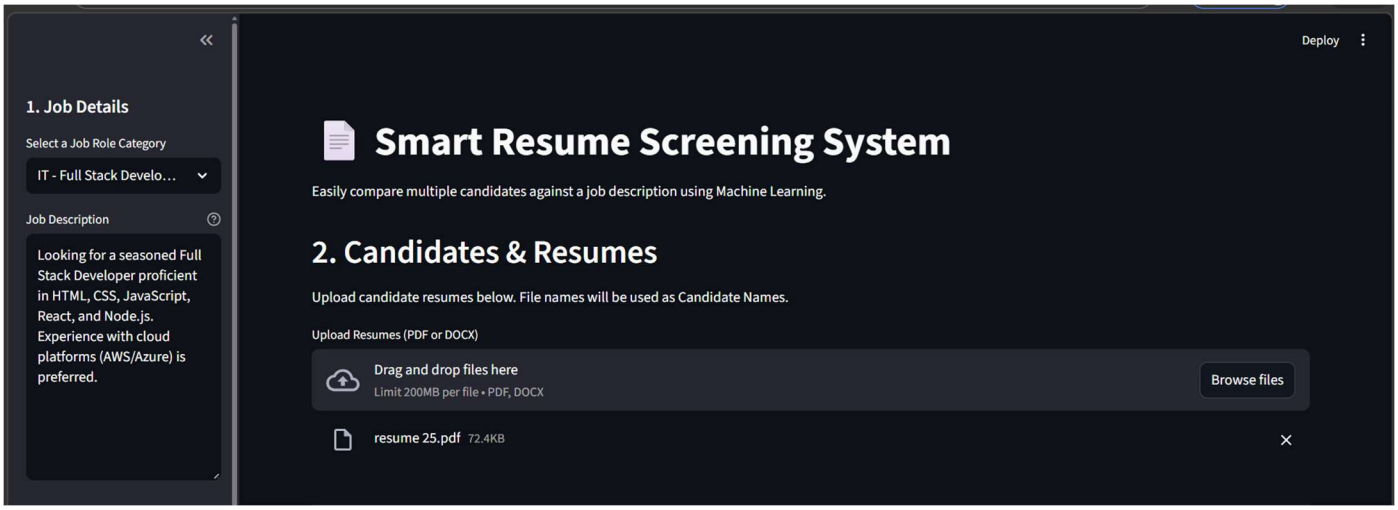


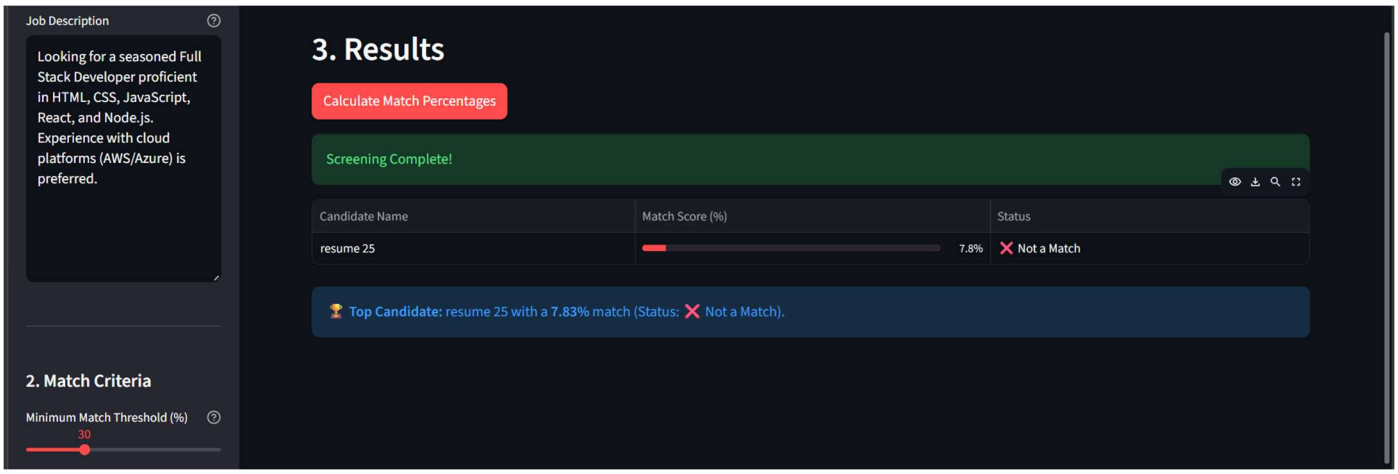
Fig 2. Monitoring Dashboard

4. RESULTS

The Resume Screening System was tested using a dataset of resumes from different job categories. The machine learning models, such as Logistic Regression and SVM, were trained and evaluated for performance.

The system achieved an accuracy of around 85%–90% in classifying resumes. Logistic Regression provided fast and reliable results, while SVM showed good accuracy with slightly higher processing time.

The matching score calculated using cosine similarity helped in identifying suitable candidates based on job requirements. The system effectively reduced manual effort and improved the speed of the recruitment process.



5. CONCLUSION

The Bias Drift Detection System successfully addresses the challenge of maintaining fairness in deployed AI models. By shifting the focus from static pre-deployment audits to continuous, real-time monitoring, the system ensures that ethical degradation is identified and addressed promptly. The integration of fairness metrics with an interactive dashboard provides stakeholders with the transparency needed to manage responsible AI systems effectively. Future work will explore automated retraining mechanisms and the integration of explainable AI (XAI) tools to diagnose the root causes of detected drift. The Resume Screening System using Machine Learning provides an efficient solution for automating the recruitment process. The system successfully classifies resumes into different job categories and calculates a matching score with job descriptions.

By using techniques such as TF-IDF and machine learning algorithms like Logistic Regression and SVM, the system achieves good accuracy and reduces manual effort. It speeds up the screening process and helps recruiters make better decisions

6. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide for their valuable guidance, support, and encouragement throughout the development of this project. Their suggestions and feedback helped me to complete the project successfully.
I also thank the faculty members of my department for providing the necessary resources and knowledge required for this work. Their support played an important role in completing this project

7. REFERENCES

1. T. Mikolov et al., “Efficient Estimation of Word Representations in Vector Space,” *arXiv*, 2013.
2. G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, 1988.

3. C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning Journal*, 1995.
4. L. Breiman, "Random Forests," *Machine Learning*, 2001.
5. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.
6. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O'Reilly Media, 2009.
7. Kaggle, "Resume Dataset," Available: https://www.kaggle.com
8. Python Software Foundation, "Python Documentation," Available: https://www.python.org

Statistical & Technical Tool References

1. Python Software Foundation, "Python Documentation," Available: https://www.python.org
2. Scikit-learn Documentation, "Machine Learning in Python," Available: https://scikit-learn.org
3. NumPy Documentation, "Numerical Python," Available: https://numpy.org
4. Pandas Documentation, "Data Analysis in Python," Available: https://pandas.pydata.org
5. NLTK Documentation, "Natural Language Processing," Available: https://www.nltk.org
6. Matplotlib Documentation, "Data Visualization," Available: https://matplotlib.org
7. Jupyter Notebook, Available: https://jupyter.org
8. Visual Studio Code by Microsoft, Available: https://code.visualstudio.com