**CHAPTER 1**

**INTRODUCTION**

In today’s fast-paced job market, resume screening remains one of the most time-consuming and subjective stages of hiring. HR departments often receive hundreds or even thousands of resumes for a single job opening, many of which may not be relevant to the role. Manually filtering these resumes is inefficient and increases the risk of overlooking qualified candidates.

To address these challenges, this project introduces the “Resume Classification and Shortlisting System using NLP and Machine Learning.” The system aims to automate the resume screening process by leveraging Natural Language Processing (NLP) and Machine Learning (ML) techniques to classify resumes based on predefined job roles and shortlist the most relevant candidates. By using NLP, the system can extract key information from resumes and ML algorithms can classify and rank candidates according to their suitability for a given position.

This automation reduces manual effort, improves accuracy, and provides HR professionals with tools for faster and more objective decision-making.

**1.2 PROBLEM STATEMENT**

The manual process of resume screening is labour-intensive and slow, often leading to delays in recruitment and overlooked potential candidates. HR professionals must go through numerous resumes to identify suitable candidates for specific roles, which increases operational overhead and decreases efficiency. There is a need for a smart system that can automatically categorize resumes based on job role relevance and highlight the most suitable ones.

**1.3 OBJECTIVE**

* To design and develop a resume screening system that automatically classifies resumes into predefined job categories.
* To apply Natural Language Processing techniques for cleaning and transforming textual data.
* To use machine learning models for effective classification and shortlisting.
* To identify and shortlist the best-matching resumes for selected job roles.
* To export shortlisted resumes as a CSV file for practical use by HR teams.
* To visualize the distribution of job roles and filtered resumes through graphical analysis.

**CHAPTER 2**

**IMPLEMENTATION**

Resume classification and shortlisting use Natural Language Processing (NLP) and Machine Learning (ML) to process unstructured text. NLP techniques extract key information like skills and experience, which are then converted into numerical features. ML models use these features to classify resumes into job roles and rank candidates, making the screening process faster, more accurate, and less biased.

**2.1 METHODOLOGY**

Figure 2.1 illustrates the overall workflow employed in this study, outlining each stage from data collection to final analysis

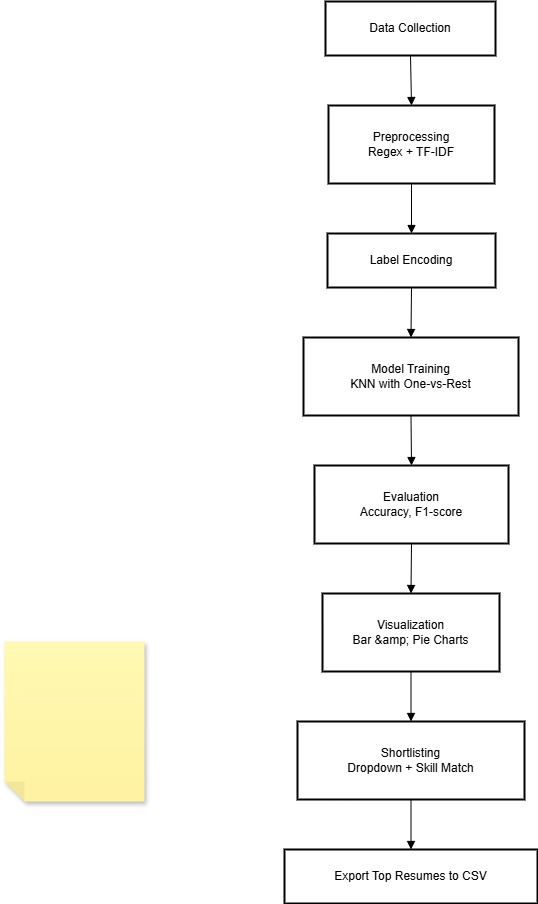


Figure 2.1: Methodology

1.Data-Collection  
A dataset containing resumes labelled with corresponding job roles serves as the foundation for both training and evaluation processes. This labelled data enables supervised learning and ensures that model performance can be quantitatively assessed.

2.Data-Preprocessing  
To prepare the resumes for analysis, the raw text is cleaned using regular expressions to remove special characters, punctuation, and other textual noise. Following this, the cleaned text is transformed into numerical format using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. This method highlights the relative importance of words across the entire corpus, making the data suitable for machine learning algorithms.

3.Label-Encoding  
The categorical job roles in the dataset are converted into numerical labels using a label encoding approach. This transformation is necessary to make the labels compatible with machine learning models that require numeric input.

4.Model-Selection-and-Training  
The classification task is addressed using the K-Nearest Neighbors (KNN) algorithm within a One-vs-Rest (OvR) framework to handle the multi-class nature of the problem. The dataset is partitioned into training and test subsets to train the model and evaluate its generalization performance on unseen data.

5.Evaluation  
The model's performance is assessed using standard evaluation metrics such as accuracy, precision, recall, and F1-score. A detailed classification report is generated to provide insights into the model's effectiveness across various job categories.

6.Visualization  
Visual representations, including bar charts and pie charts, are employed to illustrate the distribution of job categories and the count of shortlisted versus rejected resumes. These visualizations facilitate a better understanding of the dataset and the classification outcomes.

7.Shortlisting-and-Export  
An interactive interface allows users to select a job role from a dropdown menu. Based on the classification results and a comparison with predefined skill sets for each role, resumes are filtered to determine their relevance. The top-matching resumes are then exported to a CSV file, making them easily accessible for HR departments and recruiters.

**2.2 System Architecture**

The system architecture is composed of multiple interconnected modules that work collaboratively to automate resume classification and shortlisting. Each module performs a specific function to ensure the overall efficiency and reliability of the system.

•User-Interface  
Enables users (HR staff) to interact with the system, select job roles, and download shortlisted resumes.

• Data Preprocessing Module  
Handles text cleaning, normalization, and transformation into feature vectors.

• Machine Learning Engine  
Trains the classification model and predicts job categories for new resumes.

• Shortlisting Engine  
– Matches predicted categories with selected job roles.  
– Filters and identifies the most relevant resumes.  
– Exports the best-matching resumes in CSV format.

•Visualization-Module  
Displays data analysis through graphs for better understanding and decision making.

**CHAPTER 3**

**RESULT ANALYSIS**

The trained model was evaluated using test data, and classification accuracy was recorded. The system was able to correctly categorize resumes into their respective job roles with reasonable accuracy. Visualization tools provided insights into the distribution of job roles across the dataset and helped in understanding the performance of the filtering system.

Most importantly, the system enabled HR users to select a specific job role and view only the most relevant resumes. These best-matching resumes were automatically shortlisted and saved into a downloadable CSV file, reducing manual effort and increasing recruitment speed and effectiveness.

Fig 3.1and 3.2 visualizes the distribution of job categories in a dataset using two plots: a horizontal bar chart and a pie chart. It uses Seaborn and Matplotlib to show the frequency of each job category with a 'viridis' colour palette for clear visual distinction.

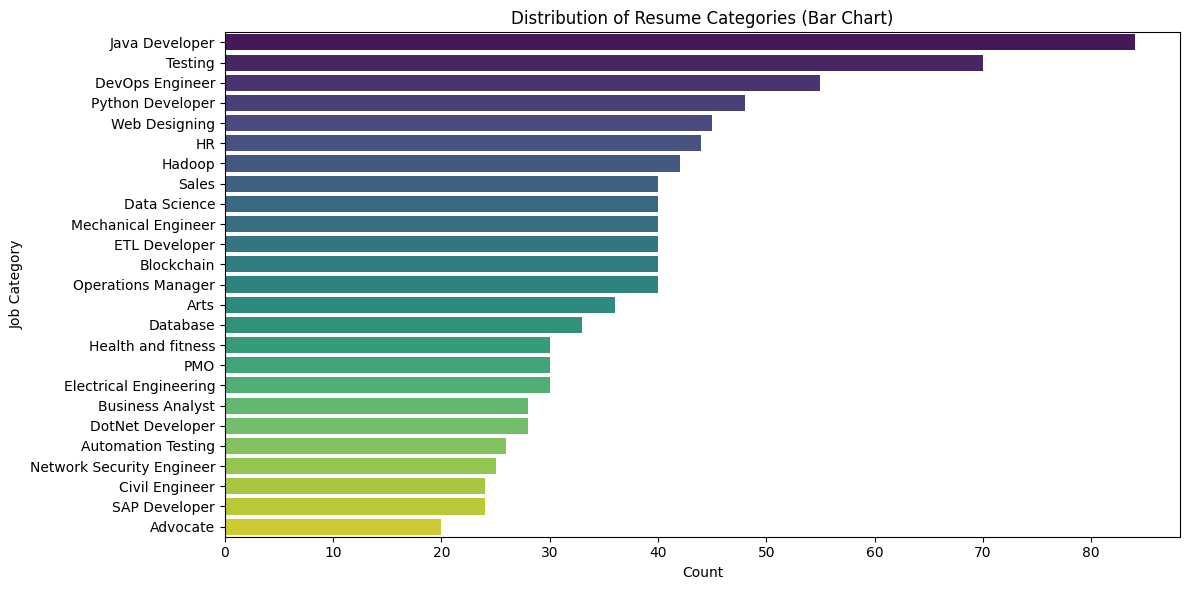


Figure 3.2:Distribution of Resume categories in dataset

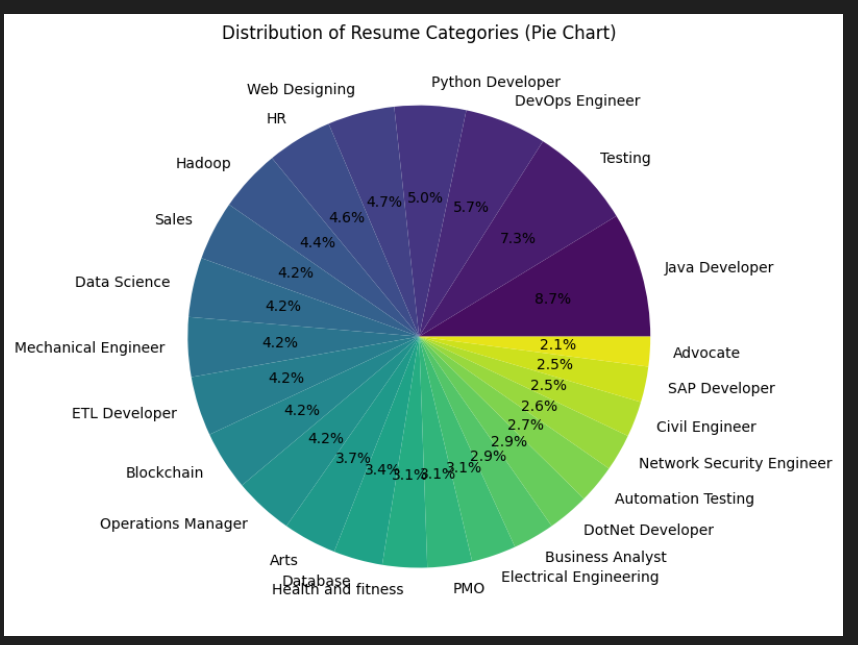


Figure 3.3: Distribution of Resume Categories

Fig 3.3 compares the accuracy of a trained K-Nearest Neighbours (KNN) classifier on both training and test datasets using a horizontal bar chart. It visually represents model performance, helping to detect overfitting or underfitting by displaying the accuracy values directly on each bar.

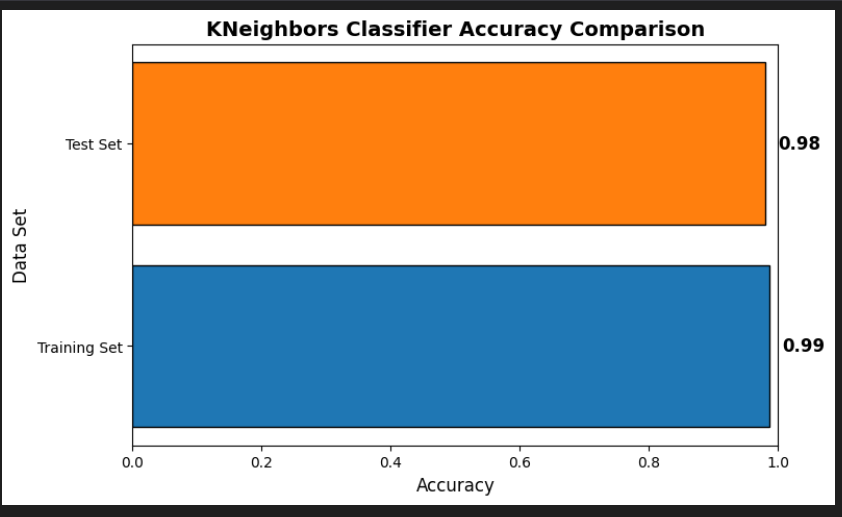


Figure 3.3: K Neighbours Classifier Accuracy Comparison

Fig 3.4 visualizes the performance of a trained KNN model using a confusion matrix heatmap. It shows classification accuracy and highlights where the model performs well or makes errors across job categories.

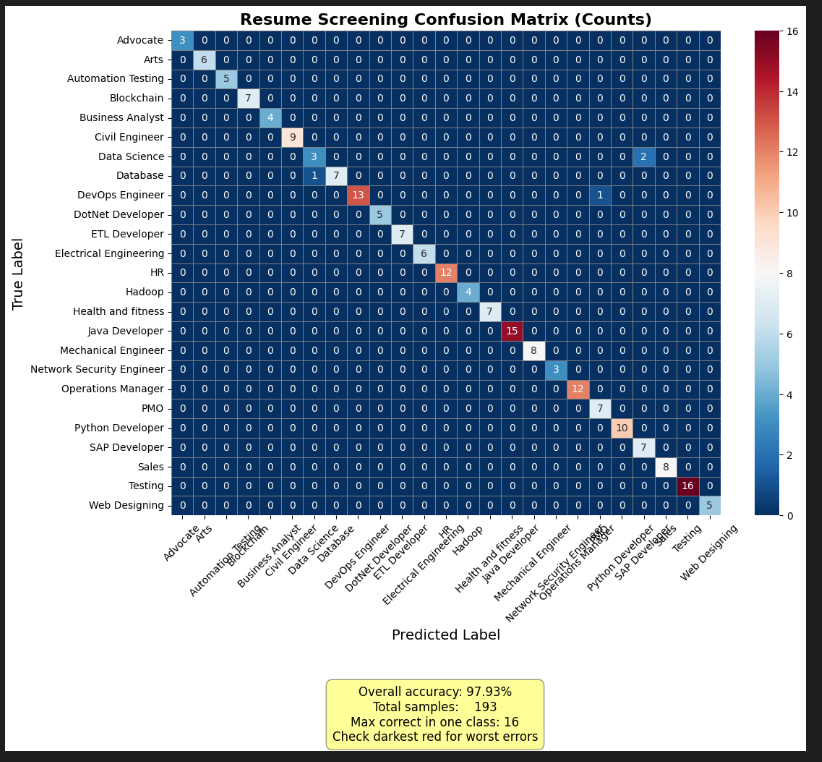


Figure 3.4: Resume Screening Confusion Matrix(counts)

Fig 3.5 builds an interactive interface using IP Widgets to filter resumes by job role from a dataset. It allows users to view the shortlisted vs. rejected resume counts with a pie chart and save the filtered resumes to a CSV file.

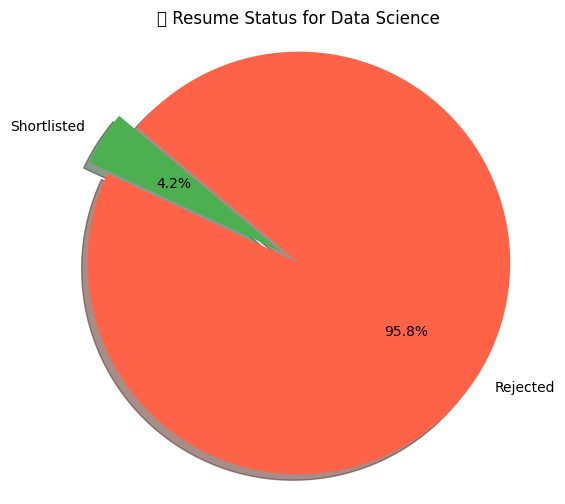


Figure 3.5: Resume status for particular job role(Data Science)

Fig 3.6 analyses resumes shortlisted for a specific job role by scoring them based on how well they match required skills. It displays a pie chart of top vs. other matches and saves the top 15 best-matching resumes to a CSV file.

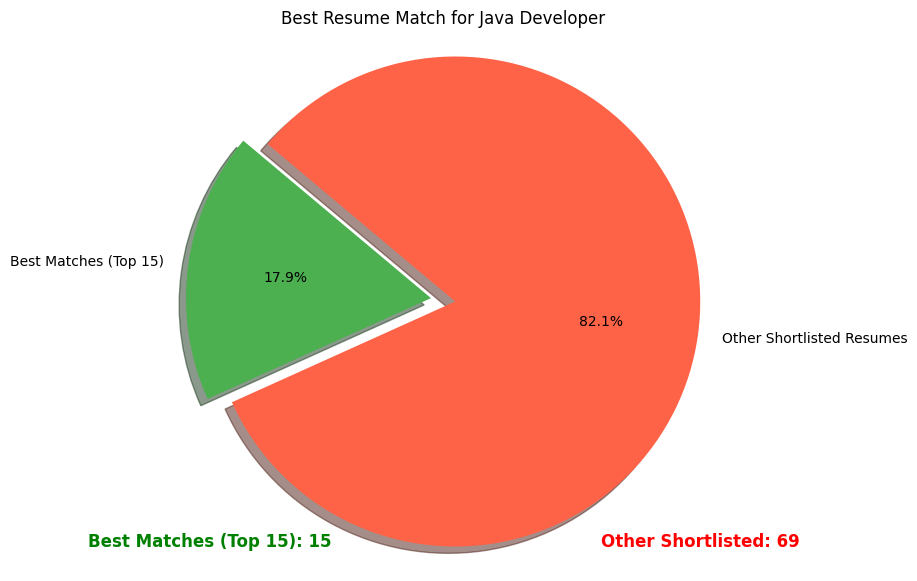


Figure 3.6: Best Resume matches for the role selected

**CHAPTER 4**

**CONCLUSION**

This project demonstrates the effective use of Natural Language Processing (NLP) and Machine Learning (ML) in automating resume screening. The system successfully categorizes resumes into job roles, identifies suitable candidates, and reduces manual effort in the recruitment process. Visualization tools provide insights into resume distribution, while the option to export shortlisted candidates in CSV format adds practical value.

Although K-Nearest Neighbors (KNN) produced acceptable results, future improvements could include implementing advanced models like BERT for better semantic understanding and accuracy. Overall, the project presents a scalable and efficient solution that enhances the speed and quality of candidate selection.

**REFERENCES**

[1] A. Heakl, J. Smith, and R. Chen, *ResumeAtlas: Revisiting Resume Classification*, arXiv preprint arXiv:2406.181252, 2024. [Online]. Available: <https://arxiv.org/abs/2406.181252>

[2] V. Paranthaman, S. Kumar, and L. Devi, “Resume screening automation,” *International Journal of Medical Informatics*, vol. 12, no. 2, 2024. [Online]. Available: <https://healthinformaticsjournal.com/index.php/IJMI/article/view/10933>

[3] Anonymous, “Resume recommendation system,” *Procedia Computer Science*, vol. 172, pp. 377–382, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187705092030750X>

[4] Y. Gan, H. Liu, and T. Zhao, *LLM Agents in Recruitment*, arXiv preprint arXiv:2401.08315, 2024. [Online]. Available: <https://arxiv.org/abs/2401.08315>

[5] A. Bhatia, M. Singh, and R. Sharma, *Resume Parsing using BERT*, arXiv preprint arXiv:1910.03089, 2019. [Online]. Available: <https://arxiv.org/abs/1910.03089>