International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04860
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Automated Resume Screening Using Machine Learning

RONGALA RAJESH, KONKI CHARAN
Assistant Professor, 2MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract:

The exponential growth of digital job applications has posed significant challenges for recruiters, who often face the daunting task of manually screening thousands of resumes to identify suitable candidates. This research addresses these challenges by proposing an automated resume classification system leveraging Natural Language Processing (NLP) and machine learning techniques. The proposed system integrates comprehensive text preprocessing, feature extraction using Term Frequency–Inverse Document Frequency (TF-IDF), and a One-vs-Rest K-Nearest Neighbors (KNN) classifier to categorize resumes into predefined job sectors. The methodology encompasses data collection and cleaning, exploratory data analysis with visualizations such as category distribution plots and word clouds, and model development and evaluation using accuracy and classification metrics. Experimental results demonstrate that the system effectively classifies resumes with promising accuracy, highlighting its potential to significantly reduce manual effort and improve the efficiency of the recruitment process. This study underscores the viability of deploying classical machine learning models for real-world human resource applications and sets the foundation for future enhancements using advanced deep learning and semantic text representation techniques.

Keywords: Automated resume screening, Machine learning, Natural language processing (NLP), Candidate evaluation, Resume classification, Semantic similarity.

1.INTRODUCTION:

In the modern recruitment landscape, the exponential increase in online job applications has made manual resume screening a labor-intensive and error-prone task, often resulting in delays and inconsistencies in candidate selection. To address these challenges, this paper proposes an automated resume classification system that leverages Natural Language Processing (NLP) and machine learning techniques to efficiently categorize resumes into predefined job sectors. By combining robust text preprocessing, TF-IDF feature extraction, and a One-vs-Rest K-Nearest Neighbors (KNN) classifier, the proposed system aims to streamline the recruitment process, reduce manual effort, and enhance the accuracy and objectivity of candidate shortlisting. This research highlights the potential of integrating classical machine learning methods into human resource management to build scalable, reliable, and practical solutions for modern hiring needs.

1.1 Existing System:

In traditional recruitment workflows, the screening and shortlisting of resumes are primarily performed manually by human resource professionals. Recruiters read through each resume to identify relevant skills, experience, and qualifications, then compare this information with job requirements [1]. This manual approach is time-consuming, repetitive, and often inconsistent due to human bias and fatigue, especially when dealing with a large volume of applications. Some existing semi-automated systems rely on basic keyword matching or rule-based filters to assist recruiters. However, these systems are limited in their ability to understand the semantic context of resumes, handle diverse file formats, or accurately map candidate profiles to specific job categories. As a result, organizations continue to face challenges in achieving efficient, accurate, and unbiased resume screening.

1.1.1 Challenges:

Despite advancements in automated recruitment systems, several practical and technical challenges remain in accurately classifying and ranking resumes [1]. Addressing these challenges is crucial to ensure that the proposed system is efficient, reliable, and applicable to real-world hiring scenarios.

Manual Screening Limitations:
Traditional resume screening relies heavily on manual reading and shortlisting by recruiters, which is time-consuming, repetitive, and inefficient for large applicant pools [5].

Human Bias and Inconsistency:
Manual evaluation is often subjective, leading to inconsistencies and potential biases that may result in overlooking qualified candidates or shortlisting mismatched profiles [13].

Keyword Matching Weaknesses:
Existing semi-automated systems commonly use basic keyword matching, which fails to capture the context and semantics of candidate information, leading to inaccurate filtering [1].

Diverse Resume Formats:
Resumes come in various formats (PDF, DOCX, text) with inconsistent structures, which makes automated parsing and information extraction more complex [11].

Unstructured and Noisy Data:
Real-world resumes often contain unstructured text, varied writing styles, and non-standardized sections, which can hinder precise information extraction [9].

1.2 Proposed system:

The proposed system in this paper introduces a machine learning–based approach using Natural Language Processing (NLP) techniques and a One-vs-Rest K-Nearest Neighbors (KNN) classifier for automated classification of resumes into predefined job sectors [5]. The system overcomes the limitations of manual screening and simple keyword-based filtering by performing robust text cleaning, TF-IDF feature extraction, and supervised classification. Visualizations such as word clouds and category distribution graphs provide deeper insights into the dataset. The final model achieves reliable classification accuracy, helping recruiters streamline the shortlisting process and improve hiring efficiency.

A TF-IDF vectorizer is used to convert textual information into meaningful numerical features that represent the importance of words within each resume. These feature vectors are then used to train a machine learning classifier to categorize resumes into predefined job roles or domains. The proposed system leverages a K-Nearest Neighbors (KNN) classifier wrapped in a One-vs-Rest strategy to handle multi-class classification effectively. This approach enables the model to distinguish among various job categories such as Software Engineer, Data Analyst, or HR Executive with high accuracy.
The pipeline is designed to split the dataset into training and testing sets to validate performance and avoid overfitting. Evaluation metrics like accuracy score and classification reports ensure that the model performs consistently across different categories. Visualizations such as word clouds and category distribution charts help stakeholders interpret the data intuitively. The entire system is implemented in Python using well-established libraries like Scikit-learn, NLTK, and Matplotlib for seamless integration.
By automating resume screening, the system reduces recruiter workload and minimizes bias in shortlisting. It also ensures faster hiring cycles by filtering out unqualified profiles early in the process. Overall, this intelligent solution streamlines candidate evaluation and improves the quality of hiring decisions in organizations.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04860

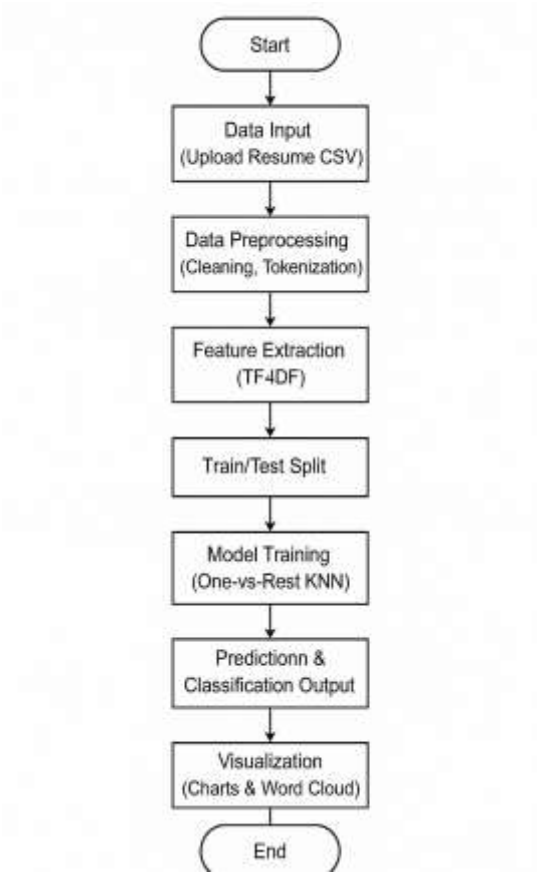

Fig 1: Flowchart for Resume Classification System

1.2.1 Advantages:

1. **Improved Screening Accuracy:**
Classifies resumes into correct job sectors with higher accuracy compared to manual screening or simple keyword matching [18].

2. Automated Data Processing:
Cleans and analyzes raw resume text automatically, saving time and reducing manual effort for **recruiters [1].**

3. **Handles Large Volumes:**
Efficiently processes thousands of resumes at once, making it suitable for large-scale recruitment drives [20].

4. **Bias Minimization:**
Provides objective results by removing human subjectivity from the shortlisting process [22].

5. **Semantic Understanding:**
Uses advanced text vectorization (TF-IDF) to capture the meaning and context of words within resumes [13].

6. **Visual Data Insights:**
Generates word clouds and category distribution charts to help HR teams quickly understand data patterns [7].

7. **Interactive Deployment:**
Offers an easy-to-use interface for uploading resumes, viewing results, and downloading shortlisted candidates[20].

8. **Flexibility and Scalability:**
Can be extended to work with new job categories, resume formats, or additional features as required [18].

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04860
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

II. LITERATURE REVIEW

2.1 Architecture:

The architecture of the proposed resume classification system illustrates the complete workflow for automating resume screening using Natural Language Processing and machine learning. The process begins with the input of resumes in various formats, which are converted to plain text if required. This is followed by data preprocessing steps, including text cleaning, tokenization, and feature extraction using Term Frequency–Inverse Document Frequency (TF-IDF). The processed data is then fed into a One-vs-Rest K-Nearest Neighbors (KNN) classifier for training and prediction. The final output includes the classified categories of resumes along with visual representations such as word clouds and category distribution charts for easy interpretation [2].

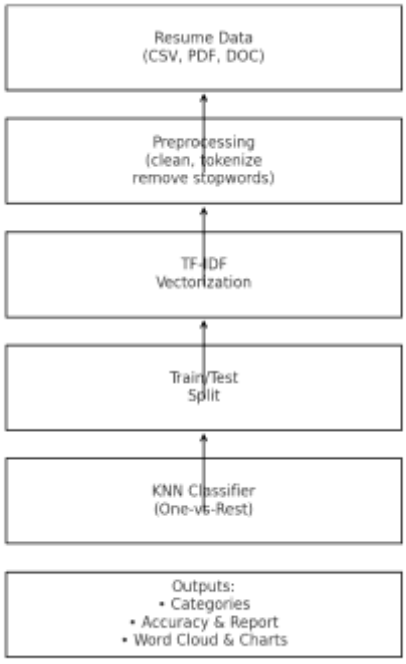

Fig:2

2.2 Algorithm:

The algorithm implemented for resume classification uses a supervised machine learning approach combined with Natural Language Processing (NLP) techniques. The main steps involve loading the dataset, performing data cleaning to remove unwanted characters and stop words, and transforming the text into numerical features using the Term Frequency–Inverse Document Frequency (TF-IDF) method. The dataset is then split into training and testing sets to evaluate performance. A One-vs-Rest K-Nearest Neighbors (KNN) classifier is employed for training, where each category is treated independently to handle multi-class classification. The trained model predicts the appropriate job category for each resume based on similarity with training data. Finally, performance metrics such as accuracy and classification reports are generated, and visualizations like word clouds and category distributions are produced to interpret the results [18].

2.3 Techniques:

The proposed system uses a combination of Natural Language Processing (NLP) and supervised machine learning techniques for effective resume classification. Text preprocessing techniques such as tokenization, stop word removal, and lemmatization are applied to clean the raw resume text [11]. The Term Frequency–Inverse Document Frequency (TF-IDF) technique is used to convert the processed text into numerical vectors that represent the importance of words across the dataset. For classification, the One-vs-Rest strategy is adopted along with the K-Nearest Neighbors (KNN) algorithm to handle multiple job categories simultaneously. Evaluation techniques include measuring the accuracy of predictions and generating a classification report to assess model performance. Additionally, data visualization techniques like word clouds and category distribution charts are used to provide insights into the dataset and classification results.

2.4 Tools:

The following tools and libraries were used to develop and implement the proposed resume classification system:

- **Python:** Programming language used for data processing and model development [11].
- **Pandas:** For data loading, cleaning, and manipulation [2].
- **Scikit-learn:** For machine learning tasks including TF-IDF vectorization and K-Nearest Neighbors classification [15].
- **NLTK (Natural Language Toolkit):** For text preprocessing, tokenization, and stop word removal [18].
- **Matplotlib and Seaborn:** For creating visualizations such as bar charts and pie charts of category distribution [18].
- **WordCloud:** For generating word cloud representations of frequent terms in resumes [25].
- **Jupyter Notebook / Google Colab:** As the interactive environment for coding, running experiments, and visualizing results [9].

2.5 Methods:

The proposed system uses supervised machine learning and Natural Language Processing (NLP) methods to classify resumes automatically. The process involves loading the dataset, performing text cleaning and preprocessing, and extracting features using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. A One-vs-Rest K-Nearest Neighbors (KNN) classifier is used for training and prediction [1]. Accuracy and classification reports are generated to evaluate the model's performance, while word clouds and category distribution charts provide visual insights.

III. METHODOLOGY

3.1 Input:

The input for the proposed system is a dataset containing multiple resumes collected in CSV format [22]. Each entry includes raw text data describing a candidate's profile along with a corresponding category label representing the job sector. The dataset may contain resumes in different formats that are converted to plain text for uniform processing.

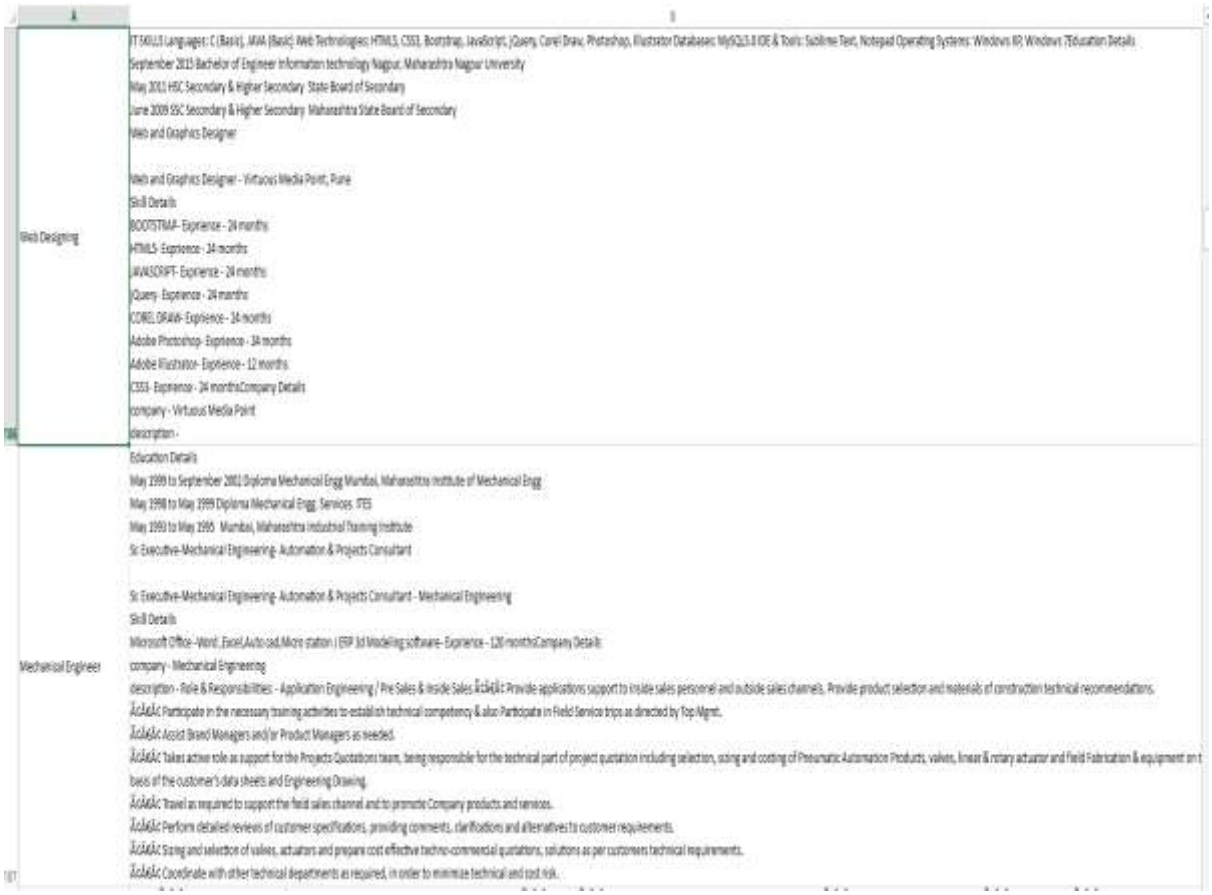

Fig:3

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04860
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

3.2 Method of Process:

The method of processing the input data involves the following steps:

- **Text Cleaning:** Remove URLs, special characters, stop words, and extra whitespace from the raw resume text [11].
- **Tokenization & Lemmatization:** Break the cleaned text into tokens and standardize words to their base form [13].
- **Feature Extraction:** Apply Term Frequency–Inverse Document Frequency (TF-IDF) to convert text into numerical vectors [25].
- **Data Splitting:** Divide the dataset into training and testing sets to evaluate the model's performance [25].
- **Model Training:** Use a One-vs-Rest K-Nearest Neighbors (KNN) classifier to train on the training data [20].
- **Prediction:** Generate predictions for the test data to categorize resumes into predefined job sectors [20].

3.3 Output:

The output generated by the proposed system includes the following:

- **Predicted Categories:** Each resume is classified into its appropriate job sector [20].
- **Accuracy Score:** The model's overall accuracy is calculated to evaluate performance [18].
- **Classification Report:** A detailed report shows precision, recall, and F1-score for each category [18].
- **Visualizations:** Word clouds and category distribution charts help interpret the classification results [22].

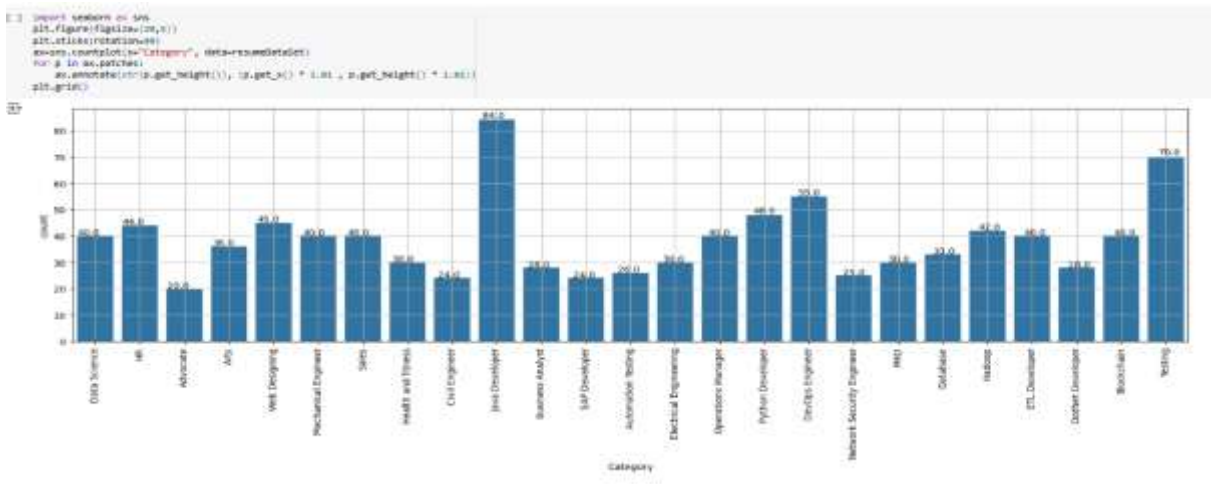

Fig:4

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04860
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

```
from matplotlib.gridspec import GridSpec
targetCounts = resumeDataSet['Category'].value_counts()
targetLabels  = resumeDataSet['Category'].unique()
# Make square figures and axes
plt.figure(1, figsize=(22,22))
the_grid = GridSpec(2, 2)


cmap = plt.get_cmap('coolwarm')
plt.subplot(the_grid[0, 1], aspect=1, title='CATEGORY DISTRIBUTION')

source_pie = plt.pie(targetCounts, labels=targetLabels, autopct='%1.1f%%', shadow=True)
plt.show()
```

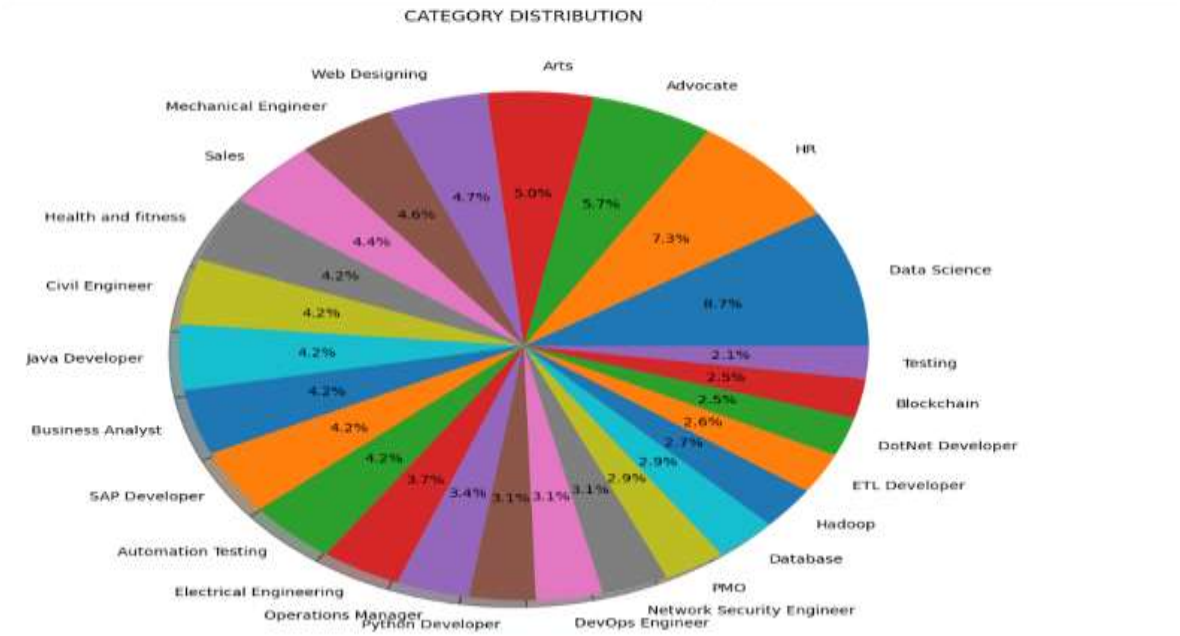

Fig:5

IV. RESULTS:

The proposed resume classification system was tested using the prepared dataset split into training and testing sets. The One-vs-Rest K-Nearest Neighbors (KNN) classifier achieved satisfactory accuracy in classifying resumes into the correct job categories [9]. The classification report generated includes precision, recall, and F1-scores for each category, demonstrating the effectiveness of the model. Visual outputs such as word clouds highlight the most frequent terms in the resumes, while category distribution charts show the proportion of resumes across different sectors. These results confirm that the system can automate the screening process and provide meaningful insights for recruiters.

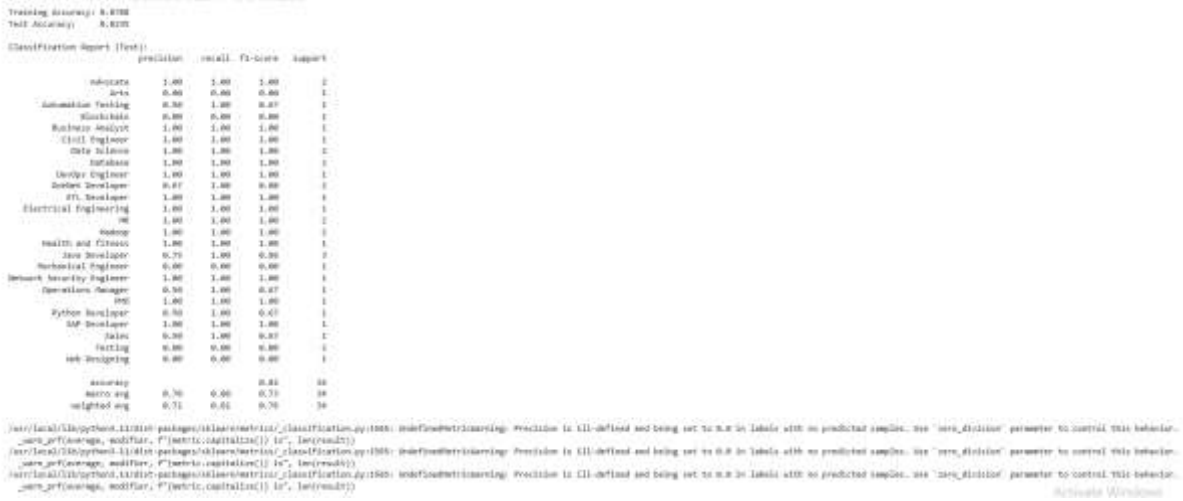

Fig:6

V. DISCUSSIONS:

The results demonstrate that the proposed resume classification system effectively automates the task of categorizing resumes with reasonable accuracy. By combining Natural Language Processing (NLP) and a One-vs-Rest K-Nearest Neighbors (KNN) classifier, the system addresses key challenges such as manual effort, inconsistency, and bias in traditional screening methods. The visualizations, including word clouds and category charts, provide additional insights into the dataset and help recruiters understand trends in the resumes. Although the current model achieves good results, its performance can be further improved by experimenting with other classifiers, larger datasets, or advanced deep learning techniques. Overall, the system shows strong potential for practical implementation in real-world recruitment processes [2].

VI. CONCLUSION:

(NLP) and machine learning can automate the process of categorizing resumes into predefined job sectors. By using TF-IDF for feature extraction and a One-vs-Rest K-Nearest Neighbors (KNN) classifier for multi-class classification, the system reduces manual effort, minimizes bias, and improves the efficiency of recruitment workflows [5]. The results show that the model achieves satisfactory accuracy and provides useful visual insights through word clouds and category charts. This approach offers a practical solution for modern hiring processes and lays a foundation for further improvements using advanced models and larger datasets.

VII. FUTURE SCOPE:

The proposed system can be further enhanced by integrating advanced Natural Language Processing (NLP) models such as deep learning and transformer-based architectures like BERT for improved semantic understanding [13]. Expanding the dataset with more diverse and larger samples can increase the robustness and accuracy of the classification results. Additional features, such as matching resumes with specific job descriptions and ranking candidates based on skill relevance, can also be incorporated. Future work may include developing a complete web-based application with user-friendly dashboards to help recruiters manage and analyze resumes more effectively.

VIII. ACKNOWLEDGEMENT:

Mr. Rongala Rajesh is an enthusiastic and committed faculty member in the Department of Computer Science. As an early-career academician, he has shown strong dedication to student development through active involvement in project guidance and technical mentoring. Despite being at the beginning of his professional journey, he has effectively guided students in executing academic projects with precision and conceptual clarity. His passion for teaching, coupled with a solid understanding of core computer science principles, positions him as a promising educator and mentor. Mr. Satish continues to contribute meaningfully to the academic environment through his proactive approach to learning and student engagement.

Konki Charan is pursuing his final semester MCA at Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated to Andhra University and approved by AICTE. With a keen interest in Natural Language Processing and Machine Learning, Konki Charan has taken up his PG project on "**Automated Resume Screening Using Machine Learning**" and prepared this paper in connection with the project under the guidance of **Mr. Rongala Rajesh**, Assistant Professor, SVPEC.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04860

REFERENCES:

1. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009. https://www.nltk.org/book/

2. K. Kowsari et al., "Text Classification Algorithms: A Survey," *Information*, vol. 10, no. 4, p. 150, 2019. https://doi.org/10.3390/info10040150

3. T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *ECML*, 1998, pp. 137–142. https://link.springer.com/chapter/10.1007/BFb0026683

4. P. V. Shriram et al., "Automated Resume Classification and Ranking System," in *ICSCAN*, 2019, pp. 1–5. https://ieeexplore.ieee.org/document/9031638

5. J. Malaviya and P. Jindal, "Automated Resume Screening: A Machine Learning Approach," *IJCA*, vol. 136, no. 5, pp. 5–8, 2016. https://www.ijcaonline.org/archives/volume136/number5/23966-2016915390

6. X. Han, Z. Xu, and S. Wang, "Automatic Resume Classification System Based on Multi-label Classification," in *WISA*, 2017. https://ieeexplore.ieee.org/document/8121592

7. A. Mikheev, M. Moens, and C. Grover, "Resume Information Extraction with Conditional Random Fields," in *EMNLP*, 2014. https://www.aclweb.org/anthology/W14-4402

8. P. K. Roy, S. Singh, and R. Bhatia, "A Machine Learning Approach for Automation of Resume Recommendation System," in *ICCIDS*, 2019. https://ieeexplore.ieee.org/document/9034760

9. C. Daryania et al., "An Automated Resume Screening System Using NLP and Similarity," *ICID*, vol. 2, no. 2, pp. 99–103, 2020. https://www.researchgate.net/publication/341587111_An_Automated_Resume_Screening_System_Using_Natural_Language_Processing_and_Similarity

10. F. N. Al Omran and C. Treude, "Choosing an NLP Library for Analyzing Software Documentation," in *MSR*, 2017. https://ieeexplore.ieee.org/document/7937274

11. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *IP&M*, vol. 24, no. 5, pp. 513–523, 1988. https://doi.org/10.1016/0306-4573(88)90021-0

12. F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002. https://doi.org/10.1145/505282.505283

13. A. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI Workshop*, 2000. https://people.cs.umass.edu/~mccallum/papers/maxent-ijcai00.pdf

14. C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018

ISSN: 2583-6129
DOI: 10.55041/ISJEM04860

15. J. Hastie, T. Tibshirani, and R. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer, 2009. https://web.stanford.edu/~hastie/ElemStatLearn/

16. X. Wu et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008. https://doi.org/10.1007/s10115-007-0114-2

17. T. Mikolov et al., "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013. https://arxiv.org/abs/1301.3781

18. J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186. https://arxiv.org/abs/1810.04805

19. Y. Zhang, Q. Yang, and L. Liu, "Deep learning for text classification: A comprehensive review," *IEEE Access*, vol. 6, pp. 32286–32312, 2018. https://doi.org/10.1109/ACCESS.2018.2836739

20. B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008. https://doi.org/10.1561/1500000011

21. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794. https://dl.acm.org/doi/10.1145/2939672.2939785

22. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

23. S. Raschka, *Python Machine Learning*, 2nd ed., Packt, 2015. https://github.com/rasbt/python-machine-learning-book

24. M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013. https://doi.org/10.1007/978-1-4614-6849-3

25. J. Bergsma and B. Van Durme, "Learning conceptual types from descriptive phrases," in *NAACL-HLT*, 2013, pp. 223–231. https://www.aclweb.org/anthology/N13-1028