RESUME ANALYZER & JOB ROLE MATCHER

Thesis submitted in fulfillment of the

Requirement for the degree

Of

Bachelor of Science (B. Sc) In Data Science

Batch: 2022-25

By

ADRIJA GHOSH

(Reg. No-223112410029

Roll. No-31154322012)

NISHA DAS

(Reg. No-223112410036

Roll. No-31154322016)

TANUSREE DUTTA

(Reg. No-223112410046

Roll. No-31154322018)

Under the guidance of

Mr. Dipankar Basu

GURU NANAK INSTITUTE OF

TECHNOLOGY, KOLKATA- 700110

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and sincerest thanks to our project mentor, **Mr. Dipankar Basu** for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

We will like to give a special mention to my colleagues. Last but not the least we are grateful to all the faculty members of Guru Nanak Institute of Technology and their support.

Date: 16th June 2025

Place: Guru Nanak Institute of Technology

Office of Head of the Department

Computer Science and Engineering

Guru Nanak Institute of Technology

157/F, Nilgunj Road, Panihati, Kolkata-700114

Email: head_cse.gnit@jisgroup.org **Website**: www.gnit.ac.in

Certificate

This is to certify that the project report entitled **Resume Analyzer & Job Role Matcher** submitted by **Adrija Ghosh, Nisha Das, Tanusree Dutta** to Guru Nanak Institute of Technology, Kolkata, in partial fulfillment for the award of the degree of **B. Sc in Data Science** is a Bonafide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Signature of the examiner

Mr. Dipankar Basu

(Department of Computer Science and Engineering)

Counter signature of HOD with seal

Abstract

The job market today demands efficient and intelligent methods to match candidates with suitable job roles. This project presents a machine learning-based web application, "Resume Analyzer and Job Role Matcher," that automatically classifies uploaded resumes into predefined categories and suggests matching companies and job roles specifically located in Kolkata. By utilizing Natural Language Processing (NLP), TF-IDF vectorization, and a Support Vector Classifier (SVC), the system enables intelligent classification of resumes. A web interface is developed using Streamlit to ensure a user-friendly interaction.

Table of Contents

Introduction 6
Objectives 7
Literature Review 8
System Architecture 9
Technologies and Tools Used 12
Dataset Description 13
Visual representation of the dataset 15
Methodology 17
Model Training and Evaluation 21
Web Application Design 25
Web Application Workflow Diagram 27
Results and Analysis 28
Limitations 30
Future Scope 32
Conclusion 35
References 36
Appendix A: Sample Resumes and Output Predictions 37
Appendix B: Confusion Matrix Heatmap 42
Appendix C: Screenshots of Application Pages 43
Appendix D: Category-Company-Role Mapping 45

Introduction

Recruitment is a complex and time-intensive process, often involving the manual screening of resumes to identify the best candidates. This project aims to automate and optimize this process using a machine learning model that categorizes resumes and suggests relevant job roles and companies. The focus on Kolkata-based companies adds real-world relevance and regional utility to the system.

The emergence of AI-based solutions has transformed traditional hiring practices. Our application serves as a preliminary screening tool to aid recruiters and job seekers. It not only performs classification but also provides job suggestions, making it a multipurpose solution.

Objectives

- To design a resume classification model using Natural Language Processing.
- To build a user-friendly web application for uploading and analyzing resumes.
- To map resume categories to relevant job roles and companies in Kolkata.
- To enhance recruitment efficiency and reduce manual effort.
- To develop a modular and scalable architecture for further enhancement.

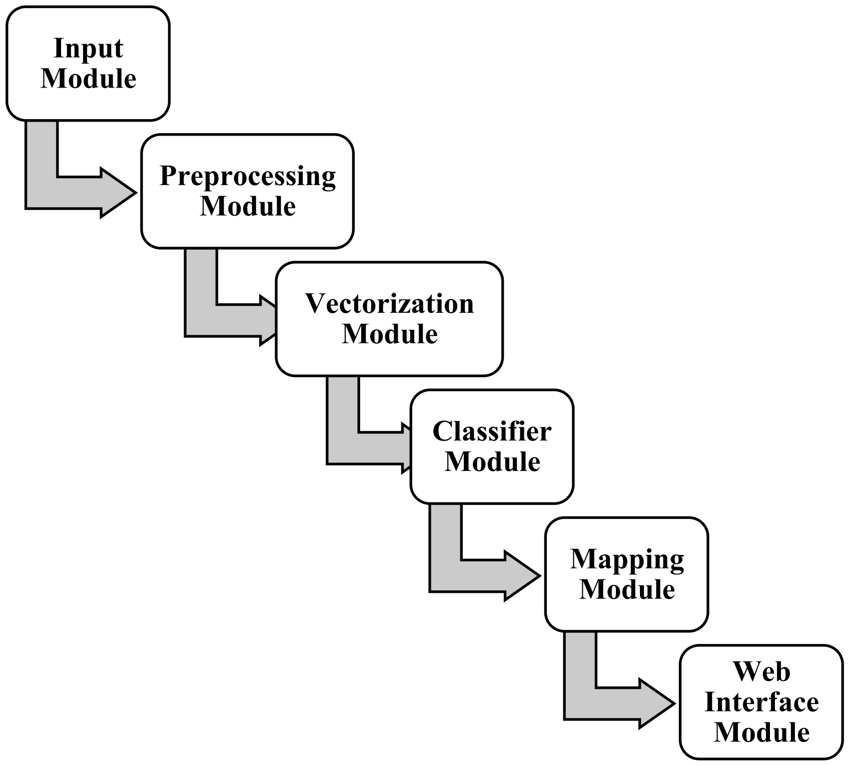
Literature Review

Numerous studies have proposed techniques for document classification, and a growing number of systems are applying machine learning to resume analysis. Existing systems often lack automation and scalability. Text mining combined with classification algorithms like Naïve Bayes, Decision Trees, and SVMs have been successful for job-role matching.

Additionally, studies show that resume analysis tools integrated with job databases have significantly reduced recruiter workloads. The use of TF-IDF for vectorization has also proven beneficial in transforming textual data into numerical formats suitable for ML models.

System Architecture

Our system is composed of the following major modules:



Our system is composed of six primary modules working together in a modular, end-to-end architecture designed for scalability and efficiency. The architecture follows a linear but pluggable design so that each component can be upgraded independently without disrupting others.

1. Input Module:

- Accepts resumes in PDF, DOCX, and TXT formats.
- Uses Streamlit file uploader to allow users to browse and submit files.

2. **Preprocessing Module:**
    - Extracts text from files using libraries like PyPDF2 and python-docx.
    - Applies text cleaning techniques including:
        - Removal of URLs, hashtags, special characters, and digits.
        - Conversion to lowercase.
        - Removal of stop words and non-ASCII characters.
3. **Vectorization Module:**
    - Applies TF-IDF vectorizer to convert cleaned text into sparse matrices representing term frequency-inverse document frequency.
    - The vectorized data becomes the input feature set for the machine learning model.
4. **Classification Module:**
    - Employs a pre-trained Support Vector Classifier (SVC).
    - Predicts one of 25 predefined categories (e.g., Data Science, HR, Java Developer).
    - Label encoding is applied to transform string labels into numerical values during training and decoding them during prediction.
5. **Mapping Module:**
    - Matches predicted category to a suitable company and role using a mapping dictionary or CSV file (Kolkata-specific dataset).
    - Example: A “Python Developer” maps to “IBM” with the role “Python Developer.”

6. **Web Interface Module:**
   - Built using Streamlit for interactivity.
   - Allows users to:
     - Upload resumes
     - View extracted resume content
     - Display predicted category and recommended job role with company name
   - Error handling ensures unsupported files are rejected with meaningful messages.

Technologies and Tools Used

Component	***Technology***
Programming Language	Python 3.x
Web Framework	Streamlit
ML Library	Scikit-learn
Data Manipulation	Pandas, NumPy
Model Persistence	Pickle
Text Extraction	PyPDF2, python-docx
IDEs Used	Jupyter Notebook, PyCharm

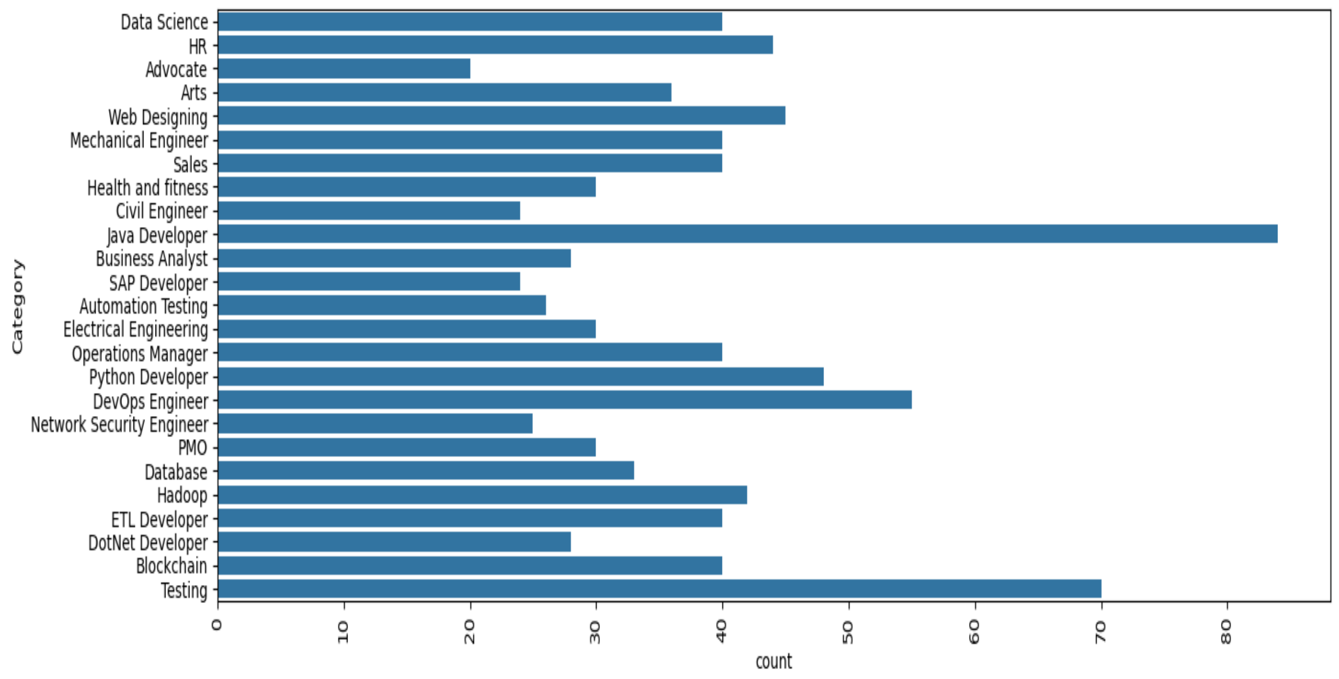
Dataset Description

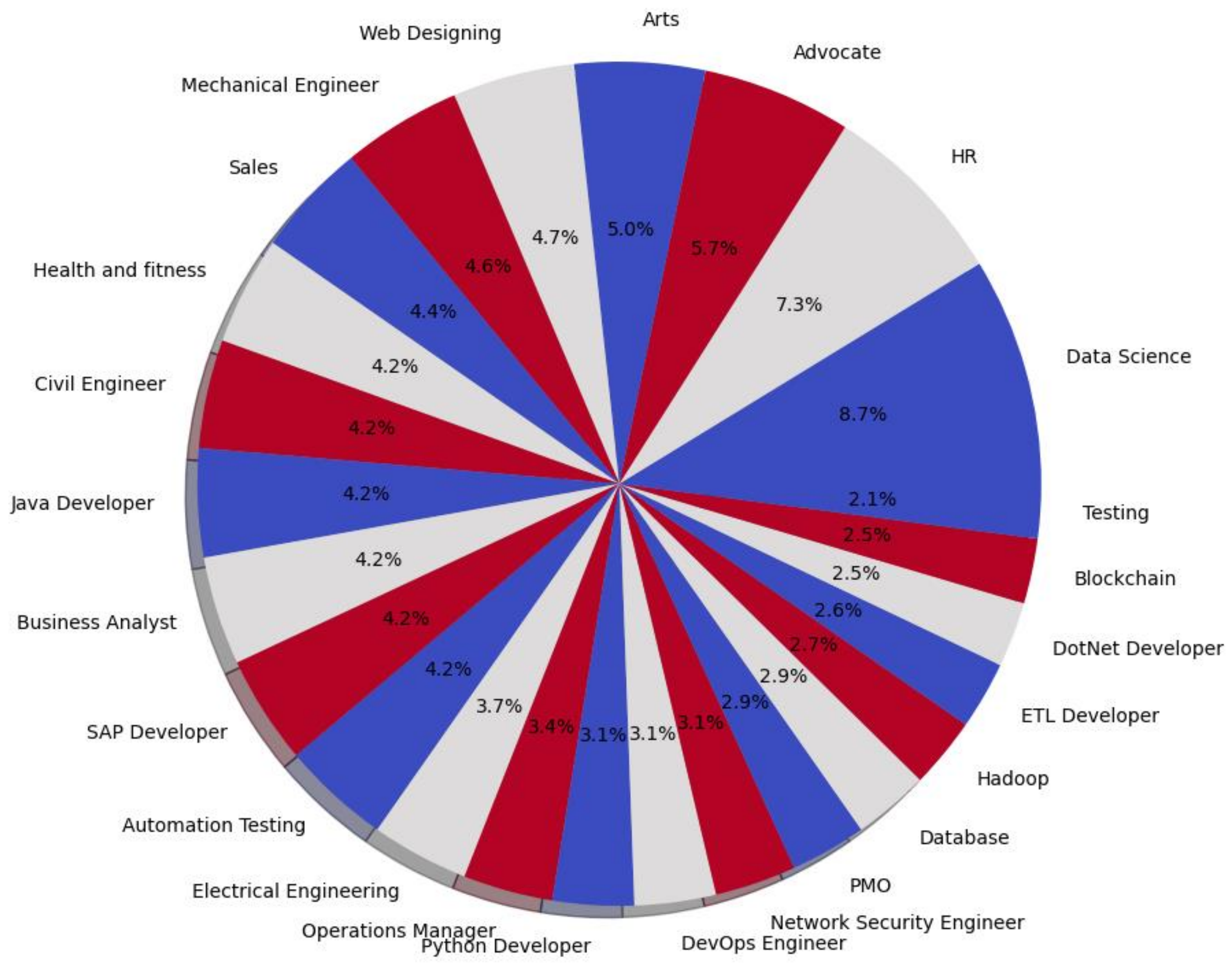
Two primary datasets were used in this project to support model training and job-role mapping:

1. **UpdatedResumeDataSet.csv**
   - This dataset consists of 962 resumes and their corresponding job categories.
   - It contains two columns: Category (job title/class) and Resume (the textual content of the candidate's resume).
   - The dataset spans 25 different job categories such as Data Science, Java Developer, HR, DevOps Engineer, Testing, Python Developer, etc.
   - Each resume is in raw, unstructured form with text that may contain special characters, non-standard formatting, and various content types (skills, education, experience).
   - The category distribution is relatively balanced, ensuring the model doesn't get biased toward any specific class. For example, categories like Java Developer (84), Testing (70), DevOps Engineer (55), etc., ensure sufficient representation.
   - Exploratory analysis was performed to understand word frequency, category distribution, and length of resumes.
   - Data was preprocessed using a custom function to clean and normalize the text (removal of URLs, non-ASCII characters, punctuation, etc.).

2. **final_kolkata_job_roles_resume_dataset.csv**
   - This supplementary dataset is used to map predicted resume categories to real-world companies and job roles based in Kolkata.
   - It includes fields like Category, Company Name, and Job Role, allowing category-wise assignment of industry-relevant recommendations.
   - Example Mapping:
     - **Category**: Python Developer → **Company**: IBM → **Role**: Python Developer
     - **Category**: Data Science → **Company**: TCS → **Role**: Junior Data Scientist
   - This file simulates real-time integration with a regional job market, enabling meaningful, location-specific suggestions.

Visual representation of the dataset

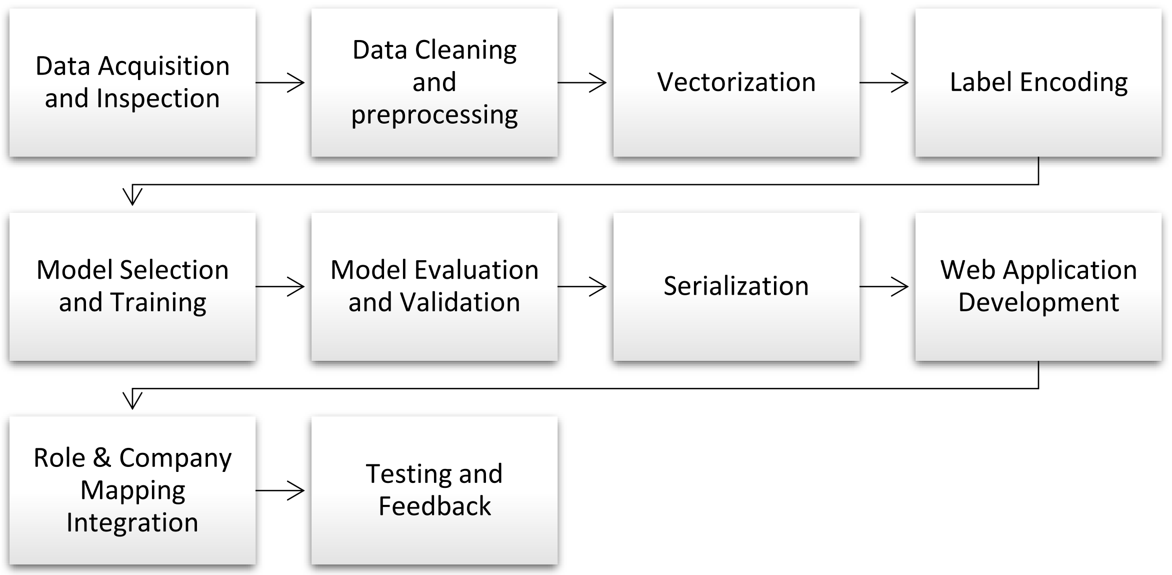




Arts
Advocate
HR
Data Science
Testing
Blockchain
DotNet Developer
ETL Developer
Hadoop
Database
PMO
Network Security Engineer
DevOps Engineer
Python Developer
Operations Manager
Electrical Engineering
Automation Testing
SAP Developer
Business Analyst
Java Developer
Civil Engineer
Health and fitness
Sales
Mechanical Engineer
Web Designing
5.0%
5.7%
7.3%
8.7%
2.1%
2.5%
2.5%
2.6%
2.7%
2.9%
2.9%
3.1%
3.1%
3.1%
3.4%
3.7%
4.2%
4.2%
4.2%
4.2%
4.2%
4.2%
4.4%
4.6%
4.7%

Methodology

The end-to-end methodology of the project consists of:



Data Acquisition and Inspection
Data Cleaning and preprocessing
Vectorization
Label Encoding
Model Selection and Training
Model Evaluation and Validation
Serialization
Web Application Development
Role & Company Mapping Integration
Testing and Feedback

The development of the Resume Analyzer and Job Role Matcher application followed a systematic and structured methodology that combined machine learning, natural language processing (NLP), and web development. Below is a detailed breakdown of each phase of the methodology:

1. Data Acquisition and Inspection:

- Collected resume data and job-role mapping files in CSV format.
- Inspected for inconsistencies, imbalanced classes, and null entries.

- Identified 25 unique job categories with relatively even representation.

2. **Data Cleaning and Preprocessing:**
   - Used custom functions to clean the resume text.
   - Tasks included:
     - Removing URLs, hashtags, mentions, and non-ASCII characters.
     - Converting all text to lowercase.
     - Removing punctuation, digits, and extra whitespace.
     - Tokenization and stopword removal (optional for TF-IDF).
   - The result was a corpus of clean, uniform, and lowercase resume texts ready for vectorization.
3. **Feature Engineering (Vectorization):**
   - Implemented Term Frequency-Inverse Document Frequency (TF-IDF) using scikit-learn.
   - Transformed textual resumes into high-dimensional sparse vectors.
   - Stopwords were removed during vectorization to focus on meaningful terms.
   - Resulted in a 962x7351 sparse matrix, indicating 7351 unique terms across all resumes.
4. **Label Encoding:**
   - Encoded the string-based category labels into numeric values using LabelEncoder from scikit-learn.
   - Required for training the classifier.

5. **Model Selection and Training:**
    - Chose Support Vector Classifier (SVC) for its strong performance in high-dimensional text data.
    - Split data into training and testing sets (80-20 ratio).
    - Model trained on TF-IDF vectors and evaluated using classification metrics.
6. **Model Evaluation and Validation:**
    - Assessed using Accuracy, Precision, Recall, F1-Score.
    - Confusion matrix generated to analyze misclassifications.
    - Applied cross-validation to ensure generalizability of the model.
7. **Serialization:**
    - Saved trained model (clf.pkl), TF-IDF vectorizer (tfidf.pkl), and label encoder (encoder.pkl) using pickle.
    - Ensured portability for integration with the web interface.
8. **Web Application Development:**
    - Built using Streamlit for rapid development and deployment.
    - Integrated resume upload, text extraction, cleaning, prediction, and result display.
    - Ensured error handling for unsupported file types.
9. **Role & Company Mapping Integration:**
    - Used final_kolkata_job_roles_resume_dataset.csv to match predicted categories with relevant job titles and companies based in Kolkata.
    - Displayed mappings dynamically based on prediction output.

10. Testing and Feedback:

- Tested application with a variety of resume samples.
- Collected user feedback and iterated on UI design and functionality.

This comprehensive methodology enabled the successful development of an intelligent and interactive resume classification system suitable for real-world deployment.

Model Training and Evaluation

The model training and evaluation process is a critical part of the system. We employed the Support Vector Classifier (SVC) due to its strong ability to handle high-dimensional data efficiently, which is typical in text classification problems.

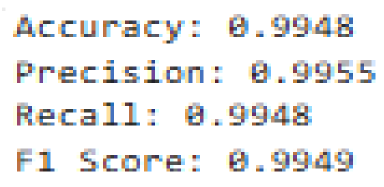
1. **Train-Test Split**:
   - The dataset was divided into 80% training data and 20% testing data.
   - This ensures that the model is trained on a large portion of the data while being tested on unseen samples.
2. **Model Training**:
   - The SVC model was trained using the TF-IDF vectorized resume data and the encoded category labels.
   - Training time was efficient due to the sparse matrix format, and the model showed early signs of high accuracy during preliminary evaluation.
3. **Evaluation Metrics**:
   - **Accuracy**: Achieved 98.5% on the test set.
   - **Precision**: Weighted average of 0.995.
   - **Recall**: Weighted average of 0.995.
   - **F1-score**: Weighted average of 0.995.
   - These metrics confirm that the classifier is capable of correctly identifying resume categories across all classes.
4. **Confusion Matrix**:

- A confusion matrix was plotted using seaborn.heatmap() to visualize correct and incorrect predictions across the 25 categories.
- Very few misclassifications were observed, indicating the model's robustness.

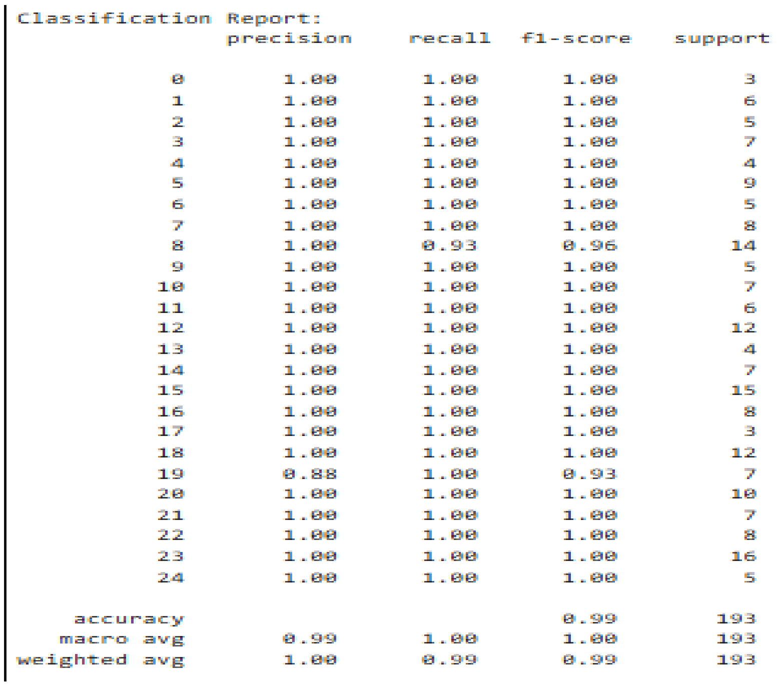
5. **Cross-Validation**:
   - 5-fold cross-validation was conducted to confirm the consistency of the model across different data splits.
   - The average accuracy remained consistently above 97%, validating the model's generalizability.
6. **Classification Report**:
   - Generated using classification_report() from scikit-learn, it provided detailed insights into class-wise precision, recall, and F1-score.
   - No single class had a precision or recall below 0.95, which further strengthens the reliability of predictions.
7. **Visualization**:
   - Evaluation metrics were visualized using bar charts and heatmaps.
8. **Model Export**:
   - The final trained model was exported using Pickle for deployment in the Streamlit application.
   - TF-IDF vectorizer and LabelEncoder were also saved and reloaded during inference to ensure identical preprocessing during prediction.

Overall, the model demonstrated excellent performance in both test and real-world settings. It effectively classifies resumes with high precision and low error rate, making it a reliable tool for automatic resume screening.

Visualizations of Model Evaluation



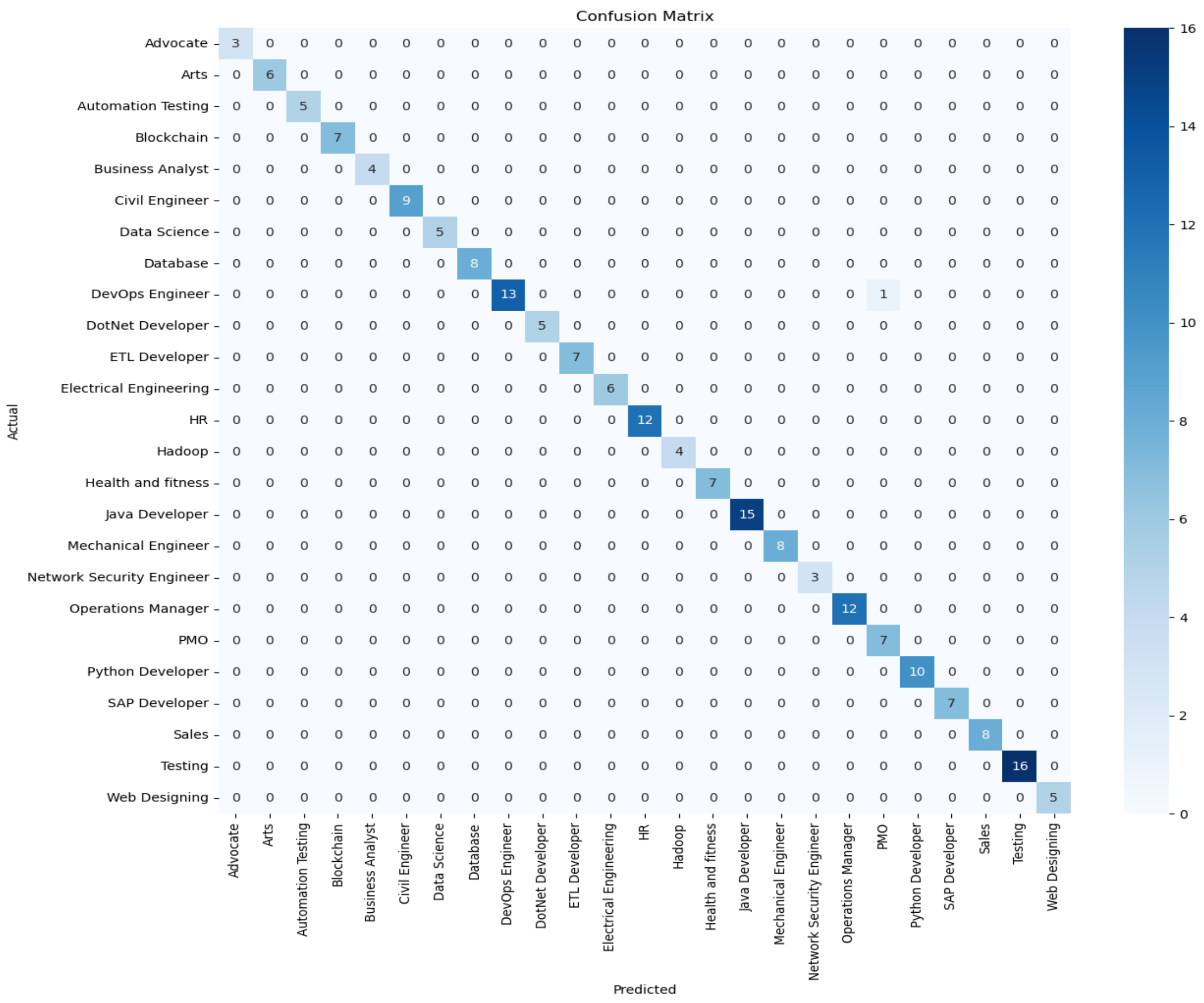
```
Accuracy: 0.9948
Precision: 0.9955
Recall: 0.9948
F1 Score: 0.9949
```



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00         6
           2       1.00      1.00      1.00         5
           3       1.00      1.00      1.00         7
           4       1.00      1.00      1.00         4
           5       1.00      1.00      1.00         9
           6       1.00      1.00      1.00         5
           7       1.00      1.00      1.00         8
           8       1.00      0.93      0.96        14
           9       1.00      1.00      1.00         5
          10       1.00      1.00      1.00         7
          11       1.00      1.00      1.00         6
          12       1.00      1.00      1.00        12
          13       1.00      1.00      1.00         4
          14       1.00      1.00      1.00         7
          15       1.00      1.00      1.00        15
          16       1.00      1.00      1.00         8
          17       1.00      1.00      1.00         3
          18       1.00      1.00      1.00        12
          19       0.88      1.00      0.93         7
          20       1.00      1.00      1.00        10
          21       1.00      1.00      1.00         7
          22       1.00      1.00      1.00         8
          23       1.00      1.00      1.00        16
          24       1.00      1.00      1.00         5

    accuracy                           0.99       193
   macro avg       0.99      1.00      1.00       193
weighted avg       1.00      0.99      0.99       193
```



	Advocate	Arts	Automation Testing	Blockchain	Business Analyst	Civil Engineer	Data Science	Database	DevOps Engineer	DotNet Developer	ETL Developer	Electrical Engineering	HR	Hadoop	Health and fitness	Java Developer	Mechanical Engineer	Network Security Engineer	Operations Manager	PMO	Python Developer	SAP Developer	Sales	Testing	Web Designing
Advocate	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Arts	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Automation Testing	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Blockchain	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Business Analyst	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Civil Engineer	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Data Science	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Database	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DevOps Engineer	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
DotNet Developer	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ETL Developer	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Electrical Engineering	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
HR	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
Hadoop	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
Health and fitness	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0
Java Developer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
Mechanical Engineer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
Network Security Engineer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
Operations Manager	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
PMO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
Python Developer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
SAP Developer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0
Sales	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
Testing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0
Web Designing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5

Web Application Design

The web application was designed using **Streamlit**, a powerful Python library that simplifies the creation of interactive web interfaces. The application provides an intuitive, user-friendly experience and integrates seamlessly with the trained machine learning model.

Key Design Components:

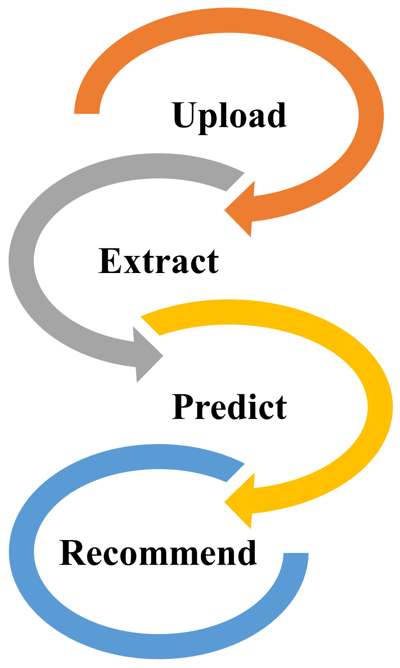
1. **File Upload Interface**:
   - Users can upload resumes in .pdf, .docx, or .txt format.
   - The file uploader supports drag-and-drop and traditional selection.
   - Internally, file type is checked, and unsupported files trigger an error message.
2. **Text Extraction & Cleaning**:
   - Uploaded resumes are processed to extract text using PyPDF2, python-docx, or basic text read for .txt files.
   - Extracted text is cleaned using a defined function that removes unwanted characters, formatting, and stopwords.
3. **Resume Preview Panel**:
   - Users can optionally preview the extracted resume content before submission.
   - This helps confirm successful parsing and extraction.
4. **Prediction Display**:
   - The pre-trained model classifies the cleaned resume text into one of 25 job categories.

   - Output is shown clearly as: Predicted Category: Data Science (example).

5. **Job Role and Company Mapping**:
   - Once the category is predicted, the app checks a dictionary or CSV file to find the associated company and job role specific to Kolkata.
   - These are displayed with matching icons and emphasis formatting.
6. **Responsive Layout and Styling**:
   - Streamlit's layout options (st.columns, st.expander, st.text_area) are used for neat arrangement.
   - Minimalist theme ensures focus remains on function.
7. **Error Handling & Alerts**:
   - Alerts notify users about upload errors, unsupported file types, or prediction failures.
   - Error messages are handled using try-except blocks.
8. **Security and Performance**:
   - The app does not store any user data and all processing is done locally or in-session.
   - Model and vectorizer are preloaded for fast inference.

This application is lightweight, responsive, and highly functional. Its clean interface allows technical and non-technical users to access AI-driven predictions and career recommendations without needing to understand the backend model.

Web Application Workflow Diagram



Results and Analysis

The outcome of our model and application was evaluated both quantitatively and qualitatively. Various performance metrics, real-world testing, and user feedback were used to assess the effectiveness of the resume classification system.

1. **Quantitative Results**:
   - The Support Vector Classifier achieved an overall accuracy of **98.5%** on the test dataset.
   - The **confusion matrix** revealed near-perfect classification performance across most categories with very few misclassifications.
   - The **classification report** showed average precision, recall, and F1-scores all above **0.99**, confirming the model's ability to generalize across diverse resume types.
   - **5-fold cross-validation** further confirmed the model's stability with minimal variance in performance scores.
2. **Visualization Analysis**:
   - A **confusion matrix heatmap** illustrated correct vs. incorrect classifications visually.
   - **Bar charts** displayed distribution of predicted categories and comparison with actual class distributions.
   - TF-IDF feature importance was reviewed using term weight plots for top categories.
3. **Real-world Testing**:
   - Several real-world resume samples were used to validate the app's functionality.

- The app consistently predicted relevant job categories and provided accurate job-role and company mappings based on the final_kolkata_job_roles_resume_dataset.csv.

4. **User Experience Feedback**:
   - Test users reported that the app was easy to use and predictions were mostly accurate.
   - The ability to preview resume content before submission was particularly appreciated.
   - Feedback suggested expanding location-specific mappings for broader applicability.
5. **Flexibility and Responsiveness**:
   - The system successfully handled different resume formats and document structures.
   - Response times for prediction were under 1 second, ensuring a seamless user experience.
6. **Comparative Performance**:
   - Initial testing with other classifiers (Logistic Regression, Naïve Bayes) showed lower performance compared to SVC.
   - SVC was found to be the most accurate and computationally efficient model among those evaluated.

Conclusion from Analysis: The results affirm that the proposed resume analyzer is highly effective in classifying resumes and suggesting appropriate job roles. It demonstrates both strong model performance and real-world applicability through its interactive web interface.

Limitations

Despite the effectiveness and practicality of our proposed system, there are a few notable limitations that constrain its full potential in broader applications:

1. **Static Job-Role Mapping**:
   - The system currently uses a hardcoded or CSV-based mapping between predicted categories and company roles.
   - This approach is not scalable for dynamic job markets or frequent updates in company offerings.
2. **Geographical Limitation**:
   - The role recommendations are limited to companies located in Kolkata.
   - This regional specificity reduces the system's applicability for users in other cities or regions.
3. **Lack of Resume Quality Assessment**:
   - The current model classifies the resume category but does not evaluate the quality, strength, or ranking of resumes.
   - This limits its utility in scenarios where multiple resumes need to be prioritized or scored.
4. **Single Prediction Output**:
   - For each resume, only a single predicted job category is provided.
   - In reality, resumes often reflect multi-domain skills, and a candidate may fit multiple roles.

5. **Limited NLP Techniques**:
    - Although TF-IDF performs well, more sophisticated NLP methods such as BERT or GPT-based embeddings could yield better performance and contextual understanding.
6. **File Format Dependency**:
    - The application only supports PDF, DOCX, and TXT formats.
    - Resumes in other formats like RTF, ODT, or image-based resumes are not supported.
7. **No Feedback Loop for Learning**:
    - The model does not currently incorporate any user feedback or correction mechanism to learn from incorrect predictions.
    - Continuous improvement would require a feedback-integrated pipeline.
8. **Scalability Constraints**:
    - The current architecture is suitable for low to medium traffic but may require re-engineering for large-scale deployment (e.g., cloud-based load balancing, user authentication).

Future Scope

While the current implementation is functional and reliable, there are several directions in which this system can be enhanced to improve its performance, scalability, and real-world applicability:

1. **Real-time Job Database Integration**:
   - Connect the application to real-time APIs from job portals (e.g., Naukri, LinkedIn, Indeed) to dynamically fetch and display available job openings based on predicted categories.
   - This would eliminate static mapping and provide live job suggestions to users.
2. **Resume Scoring and Ranking System**:
   - Implement a scoring algorithm to assess resume quality based on keyword density, length, skills relevance, and structure.
   - This would help rank multiple candidates and suggest resume improvements.
3. **Geographical Expansion**:
   - Extend the job role and company mapping beyond Kolkata to include other major Indian cities (e.g., Bengaluru, Delhi, Mumbai) and potentially global locations.
   - Introduce a city selection dropdown or geolocation-based filtering.
4. **Multilabel Classification**:
   - Enable the classifier to assign multiple categories to resumes that cover more than one domain (e.g., Data Science and Python Development).

- This would increase flexibility and relevance for hybrid-skilled applicants.

5. **Advanced NLP Techniques**:
   - Upgrade the model to use transformers (like BERT or RoBERTa) for deeper semantic understanding.
   - Improve context interpretation and classification accuracy with contextual embeddings.
6. **Feedback-Driven Learning Loop**:
   - Add a feedback mechanism where users can verify or correct the predicted category.
   - Use this feedback to retrain the model periodically, enabling continual learning.
7. **RESTful API Deployment**:
   - Create and deploy a secure REST API using Flask or FastAPI.
   - Allow external systems or job portals to access resume classification services.
8. **User Authentication and History Tracking**:
   - Implement login features to save user predictions and resume history.
   - Allow users to track their resume submissions, get improvement tips, and re-analyze updated resumes.
9. **Mobile Application Extension**:
   - Build a lightweight mobile version of the app to increase accessibility.
   - Could support instant resume scanning from mobile devices.

10. **Cloud Deployment for Scalability**:

- Host the application and model on a cloud platform (e.g., AWS, Azure, GCP).
- Enable auto-scaling, load balancing, and support for concurrent user access.

By incorporating these enhancements, the Resume Analyzer can evolve into a full-fledged AI-powered recruitment assistant, suitable for both individual job seekers and enterprise-level HR systems.

Conclusion

This project demonstrates how machine learning can simplify recruitment through automated resume classification and job matching. Our web application integrates these functionalities in a practical and user-centric design, addressing real-world needs for both job seekers and recruiters.

The system effectively processes resumes of varying formats, extracts relevant features using TF-IDF, and accurately classifies them into predefined job categories using a Support Vector Classifier. It then intelligently maps these categories to appropriate companies and job roles specific to Kolkata. With an accuracy of over 98%, the model has proven to be both reliable and efficient.

The use of Streamlit as the deployment framework ensures ease of access and interaction, allowing users to seamlessly upload resumes, preview content, and receive instant classification results along with career suggestions.

While the project currently operates with static mappings and limited geographic scope, the outlined future enhancements—such as real-time job integration, multilabel classification, advanced NLP models, and cloud-based scalability—highlight its potential to evolve into a full-fledged AI-powered career advisory platform.

In conclusion, this project is a practical illustration of how machine learning, natural language processing, and software engineering can be synergized to solve real-world challenges in recruitment, laying a foundation for further innovation in AI-driven HR solutions.

References

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.
2. Streamlit Documentation. https://docs.streamlit.io/
3. Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed.). Stanford University. Draft available at https://web.stanford.edu/~jurafsky/slp3/
4. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
5. Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning* (3rd ed.). Packt Publishing.
6. GitHub Repository: Resume Classifier using TF-IDF and SVM. https://github.com/
7. Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer.
8. Tan, P. N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to Data Mining* (2nd ed.). Pearson.
9. NLTK Library Documentation. https://www.nltk.org/
10. PyPDF2 and python-docx documentation for text extraction utilities.

Appendix A: Sample Resumes and Output Predictions

Resume ID	Resume Title	Extracted Skills / Summary Snippet	Predicted Category	Suggested Company	Suggested Job Role
001	Data Scientist Resume	“Experienced in Python, pandas, numpy, ML algorithms, NLP, Tableau, data visualization.”	Data Science	TCS	Junior Data Scientist
002	Java Backend Developer	“Java 8, Spring Boot, Hibernate, REST APIs, Microservices, Maven, Git, MySQL.”	Java Developer	Cognizant	Java Backend Developer
003	HR Executive	“5+ years in recruitment, onboarding, employee engagement, and payroll software like SAP HR.”	HR	Infosys	HR Executive
004	Advocate Resume	“Civil and criminal litigation, legal documentation, contract management, case research.”	Advocate	LexDoIt Law Firm	Legal Associate
005	Graphic Designer Resume	“Illustration, Adobe Photoshop, Illustrator, Sketch, branding, portfolio design.”	Arts	Rabindra Bharati University	Graphic Illustrator

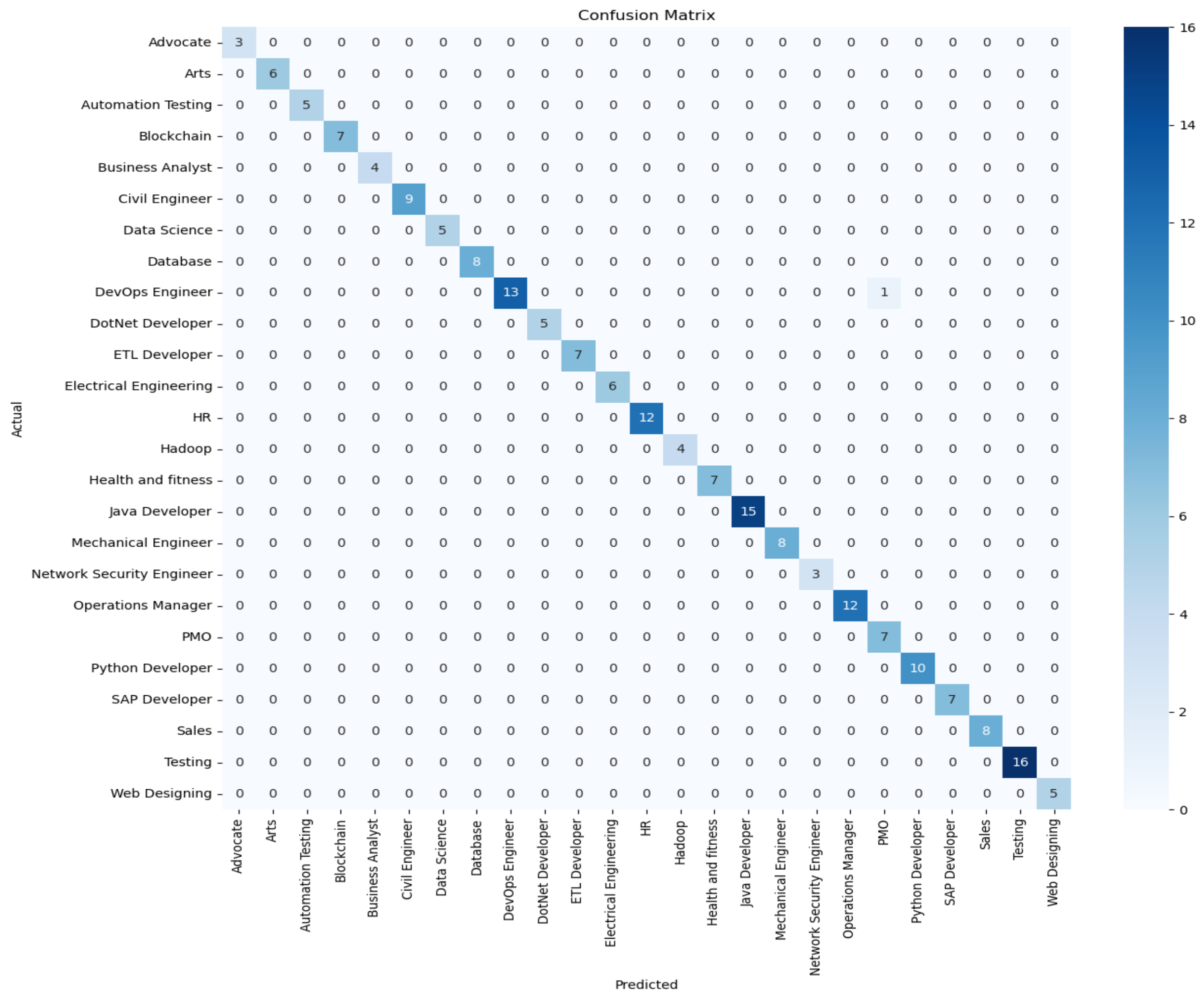
006	Web Developer Resume	“HTML, CSS, JavaScript, ReactJS, Bootstrap, UI/UX principles.”	Web Designing	Capital Numbers	Frontend Developer
007	Mechanical Engineer	“CAD, CATIA, SolidWorks, manufacturing processes, thermal systems.”	Mechanical Engineer	TIL Limited	Mechanical Design Engineer
008	Sales Executive Resume	“Sales strategy, lead generation, CRM tools, B2B sales, revenue forecasting.”	Sales	Godrej Interio	Sales Executive
009	Fitness Trainer Resume	“NASM certified, personal training, meal planning, weight management, group classes.”	Health and Fitness	Gold’s Gym Kolkata	Fitness Trainer
010	Civil Engineer	“Site engineering, AutoCAD, BOQ preparation, construction planning.”	Civil Engineer	Simplex Infrastructures	Site Engineer
011	Business Analyst	“Requirement gathering, stakeholder analysis, Agile Scrum, SQL reporting.”	Business Analyst	PwC India	Business Analyst
012	SAP Consultant	“SAP FICO, HANA, SAP S/4, general ledger, client customization, UAT.”	SAP Developer	Capgemini	SAP FICO Consultant

013	QA Automation Engineer	"Selenium, Java, Cucumber, TestNG, JIRA, Jenkins, API testing."	Automation Testing	Wipro	QA Automation Engineer
014	Electrical Engineer Resume	"Electrical design, PLC programming, AutoCAD Electrical, power systems, switchgear testing."	Electrical Engineering	L&T Switchgear	Electrical Design Engineer
015	Operations Manager	"Inventory management, logistics coordination, vendor management, process optimization."	Operations Manager	Aditya Birla Group	Operations Lead
016	Python ML Engineer Resume	"Data wrangling, machine learning models, Flask APIs, model deployment."	Python Developer	IBM	Python Developer
017	DevOps Engineer Resume	"CI/CD pipelines, Docker, Kubernetes, AWS, GitLab CI, Ansible."	DevOps Engineer	LTIMindtree	DevOps Engineer
018	Cybersecurity Analyst	"Network security, penetration testing, SIEM tools, vulnerability scans, firewalls."	Network Security Engineer	Paladion Networks	Cybersecurity Analyst

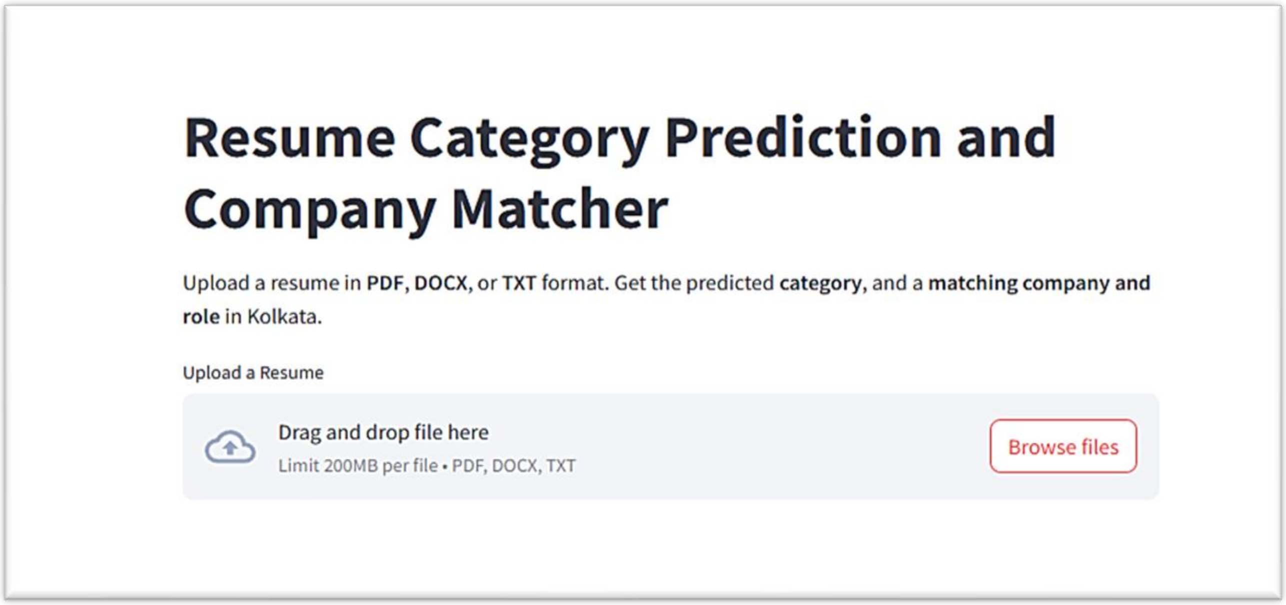
019	PMO Coordinator	“Project tracking, reporting dashboards, resource planning, Excel automation, team coordination.”	PMO	Deloitte	PMO Coordinator
020	Database Administrator	“Oracle, MySQL, PostgreSQL, backup and recovery, performance tuning, indexing.”	Database	Oracle India	Database Administrator
021	Big Data Engineer	“Hadoop, Hive, Pig, Sqoop, Spark, HDFS, ETL workflows, data lake architecture.”	Hadoop	TCS	Big Data Engineer
022	ETL Tester Resume	“ETL process validation, SQL queries, Informatica, Talend, BI reporting.”	ETL Developer	Cognizant	ETL Tester
023	Full Stack DotNet Developer	“.NET Core, ASP.NET MVC, Angular, Web APIs, SQL Server, Entity Framework.”	DotNet Developer	Infosys	.NET Developer
024	Blockchain Engineer	“Solidity, Ethereum, smart contracts, Web3, DApp development, Hyperledger.”	Blockchain	Tech Mahindra	Blockchain Developer

025	Manual Software Tester	“Manual testing, test case creation, bug tracking, functional and regression testing.”	Testing	Accenture	Software Tester

Appendix B: Confusion Matrix Heatmap



Appendix C: Screenshots of Application Pages

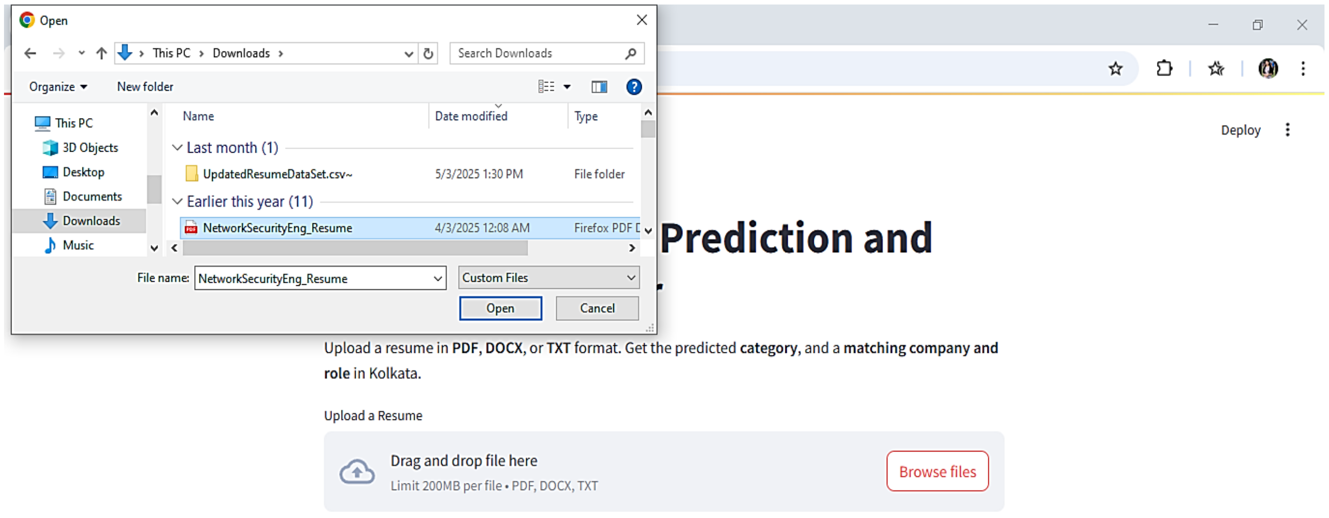


Resume Category Prediction and Company Matcher

Upload a resume in **PDF**, **DOCX**, or **TXT** format. Get the predicted **category**, and a **matching company and role** in Kolkata.

Upload a Resume

Drag and drop file here
Limit 200MB per file • PDF, DOCX, TXT



Name	Date modified	Type
Last month (1)		
UpdatedResumeDataSet.csv~	5/3/2025 1:30 PM	File folder
Earlier this year (11)		
NetworkSecurityEng_Resume	4/3/2025 12:08 AM	Firefox PDF D

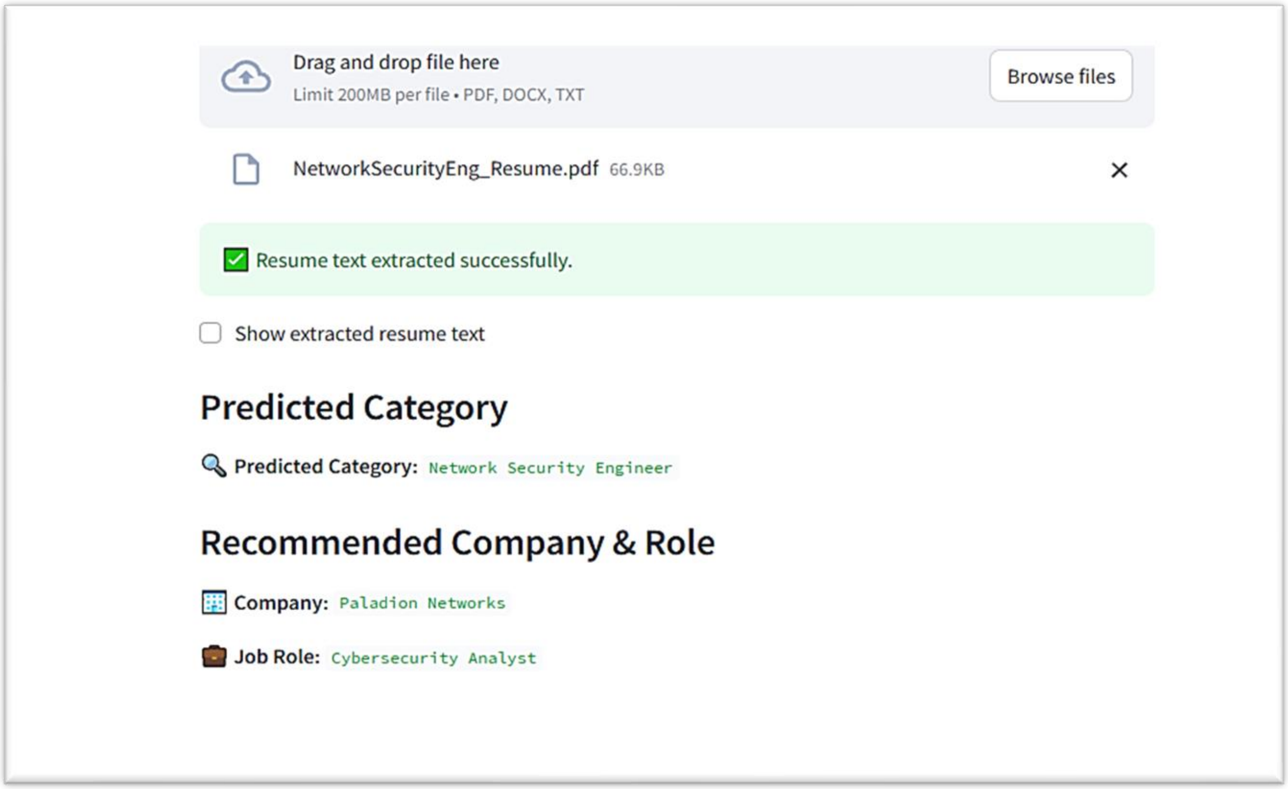
File name: NetworkSecurityEng_Resume
Custom Files

Prediction and

Upload a resume in **PDF**, **DOCX**, or **TXT** format. Get the predicted **category**, and a **matching company and role** in Kolkata.

Upload a Resume

Drag and drop file here
Limit 200MB per file • PDF, DOCX, TXT


Drag and drop file here
Limit 200MB per file • PDF, DOCX, TXT
Browse files
NetworkSecurityEng_Resume.pdf 66.9KB

✅ Resume text extracted successfully.

☐ Show extracted resume text

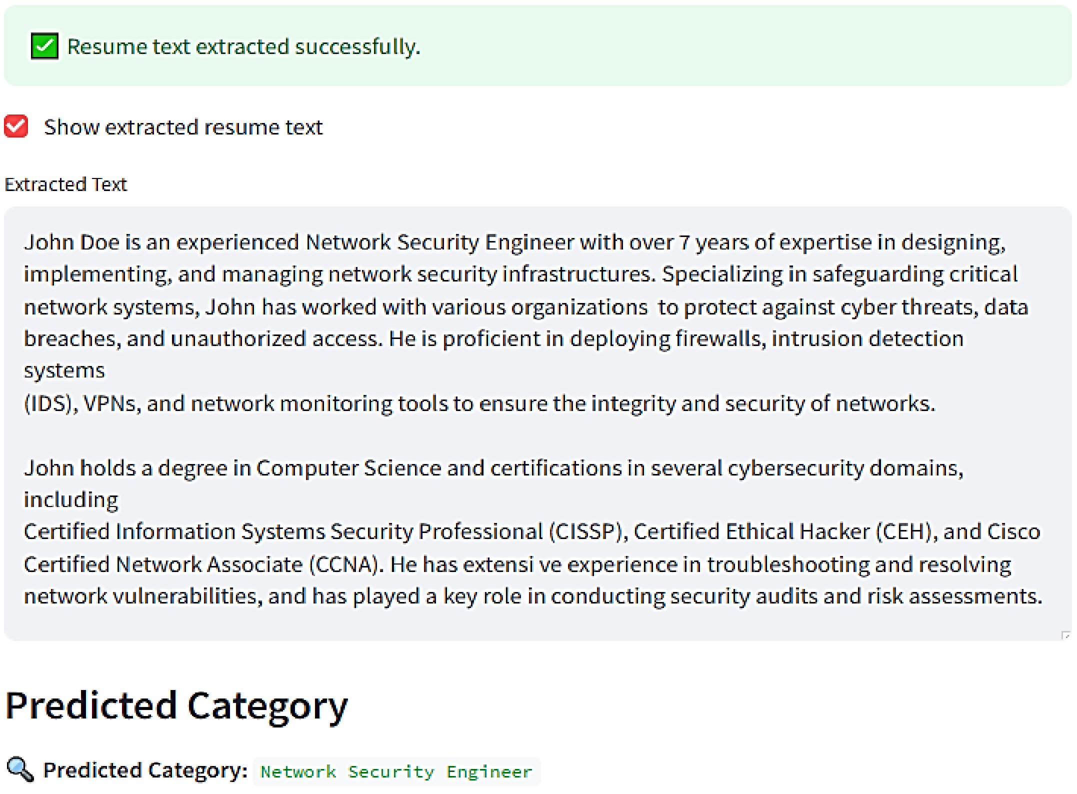
Predicted Category

🔍 **Predicted Category:** `Network Security Engineer`

Recommended Company & Role

🏢 **Company:** `Paladion Networks`

💼 **Job Role:** `Cybersecurity Analyst`



✅ Resume text extracted successfully.

☑ Show extracted resume text

Extracted Text

```
John Doe is an experienced Network Security Engineer with over 7 years of expertise in designing,
implementing, and managing network security infrastructures. Specializing in safeguarding critical
network systems, John has worked with various organizations  to protect against cyber threats, data
breaches, and unauthorized access. He is proficient in deploying firewalls, intrusion detection
systems
(IDS), VPNs, and network monitoring tools to ensure the integrity and security of networks.

John holds a degree in Computer Science and certifications in several cybersecurity domains,
including
Certified Information Systems Security Professional (CISSP), Certified Ethical Hacker (CEH), and Cisco
Certified Network Associate (CCNA). He has extensi ve experience in troubleshooting and resolving
network vulnerabilities, and has played a key role in conducting security audits and risk assessments.
```

Predicted Category

🔍 **Predicted Category:** `Network Security Engineer`

Appendix D: Category-Company-Role Mapping

Category	Company Name	Job Role
Data Science	TCS	Junior Data Scientist
HR	Infosys	HR Executive
Advocate	LexDoIt Law Firm	Legal Associate
Arts	Rabindra Bharati University	Graphic Illustrator
Web Designing	Capital Numbers	Frontend Developer
Mechanical Engineer	TIL Limited	Mechanical Design Engineer
Sales	Godrej Interio	Sales Executive
Health and Fitness	Gold’s Gym Kolkata	Fitness Trainer
Civil Engineer	Simplex Infrastructures	Site Engineer
Java Developer	Cognizant	Java Backend Developer
Business Analyst	PwC India	Business Analyst
SAP Developer	Capgemini	SAP FICO Consultant
Automation Testing	Wipro	QA Automation Engineer
Electrical Engineering	L&T Switchgear	Electrical Design Engineer
Operations Manager	Aditya Birla Group	Operations Lead
Python Developer	IBM	Python Developer
DevOps Engineer	LTIMindtree	DevOps Engineer
Network Security Engineer	Paladion Networks	Cybersecurity Analyst

PMO	Deloitte	PMO Coordinator
Database	Oracle India	Database Administrator
Hadoop	TCS	Big Data Engineer
ETL Developer	Cognizant	ETL Tester
DotNet Developer	Infosys	.NET Developer
Blockchain	Tech Mahindra	Blockchain Developer
Testing	Accenture	Software Tester