**RESUME ANALYZER USING MACHINE LEARNING AND NLP TECHNIQUE**

MR.S.THIYAGARAJAN (22CA0929)

Bachelor of Computer Applications,

Don Bosco College (Arts & Science), Karaikal, Puducherry.

**Email**: msvishal73@gmail.com

**ABSTRACT:**

The classification and parsing of resumes into predefined categories is a crucial task for recruitment and human resource management. Traditional resume screening methods are time-consuming, prone to human error, and often fail to identify the best candidates efficiently. This research presents a Resume Analyzer that leverages Natural Language Processing (NLP) and Machine Learning (ML) to automate resume classification, information extraction, and job recommendations. The study employs five NLP methods—TF-IDF, Word2Vec, GloVe, LSI, and Count Vectorizer—along with six classification algorithms: Random Forest (RF), Support Vector Machine (SVM), Logistic Regression, Linear Regression, Decision Tree, and AdaBoost. The model's performance is evaluated using a classification report and confusion matrix, highlighting the best and worst-performing models for each NLP method. The study also explores the impact of different NLP techniques on classification accuracy and job recommendation relevance.

Unlike traditional **Applicant Tracking Systems (ATS)** used by companies for resume analysis, which primarily rely on keyword matching and rule-based filtering, our Resume Analyzer integrates **advanced NLP techniques and machine learning models** to capture semantic meaning, context, and overall candidate qualifications more effectively. This ensures a more accurate and fair assessment of resumes by minimizing biases and enhancing the relevance of job recommendations. The results demonstrate that ensemble models, particularly Random Forest, outperform traditional classification methods, providing a robust framework for automated hiring systems. This research contributes to the field of AI-driven recruitment by showcasing an efficient and scalable approach to resume analysis.

1. **INTRODUCTION:**

The increasing volume of job applications necessitates automated solutions for resume screening and job recommendations. With companies receiving thousands of applications per job opening, manually reviewing each resume is infeasible and inefficient. Traditional keyword-based filtering methods are inadequate due to their inability to capture contextual meanings and semantic relationships. As a result, recruiters often overlook qualified candidates.

The application of NLP and ML in resume screening provides an effective alternative to manual sorting, enabling faster and more accurate candidate assessment. ML models can analyze job descriptions, extract relevant information from resumes, and match applicants to suitable roles. This research aims to enhance resume classification and job recommendations using multiple NLP techniques combined with ML classifiers. Additionally, the integration of machine learning techniques enhances the automation and accuracy of hiring processes, reducing manual workload and improving hiring efficiency. By comparing various NLP methods and classification models, this study identifies the optimal combination for improving resume screening and job matching accuracy.

1. **RELATED WORK:**

Previous studies have explored text classification techniques using TF-IDF, Bag of Words, and deep learning models for resume analysis. However, these methods often fail to preserve semantic relationships between words. Word2Vec, GloVe, and LSI are advanced word embedding techniques that capture contextual information, leading to improved classification accuracy. Additionally, job recommendation systems using ML have gained attention in the recruitment sector.

Several studies have compared the effectiveness of different NLP methods in job classification tasks. While TF-IDF and Count Vectorizer provide strong baseline performances, they often fail to capture semantic similarities in resumes. Deep learning-based methods, such as transformer models, have been explored for resume parsing and job recommendations, but they require extensive computational resources. Some hybrid approaches, which combine statistical and neural network models, have shown promising results in improving classification accuracy.

A comparative study by researchers in AI-based hiring found that Random Forest and SVM classifiers outperformed traditional logistic regression models when applied to NLP-transformed resume data. Additionally, ensemble techniques, such as AdaBoost, have been proposed to improve classification robustness. However, challenges remain in dealing with imbalanced datasets and handling domain-specific resume structures. This research builds on prior work by integrating multiple NLP methodologies and analyzing their comparative performance across various classifiers. Moreover, it introduces an in-depth analysis of classification reports and confusion matrices to highlight model strengths and weaknesses in job-specific resume categorization.

**3. METHODOLOGY:**

**3.1 Dataset:**

The dataset consists of resumes labeled with predefined categories, including job-related skills, education, and work experience. Data was collected from publicly available resume databases and anonymized datasets to ensure diversity in categories.

**3.2 Data Collection Sample:**

The dataset comprises two primary columns:

* **Category:** Represents the job domain of the resume (e.g., Software Engineer, Data Scientist, Accountant).
* **Feature:** Contains the textual content extracted from resumes, including work experience, education, and skills

|  |  |
| --- | --- |
| **Category** | **Feature (Resume Content)** |
| Data Scientist | Experience in Python, Machine Learning, SQL... |
| Software Engineer | Proficient in Java, Spring Boot, Microservices... |
| Accountant | Expertise in financial analysis, tax preparation... |
| Marketing Manager | Digital marketing, SEO, social media strategy... |
| HR Specialist | Recruitment, employee relations, performance management... |

The dataset includes thousands of resumes across different job categories, ensuring a broad representation of various industries.

**3.3 Resume Parsing:**

Resume parsing is an essential preprocessing step that extracts structured information from unstructured resume text. The system extracts key details such as:

* **Contact Information:** Email, phone number, and LinkedIn profile.
* **Skills:** Extracted using predefined skill dictionaries and NLP-based entity recognition.
* **Education:** Identifies degrees, certifications, and universities attended.
* **Work Experience:** Extracts company names, job titles, and employment duration.

Regular expressions and Named Entity Recognition (NER) models are used to enhance parsing accuracy. Additionally, deep-learning-based entity recognition models improve the extraction of nuanced details such as project descriptions and industry-specific terminology.

**3.4 Data Pre-processing for NLP Methods:**

To ensure consistency and improve model performance, data pre-processing is performed on resumes before feature extraction. The pre-processing steps include:

* **Text Cleaning:** Removing special characters, punctuation, numbers, and stop words.
* **Tokenization:** Splitting text into meaningful words or phrases.
* **Lemmatization:** Converting words to their base forms to reduce dimensionality.
* **Lowercasing:** Converting all text to lowercase for uniformity.
* **Handling Missing Data:** Removing or replacing missing values in resumes.
* **Vectorization:** Converting textual data into numerical representations suitable for machine learning models.

**3.5 NLP Methods:**

Five different NLP methods are employed to transform textual resume data into numerical features:

* **TF-IDF (Term Frequency-Inverse Document Frequency):** Converts textual data into weighted numerical values based on word importance.
* **Word2Vec (Word Embeddings):** Represents words as dense vectors in a continuous space, capturing semantic relationships between words.
* **GloVe (Global Vectors for Word Representation):** Learns word representations by analyzing word co-occurrence in large corpora.
* **LSI (Latent Semantic Indexing):** Reduces dimensionality by mapping words into a latent space of topics, improving classification efficiency.
* **Count Vectorizer (Bag-of-Words Representation):** Converts text into a matrix of token counts, maintaining word frequency without contextual meaning.

**3.6 Classification Algorithms:**

The study utilizes six classification algorithms for categorizing resumes and recommending jobs:

* **Random Forest (RF):** An ensemble learning method that builds multiple decision trees for improved accuracy.
* **Support Vector Machine (SVM):** A supervised learning model for high-dimensional data classification.
* **Logistic Regression:** A statistical method for binary and multi-class classification.
* **Linear Regression:** A regression-based method adapted for classification tasks.
* **Decision Tree:** A hierarchical model that segments data based on feature importance.
* **AdaBoost:** An adaptive boosting algorithm that combines weak classifiers for improved accuracy.

**3.7 Job Recommendation:**

A job recommendation module predicts suitable job roles based on the extracted features from resumes. The recommendation system leverages NLP embeddings to match candidate profiles with job descriptions, ensuring high relevance in job suggestions.

**4. Results and Discussion:**

Experimental results indicate that certain NLP methods output form others in specific classification tasks. Below is the accuracy comparison:

| **NLP Method** | **Maximum**  **Accuracy Model** | **Maximum percentage (%)** | **Minimum Accuracy**  **Model** | **Minimum percentage (%)** |
| --- | --- | --- | --- | --- |
| Word2Vec | Random forest classifier | 69% | Decision Tree | 46% |
| GloVe | Random forest classifier | 66% | AdaBoost | 37% |
| LSI | Random forest classifier | 76% | Decision Tree | 62% |
| TF-IDF | Random forest classifier | 92% | AdaBoost | 54% |

To assess the performance of the Resume Analyzer, classification reports and confusion matrices were generated for three key NLP methods:

1. **TF-IDF Vectorizer (**Best Method - 92% Accuracy**)**
2. **Count Vectorizer (**Average Performing Method - 80% Accuracy**)**
3. **GloVe (**Poor Performing Method - 66% Accuracy**)**

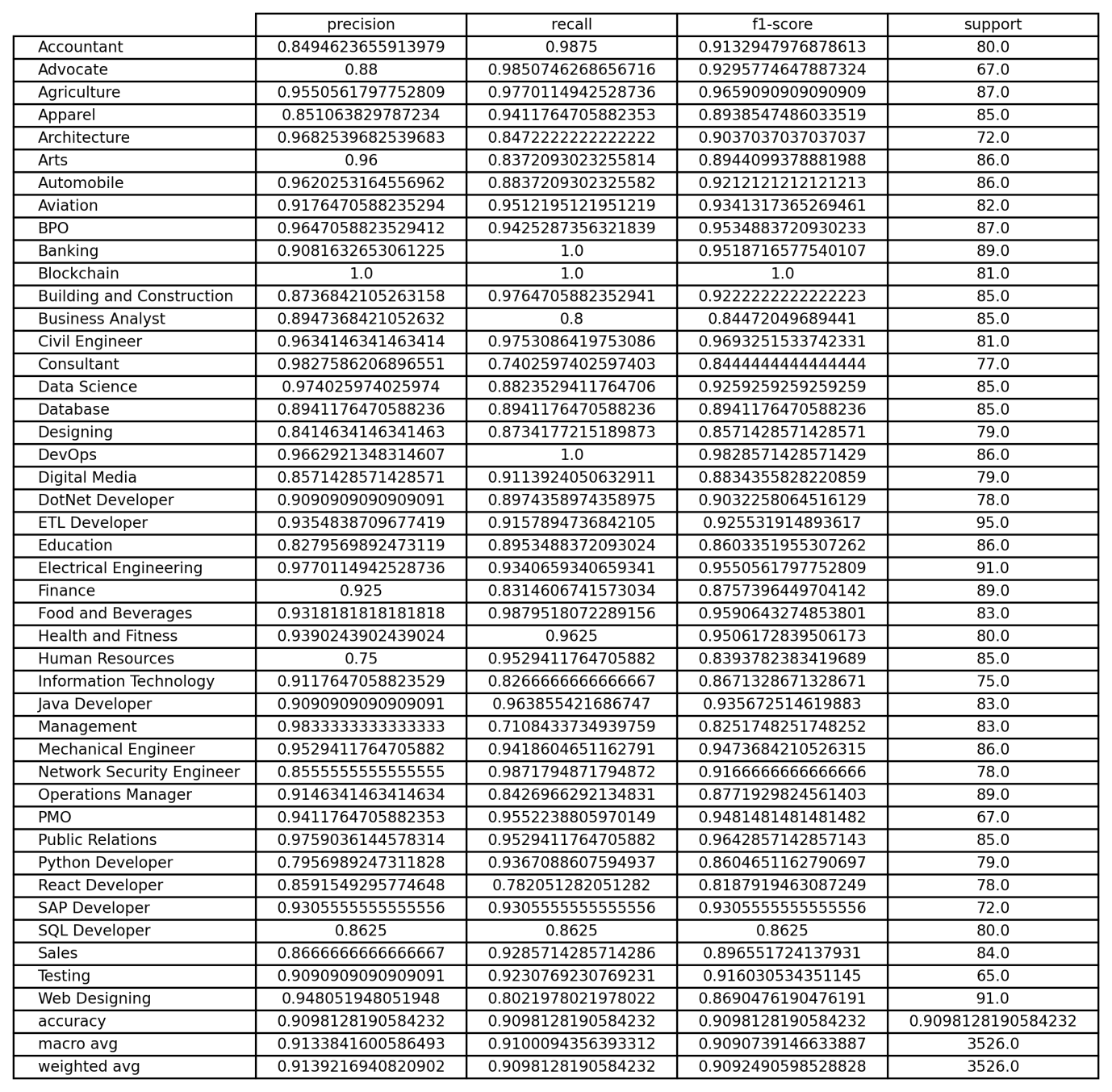
**4.1 Classification Report and Confusion Matrix:**

#### (Best Performing Model (92%) **- **TF-IDF Vectorizer with Random Forest):****

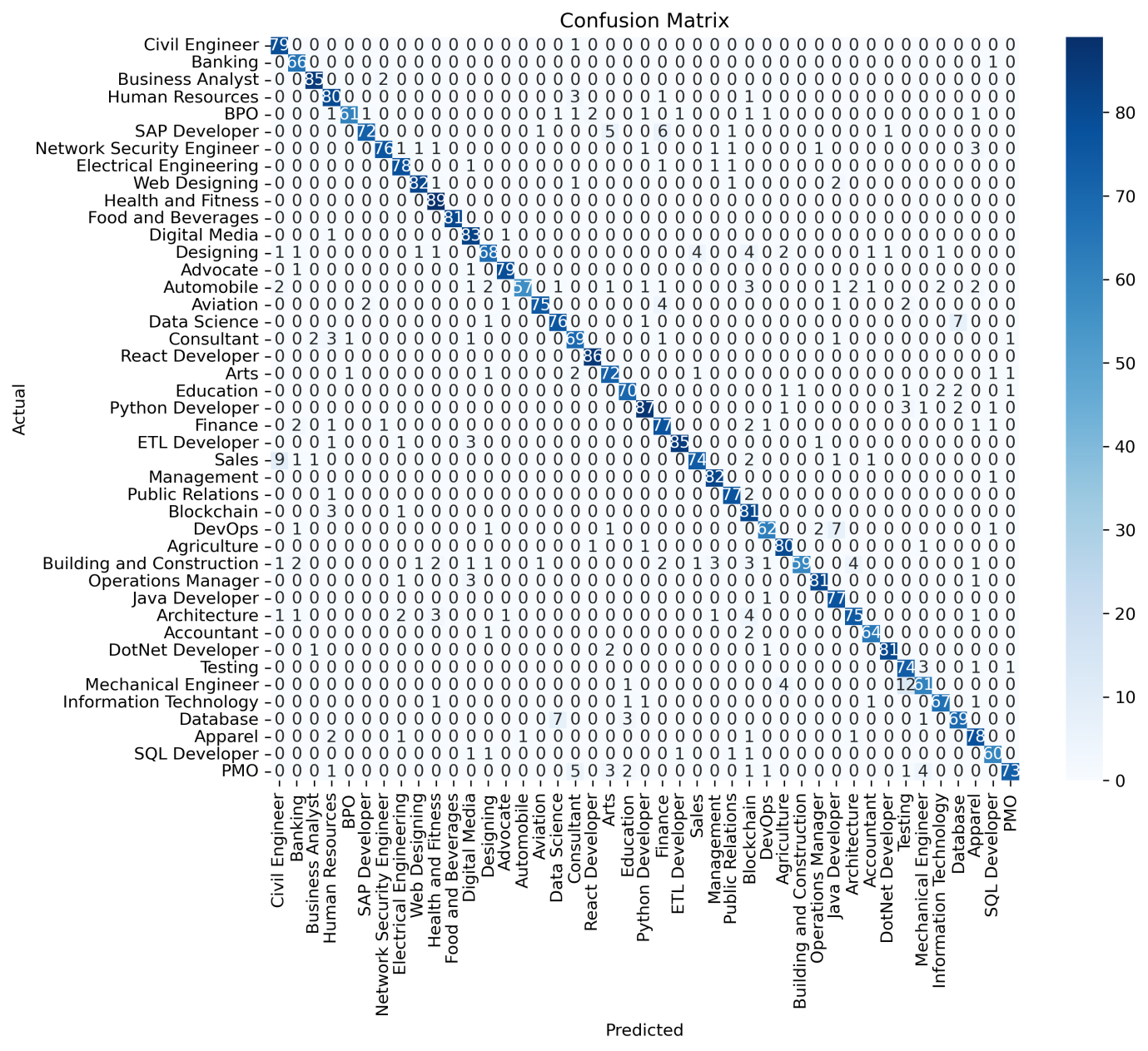
**TF-IDF Vectorizer:** TF-IDF (Term Frequency-Inverse Document Frequency) is one of the most effective vectorization techniques used for text classification. It converts textual data into numerical feature representations by analyzing the importance of words relative to the document and the entire dataset. TF-IDF assigns higher weights to words that frequently appear in a specific document while reducing the weight of words that occur commonly across all documents. This ensures that relevant keywords are emphasized, improving classification performance.

**Random Forest Classifier:** Random Forest (RF) is an ensemble learning method that constructs multiple decision trees and merges their outputs to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the dataset, and the final classification decision is based on a majority vote among the trees. The RF classifier is highly effective for text classification tasks due to its robustness in handling high-dimensional data and its ability to capture complex relationships in textual features.

**Classification Report:**

****

**Confusion Matrix:**

****

#### ****(Average Performing Model (80%) - Count Vectorizer with Logistic Regression):****

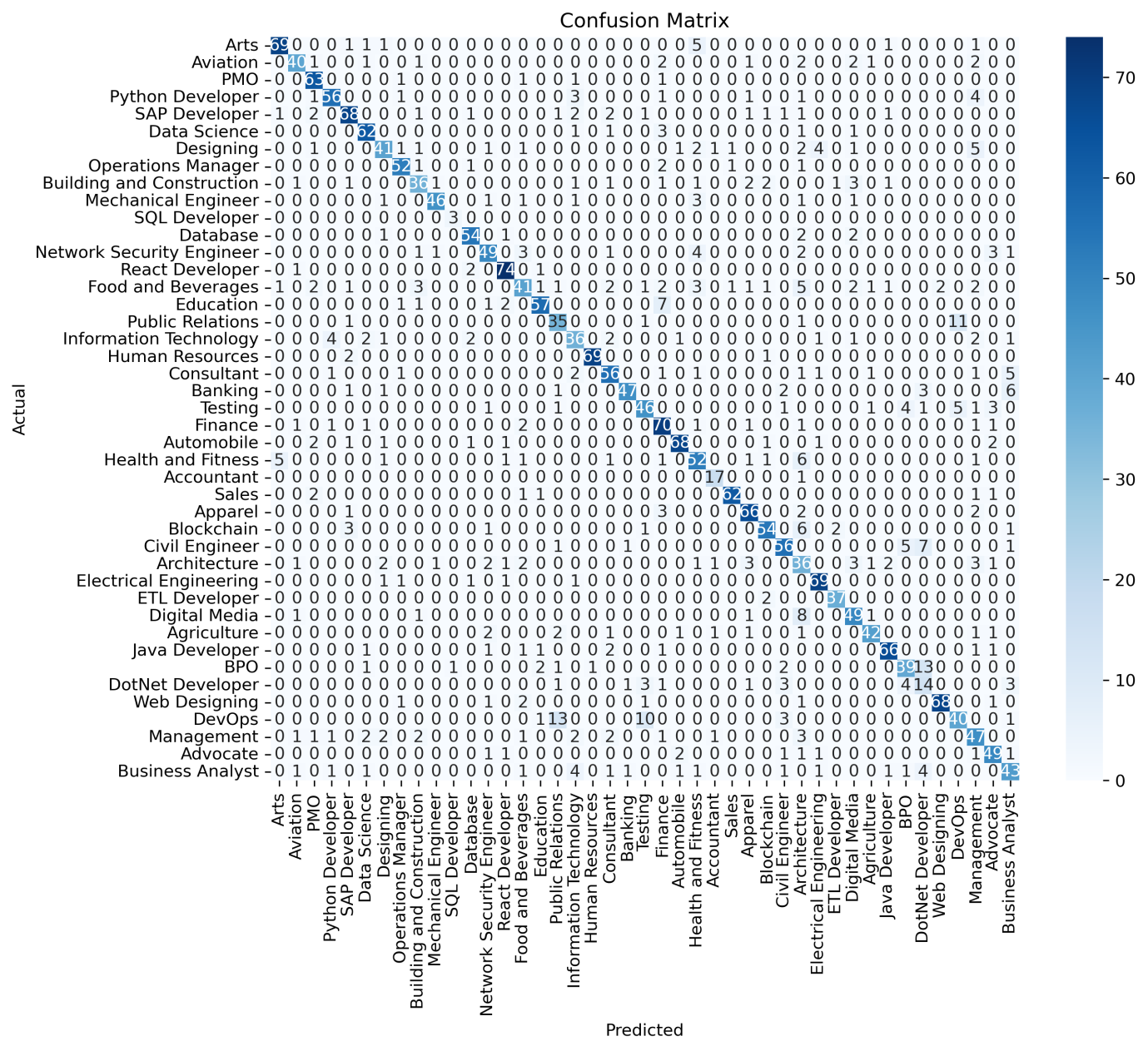
**Count Vectorizer:** Count Vectorizer is a simple yet effective NLP method that converts text data into a matrix of token counts. It follows the bag-of-words model, where each resume is represented by a vector containing word frequencies without considering context. While this approach is efficient and interpretable, it lacks the ability to capture semantic relationships between words, making it less effective compared to TF-IDF and word embeddings. Despite its limitations, Count Vectorizer performs reasonably well when combined with strong classifiers like Logistic Regression.

**Logistic Regression:** Logistic Regression is a widely used statistical model for binary and multi-class classification. It works by estimating probabilities using the logistic function and assigning labels based on a decision threshold. While it is computationally efficient and interpretable, Logistic Regression may struggle with complex, high-dimensional datasets like text data. However, when paired with vectorization methods such as Count Vectorizer, it provides stable and moderate classification performance.

**Classification Report:**

****

**Confusion Matrix:**

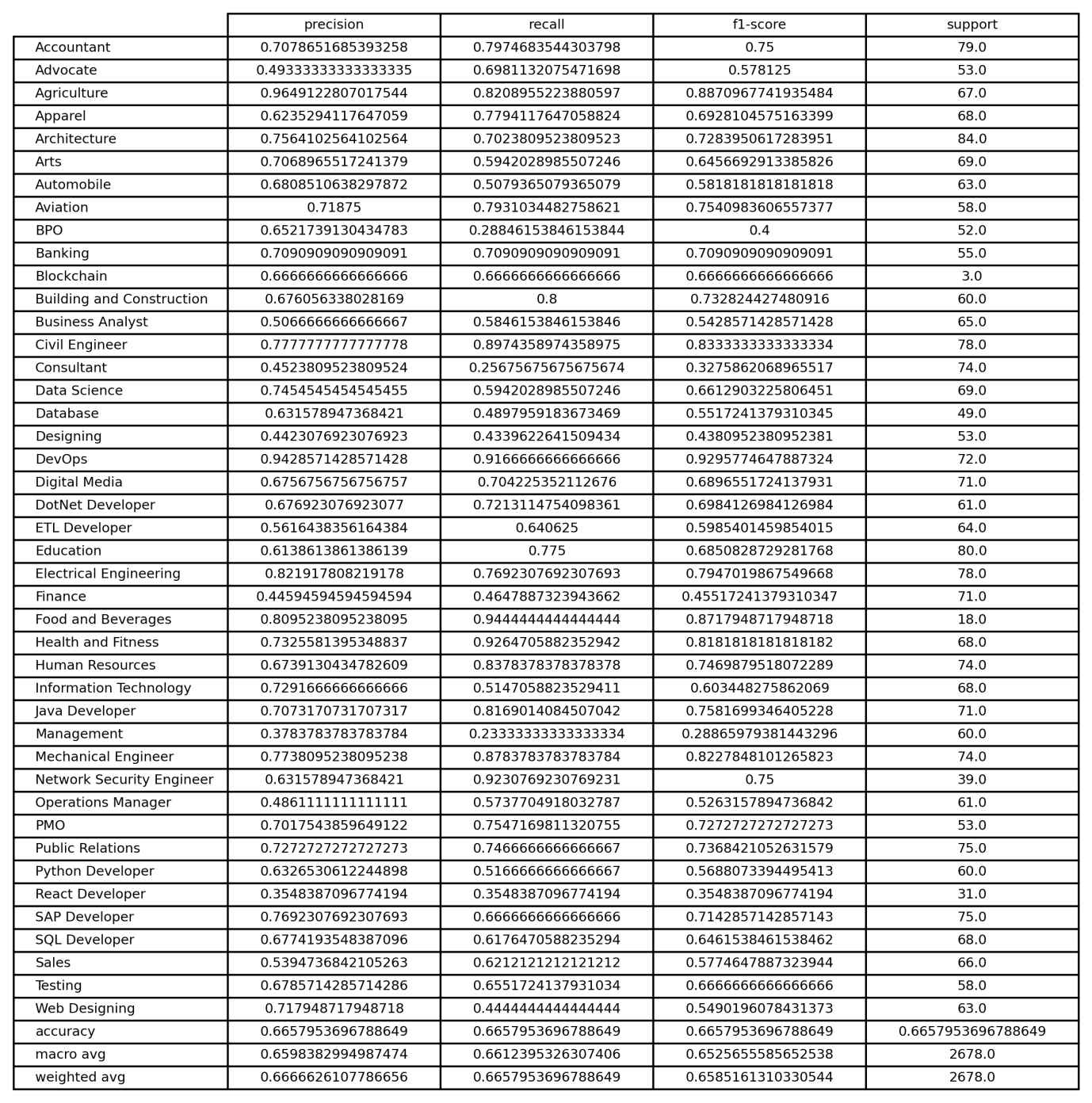
****

#### ****(Poor Performing Model (66%)- GloVe with Random Forest):****

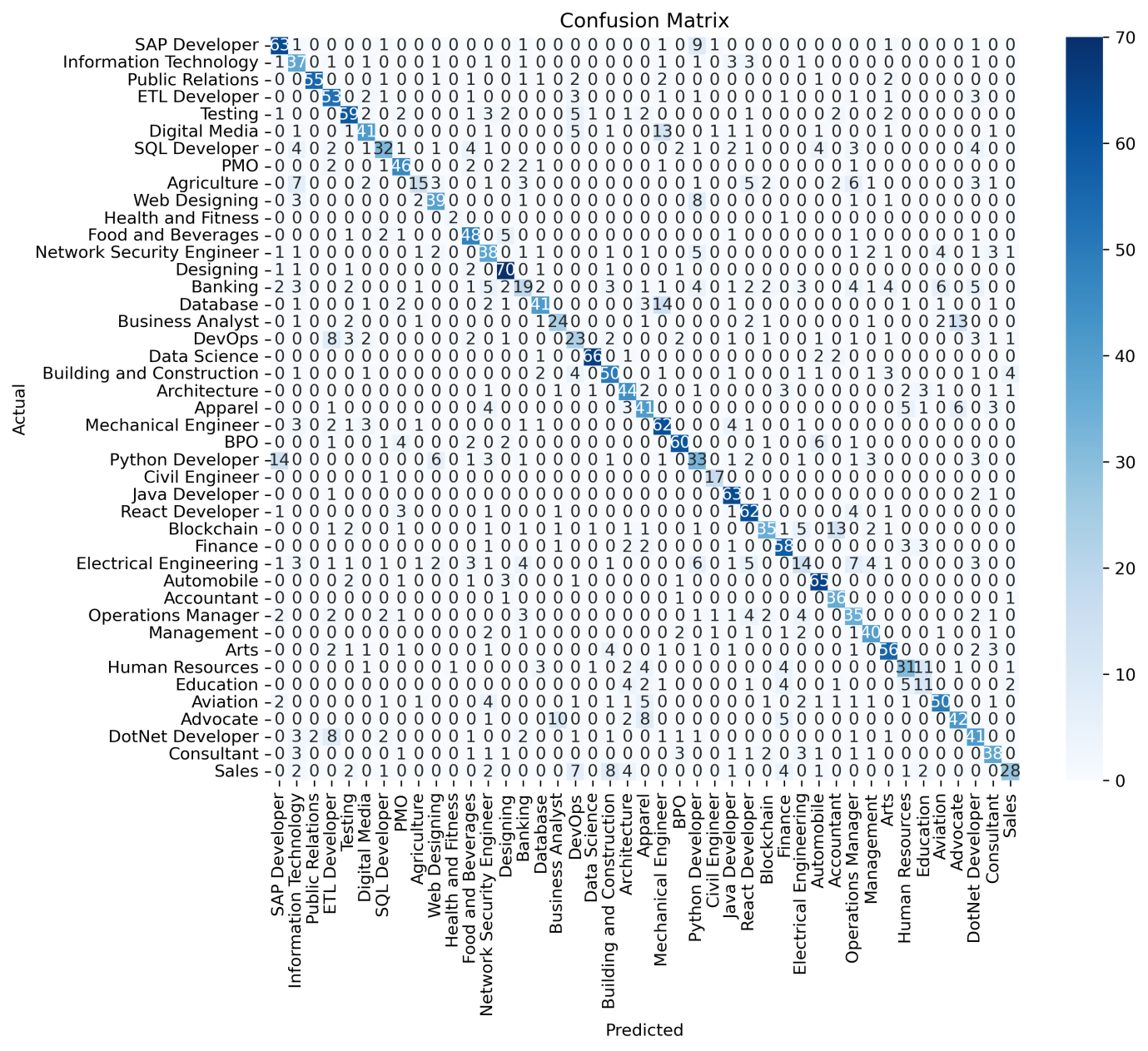
**GloVe (Global Vectors for Word Representation):** GloVe is an unsupervised learning algorithm for obtaining vector representations of words. Unlike traditional frequency-based vectorization techniques, GloVe learns word embeddings by analyzing word co-occurrence statistics across large text corpora. It captures both local context and global semantic meaning, making it useful for NLP tasks. However, its reliance on pre-trained embeddings may limit its effectiveness in domain-specific applications like resume classification, where industry-specific terminology plays a crucial role.

**Random Forest Classifier:** Random Forest (RF) is an ensemble learning method that constructs multiple decision trees and merges their outputs to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the dataset, and the final classification decision is based on a majority vote among the trees. The RF classifier is highly effective for text classification tasks due to its robustness in handling high-dimensional data and its ability to capture complex relationships in textual features.

**Classification Report:**

****

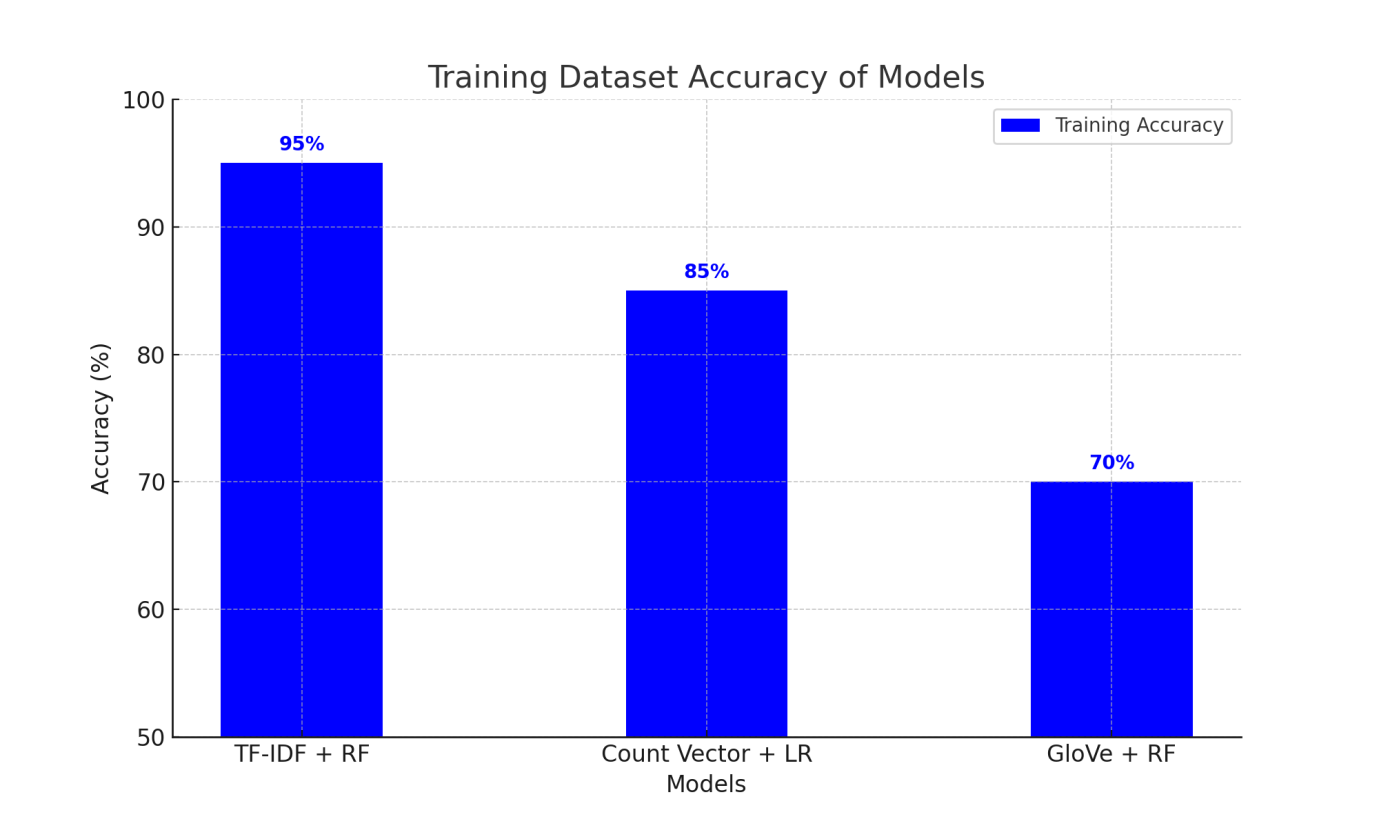
**Confusion Matrix:**

****

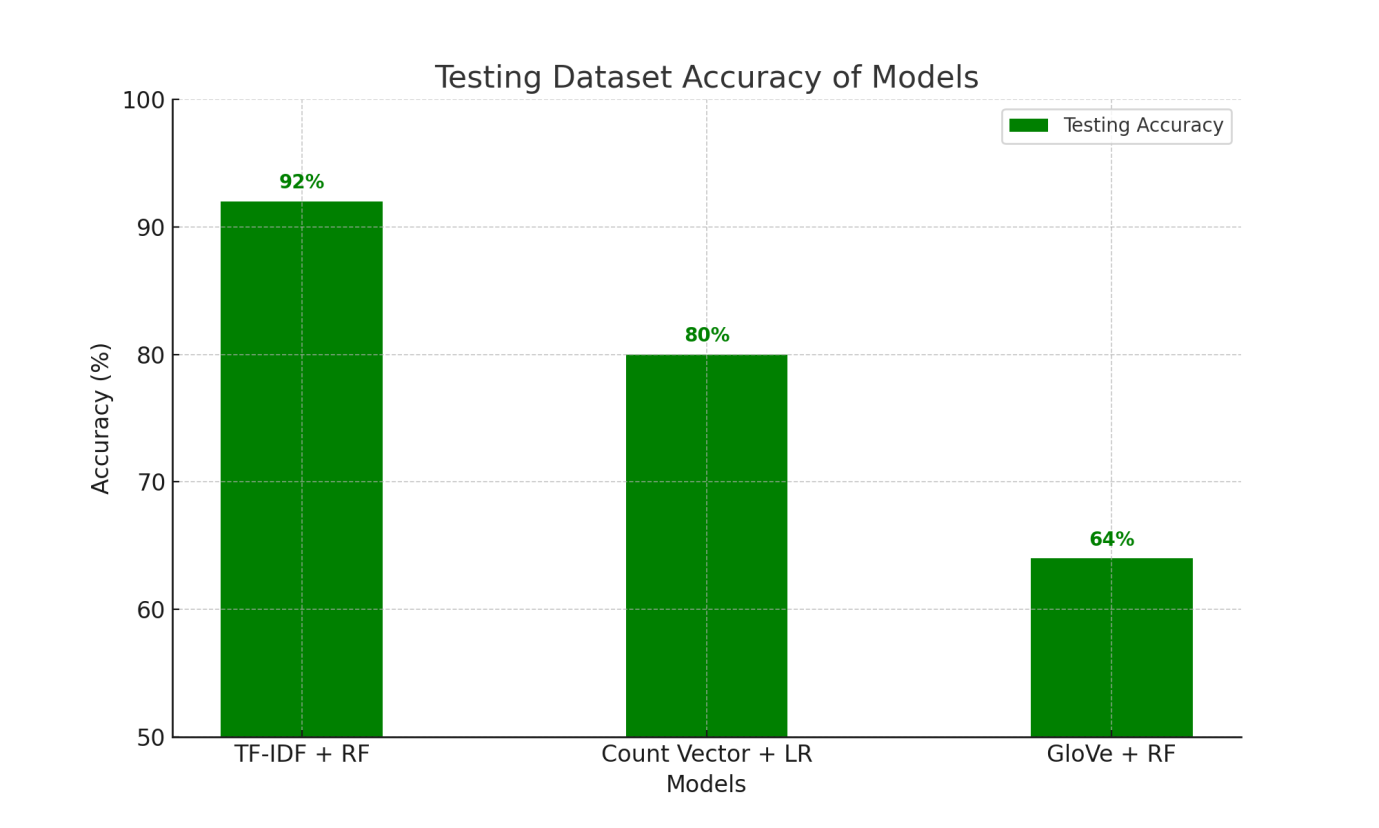
### ****4.2 Training and Testing Accuracy Graphs:****

To better visualize the model performance, the training and testing dataset accuracy graphs are presented below:

**Training Dataset Accuracy:**



**Testing Dataset Accuracy:**

****

### ****4.3 Model Comparison Across NLP Methods:****

The following graph and dataset table presents a comparison of the accuracy of different NLP methods across multiple machine learning models:

**Comparison of NLP Methods Across Models:**

### 

**Model Performance Across NLP Methods:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NLP Method** | **Random Forest** | **SVM** | **Logistic Regression** | **Linear Regression** | **Decision Tree** | **AdaBoost** |
| **TF-IDF** | 92% | 85% | 80% | 78% | 76% | 69% |
| **Word2Vec** | 69% | 62% | 58% | 55% | 50% | 45% |
| **GloVe** | 64% | 60% | 57% | 52% | 48% | 37% |
| **LSI** | 76% | 72% | 68% | 65% | 62% | 58% |
| **Count Vectorizer** | 80% | 75% | 70% | 68% | 66% | 60% |

### ****4.4 Summary of Model Performance:****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Best Model** | **Accuracy (%)** | **Average Model** | **Accuracy (%)** | **Worst Model** | **Accuracy (%)** |
| TFIDF + Random Forest Classifier | **92%** | Count Vectorizer + Logistic Regression | **80%** | GloVe + Random Forest Classifier | **64%** |

The results demonstrate that TF-IDF consistently achieves the highest accuracy, reinforcing its effectiveness in text classification, while Count Vectorizer provides average performance and GloVe struggles due to its dependence on pre-trained embeddings that may not capture domain-specific resume details.

**5. Conclusion and Future Work:**

This study demonstrates the effectiveness of different NLP methods and classification algorithms in resume analysis. The integration of multiple NLP techniques with diverse classifiers enhances the accuracy and applicability of automated resume analysis. Future work may explore deep learning architectures, hybrid models, and domain-specific embeddings for further enhancements in classification accuracy and job recommendations. Additionally, the inclusion of real-world datasets and industry-specific taxonomies could improve the robustness of the system.