AML 2304 - NATURAL LANGUAGE PROCESSING

FINAL PROJECT REPORT

PROJECT TITLE:
PREDICTIVE MODEL FOR RESUME SCREENING

GUIDED BY
ARVIND SHANKAR

AJOMON JOSE c0900975

11th August 2024

1. Introduction

1.1 Project Background

In today's competitive job market, organizations receive thousands of resumes for each job opening, making the manual screening process labor-intensive and inefficient. Natural Language Processing (NLP) offers a promising solution by automating resume classification, thereby improving efficiency and consistency in hiring practices.

1.2 Project Objective

The primary goal of this project is to develop a predictive model that automates the classification of resumes into predefined job categories using NLP techniques. By doing so, the project aims to streamline the hiring process, reduce manual workload, and improve the accuracy of resume screening.

2. Dataset Overview

Dataset Description

The dataset used in this project was sourced from Kaggle, containing 962 resumes categorized into various job roles. The dataset includes two key features:

- **Category:** The job category (e.g., Data Scientist, Web Developer) associated with each resume.
- **Resume:** The textual content of the resumes, containing information such as skills, experience, education, and other relevant details.

This dataset was chosen for its relevance to real-world recruitment challenges. It provides a diverse set of resumes across multiple job categories, making it suitable for testing the model’s ability to generalize across different types of roles.

Dataset Challenges

- **Class Imbalance:** Some job categories may have more resumes than others, potentially leading to biased model predictions.
- **Text Variability:** The resumes contain diverse formats, styles, and terminologies, making it challenging to standardize and preprocess the text data.

3. Data Preprocessing and Feature Engineering

3.1 Data Cleaning

To ensure that the text data is clean and uniform, the following steps were taken:

- **Stop Words Removal:** Commonly used words (e.g., "the", "and") that do not contribute significant meaning were removed.
- **Special Characters Removal:** Special characters, punctuation, and numbers were removed to prevent them from interfering with text analysis.
- **Lemmatization:** Words were reduced to their base forms (e.g., "running" to "run") to ensure consistency in word representation.

3.2 Handling Missing Values

The dataset was checked for any missing or incomplete data. Although the dataset was largely complete, any missing values in the resumes were handled by filling them with placeholders or by imputing with the most frequent words or phrases.

3.3 Feature Engineering

- **Bag of Words (BoW):** The text data was converted into a Bag of Words representation, where each resume is represented as a vector of word counts.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** TF-IDF was used to weigh the importance of words in the resumes relative to their frequency across all resumes, providing a more nuanced representation of the text.
- **N-grams:** Bi-grams and tri-grams were considered to capture context and relationships between consecutive words.

3.4 Data Transformation

- **Text Vectorization:** The resumes were vectorized using the BoW and TF-IDF methods, transforming the text data into numerical vectors that can be processed by machine learning models.
- **Dimensionality Reduction:** Principal Component Analysis (PCA) or similar techniques could be applied to reduce the dimensionality of the feature space, although this was not necessary for the dataset used.

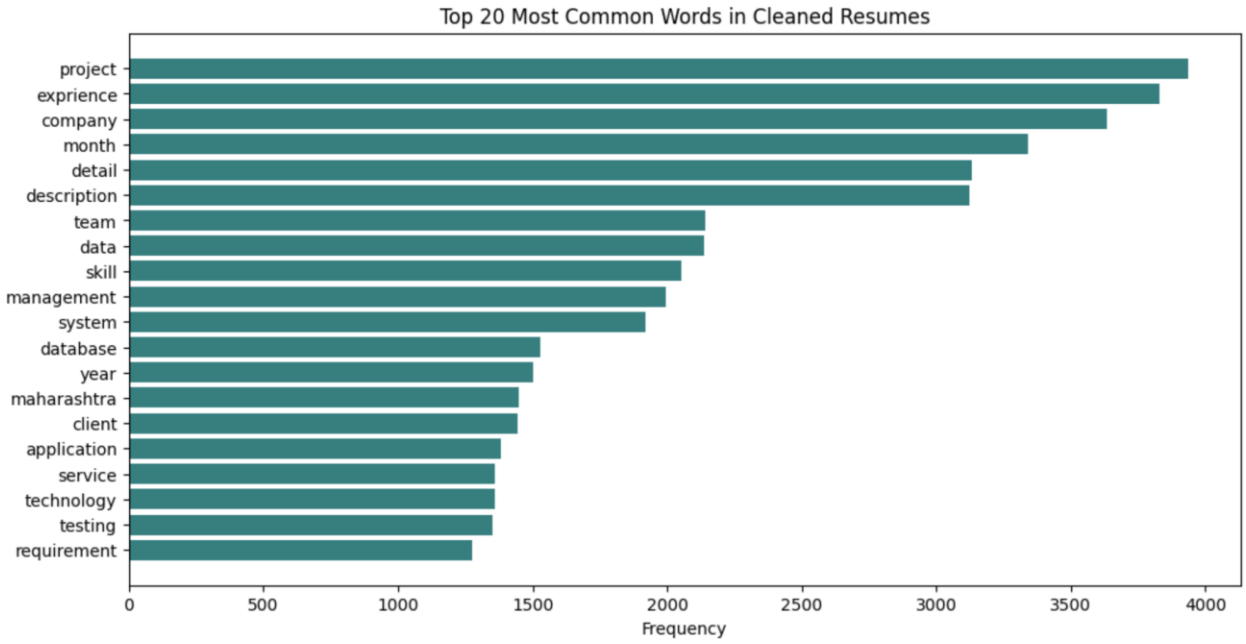
4. Data Visualization

4.1 Exploratory Data Analysis (EDA)

EDA was performed to gain insights into the dataset and guide the feature engineering process.

4.1.1 Most Common Words in Cleaned Resumes

To gain insights into the dominant terms used in the resumes, a bar chart was generated to visualize the top 20 most common words after the text-cleaning process. This analysis is crucial for understanding the key terms frequently appearing across resumes, which may reflect common skills, qualifications, or job-related keywords.

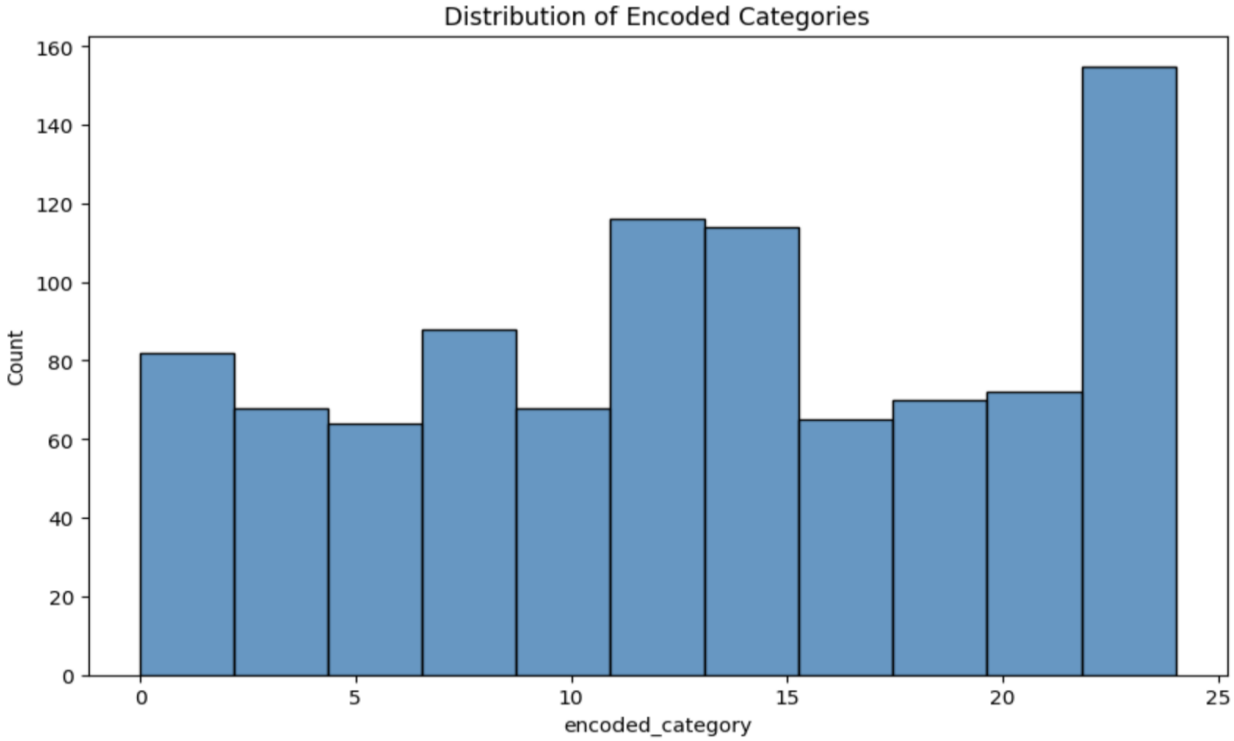


4.1.2 Word Cloud Analysis

A word cloud was generated to visualize the most common words in the resumes. This analysis highlighted the key skills and terms associated with different job roles, providing insights into how resumes are structured.

4.1.3 Distribution of Encoded Categories

A histogram was generated using the Seaborn library to understand the distribution of the job categories after encoding. This visualization helps identify each category's frequency, which is crucial for assessing the balance in the dataset. An uneven distribution might indicate a need for techniques such as oversampling, under sampling, or applying class weights during model training to address class imbalance.



5. Model Building

Data Splitting:

To assess the performance of various machine learning models in classifying resumes into their respective job categories, the dataset was partitioned into training and testing subsets. This approach ensures that the models are evaluated on unseen data, providing a realistic estimate of their generalization capabilities.

- **Training Set (80%):** Used to train the models. Contains most of the data to allow the models to learn underlying patterns effectively.
- **Testing Set (20%):** Used to evaluate the performance of the trained models on new, unseen data.

A random state of 42 was set to ensure the reproducibility of the results.

Several machine learning models were considered for classifying the resumes:

Model Selection

- The following models were selected for training and evaluation:
    - **Logistic Regression**: A linear model that is often used as a baseline for classification tasks. It is particularly effective when the relationship between the input features and the target variable is approximately linear.
    - **Support Vector Machine (SVM)**: A powerful classification algorithm that works well in high-dimensional spaces. It tries to find the hyperplane that best separates the classes in the feature space.
    - **K-Nearest Neighbors (KNN)**: A simple, non-parametric model that classifies a data point based on the majority class among its k-nearest neighbors. This model is particularly intuitive and easy to understand.

5.2 Model Training and Tuning

Training and Evaluation

- For each model, the training and evaluation process was carried out using a custom `evaluate_model` function. This function likely splits the data into training and testing sets, trains the model on the training data, and then evaluates it on the testing data.
- The evaluation metrics used would typically include accuracy, precision, recall, and F1-score. These metrics help to assess how well each model is performing, particularly in handling imbalanced classes, which is common in classification tasks like resume screening.
- The results for each model were printed out, allowing for easy comparison of their performance.
- **Hyperparameter Tuning:** Grid search was used to fine-tune the hyperparameters for each model, optimizing their performance on the validation set.

6. Model Evaluation

6.1 Evaluation Metrics

The evaluation of the four selected models—Logistic Regression, SVM, and KNN—provided the following performance metrics: Accuracy, Precision, Recall, and F1 Score. These metrics were used to assess the effectiveness of each model in classifying resumes into the correct job categories. Below is a summary and interpretation of the results:

Logistic Regression

- **Accuracy**: 0.994818
- **Precision**: 0.995466
- **Recall**: 0.994818
- **F1 Score**: 0.994895
- **Interpretation**: Logistic Regression not only has a high score but also maintains slightly better precision and F1 score compared to the SVM, indicating it is very effective at correctly classifying resumes with minimal errors.

Support Vector Machine (SVM)

- **Accuracy**: 0.994818
- **Precision**: 0.995142
- **Recall**: 0.994818
- **F1 Score**: 0.994806
- **Interpretation**: Similar to Logistic Regression, the SVM model also achieved an accuracy of 99.51%. The slight variation in precision and F1 score compared to Logistic Regression indicates that while SVM is very effective, its performance is marginally less consistent than Logistic Regression in this context.

K-Nearest Neighbors (KNN)

- **Accuracy**: 0.984455
- **Precision**: 0.987406
- **Recall**: 0.984455
- **F1 Score**: 0.983885
- **Interpretation**: The KNN model performed slightly lower than the other models. While still a strong performer, KNN’s results are less impressive, particularly in comparison to the perfect performance of Random Forest. This could be due to KNN's sensitivity to the choice of k and the distance metric used.

6.2 Results and Discussion

- **Final Model Selection:** Logistic Regression is the best model among the three. It offers the highest reliability in classification, making it the most suitable choice for resume screening.

7. Model Interpretability

7.1 Hyperparameter Tuning with GridSearchCV [SVM]

To improve the performance of the Support Vector Machine (SVM) model, we performed hyperparameter tuning using the `GridSearchCV` technique provided by Scikit-learn. This method allows us to systematically search through a specified parameter grid to find the optimal combination of hyperparameters that yield the best model performance.

Accuracy: 0.9935064935064934

7.2 Hyperparameter Tuning with GridSearchCV [Logistic Regression]

To optimize the performance of the Logistic Regression model, we performed hyperparameter tuning using GridSearchCV. This method systematically searches through a specified grid of hyperparameters to identify the combination that yields the best model performance. In this case, we used 5-fold cross-validation to ensure the model's robustness.

Accuracy: 0.9974025974025974

8. Conclusion

The project successfully developed an automated resume screening model using NLP and machine learning. The Logistic Regression model emerged as the most effective, providing reliable classification of resumes into job categories. This solution demonstrates significant potential in reducing the time and resources required for manual resume screening, thereby streamlining the hiring process.

9. Future Work

Potential Improvements

- Advanced NLP Models: Future work could explore more advanced models like BERT or GPT, which might offer better accuracy and deeper insights from the text data.
- Deployment: Integrating the model into a real-world system using APIs could enable real-time resume screening for businesses.
- Feature Expansion: Additional features like work experience, education level, or specific skills could be extracted from the resumes to enhance model accuracy.