Project Report
Automated Role Identification By Resume

IBM322
Analytics for Managerial Decision Making

Group 10
Indian Institute of Technology Roorkee
Haridwar, Uttarakhand, India

2 December 2024

Full Name	Enrollment No.
Sneha Maheshwari	22323037
Archi	22323008
Luvpreet Singh	22323029
Kishanpal Singh	22323026

Contents

1 Introduction

The increasing reliance on online job portals and recruitment systems has transformed the way organizations and individuals connect. With vast amounts of resume data available, automating the job role prediction process has become a crucial need. Predicting appropriate job roles based on candidates' resumes not only streamlines recruitment but also enhances efficiency for both employers and job seekers.

The objective of this project is to develop an intelligent system capable of analyzing resumes and categorizing candidates into suitable job roles. This involves processing unstructured text data, extracting meaningful insights, and utilizing machine learning algorithms to classify resumes into predefined job categories.

The project addresses the following challenges:

Handling diverse resume formats and unstructured textual data. Identifying key skills, qualifications, and experiences relevant to different roles. Ensuring scalability and accuracy in predicting job roles across various industries. We hypothesize that traditional methods of resume screening are inefficient and error-prone, and that a robust machine learning-based solution can effectively enhance the recruitment process by automating job role predictions with high precision :

2 Dataset

The project utilizes a dataset containing job-related features to predict job roles. This dataset consists of 500 rows and 12 columns. Below is a detailed description of the dataset:

- **Job Title:** The title of the job. (String)
- **Key Skills:** The key skills required for the job. (String)
- **Role Category:** The category of the job. (String)
- **Location:** The location of the job. (String)
- **Functional Area:** The functional area of the job. (String)
- **Industry:** The industry of the job. (String)
- **Longitude:** The longitude of the job. (Float)
- **Latitude:** The latitude of the job. (Float)
- **sal:** The salary of the job. (Float)

3 Preprocessing

Before initiating the classification task, we performed several essential preprocessing steps to prepare the dataset for modeling. These steps are as follows:

1. **Data Framing:** The dataset was reviewed to identify missing or inconsistent values. Critical columns such as **'Job Title'**, **'Role Category'**, **'Functional Area'**, and **'Role'** were found to contain null or undefined values.

   - Missing textual data ('**Job Title**','**Role Category'**, etc.) was replaced with placeholders such as "Unknown" or "Other" to retain the records for further processing.
   - This ensured the dataset's integrity and completeness without discarding potentially valuable data.

2. **Text Normalization and Stemming:** Textual columns, such as **'Job Title'** and **'Role'**, were stemmed to standardize the variations of words. This process reduced inflected or derived words to their base form, ensuring that similar terms (e.g., "managing" and "management") were treated uniformly. The Porter Stemmer was employed for this task, along with tokenization to handle multi-word phrases effectively.

3. **Salary Normalization:** To standardize salary data, normalization techniques were applied. This involved scaling the salary values to a consistent range using Min-Max normalization:

$$\text{Normalized Value} = \frac{\text{Value} - \text{Min Value}}{\text{Max Value} - \text{Min Value}}$$

   This step enabled comparability between salaries with varying scales and units. Additionally, the salary data was processed to extract **minimum** and **maximum** values for each record (if ranges were provided, such as "50,000–70,000"). This facilitated a consistent numeric representation of salary information.

4. **Text-to-Numeric Conversion:**To enable machine learning models to work with textual data, relevant text columns were transformed into numeric representations using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This method effectively captured the significance of terms across documents while reducing noise from commonly occurring words. The vectorization process was constrained to a maximum number of features to manage dimensionality.

5. **Final Preprocessing Outcome:** At the conclusion of preprocessing:

   - The dataset was clean, complete, and normalized.
   - Text data was stemmed and represented numerically.
   - Salary data was scaled, normalized, and decomposed into consistent features. This prepared the dataset for downstream classification tasks, including logistic regression and Gaussian Mixture Model (GMM) clustering.

4 ML Algorithms Used

The following machine learning algorithms were tested:

- **Logistic Regression:** A computationally efficient, interpretable baseline model for large datasets, offering resilience to irrelevant features and a solid starting point for understanding feature-job role relationships.

- **GMM:** An unsupervised probabilistic model that identifies clusters in resume data, aligning them with job roles. Ideal for uncovering patterns in ambiguous labels.
- **KNN:** A non-parametric, simple algorithm that classifies resumes based on proximity in feature space, effective for capturing non-linear relationships between features and job roles.
- **PCA:** An ensemble method for improved accuracy.

Algorithm	**Accuracy (%)**
Logistic Regression	59.18
KNN	61.22
GMM	81.63

Table 1: Accuracy of Machine Learning Algorithms befor PCA

Algorithm	**Accuracy (%)**
Logistic Regression	60.20
KNN	58.16

Table 2: Accuracy of Machine Learning Algorithms after PCA

Figures for the confusion matrices and model performance before PCA:

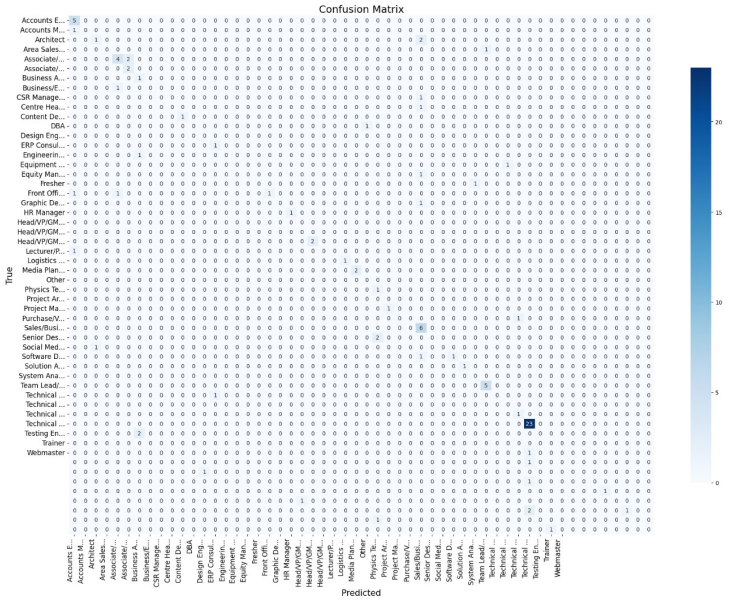

Figure 1: Confusion Matrix for KNN before PCA

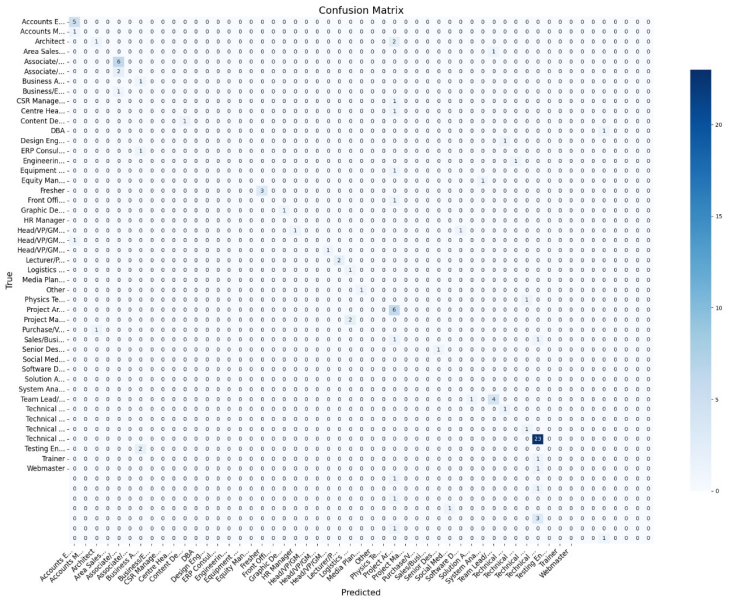

Figure 2: Confusion Matrix for Logistic Regression before PCA

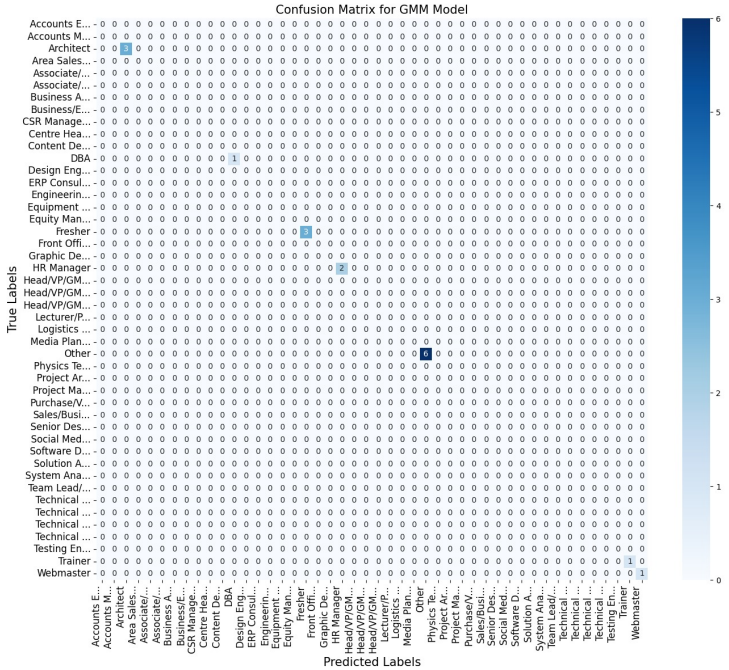

Figure 3: Confusion Matrix for GMM before PCA

Figures for the confusion matrices and model performance after PCA:

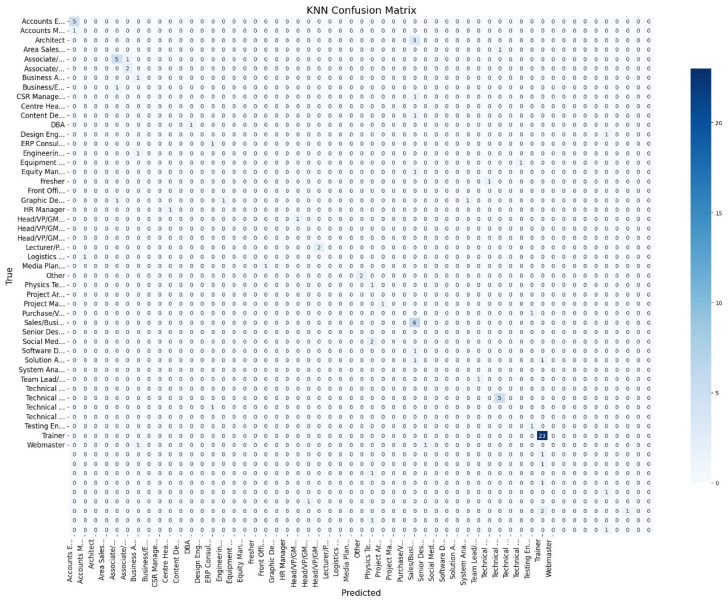

Figure 4: Confusion Matrix for KNN after PCA

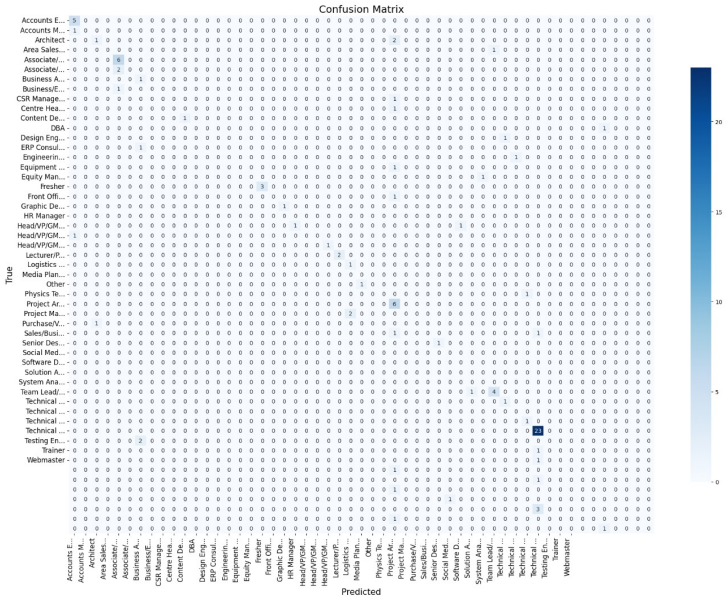

Figure 5: Confusion Matrix for Logistic Regression after PCA

5 Changes after PCA

- **Purpose:** PCA reduces data dimensionality while retaining variance by projecting data onto principal components. It helps avoid overfitting, speeds up training, and can sometimes improve generalization.
- **Challenges and Impact on Models:** Addresses sequential data dependencies.

1. **Logistic Regression After PCA:**

- **Linear Model Limitation:** PCA transforms data into a new space, which might not align with the linear assumptions of logistic regression.
- **Loss of Informative Features:** PCA may discard low-variance features that still carry valuable information for prediction, leading to performance drop.
- **Over-simplification:** PCA might oversimplify complex patterns, reducing discriminative information needed for logistic regression.

2. **KNN After PCA:**

- **Curse of Dimensionality:** In high-dimensional spaces, distances between data points become less meaningful, affecting KNN's performance.
- **Information Loss:** PCA focuses on retaining variance, not necessarily meaningful patterns for classification, reducing KNN's effectiveness.
- **Local Structure Disruption:** PCA can distort local structures in data, impairing KNN's ability to identify correct neighbors.

6 Key Observations, Conclusions, and Recommendations:

1. **GMM (81%) - Best Model:**

- Conclusion: GMM significantly outperforms other models, capturing data patterns effectively.
- Action: GMM should be the primary choice, with hyperparameter tuning to enhance performance further.

2. **Logistic Regression (59%) KNN (62%):**

- Conclusion: Both models underperform compared to GMM. Logistic regression shows minimal improvement after PCA (60%), and KNN worsens (59%).
- Action: Explore feature engineering but deprioritize these models.

3. **PCA Transformations:**

- Conclusion: PCA fails to improve performance for logistic regression and KNN, suggesting it doesn't add value in dimensionality reduction for this task.
- Action: Avoid PCA unless further analysis shows significant benefits.

4. **Final Recommendation:**

- Conclusion: GMM is the most suitable model for job title prediction, achieving the highest accuracy (81%).
- Action: Focus on optimizing GMM with feature engineering and advanced techniques for even better results.