

AI-Powered Resume Classification System

Your Name

June 2025

Problem Statement

The manual process of reviewing resumes for job applications is time-consuming and labor-intensive, especially when organizations receive hundreds or thousands of applications. This project develops an AI-powered resume classification system to automate the analysis of text-based resumes, extracting key information (e.g., skills, education, experience) and classifying them into predefined job roles such as Software Engineer, Data Analyst, or Manager. The system aims to improve efficiency and accuracy in candidate screening.

Data Representation

Resumes are processed as text data, with categorical features (e.g., degree type, skills) extracted for model training. These features are transformed into numerical formats suitable for machine learning algorithms.

One-Hot Encoding

One-hot encoding converts categorical variables into binary vectors, enabling algorithms to interpret them effectively. For instance, a candidate's degree type is encoded as follows:

- B.S. = $[1, 0, 0]$
- M.S. = $[0, 1, 0]$
- Ph.D. = $[0, 0, 1]$

Similarly, skills like Python or Java are encoded as binary vectors to represent their presence or absence in a resume.

Model Training and Classifiers

Four supervised learning classifiers were trained on preprocessed resume data, using features such as years of experience, education level, and skills.

Decision Tree Classifier

A Decision Tree Classifier builds a tree-like model by recursively splitting the dataset based on feature values (e.g., years of experience or specific skills) to predict job roles.

Random Forest Classifier

Random Forest constructs multiple decision trees, each trained on a random subset of data and features, and aggregates their predictions to enhance accuracy and reduce overfitting.

XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a gradient boosting algorithm that optimizes decision trees by minimizing a loss function and applying regularization to prevent overfitting.

KNN Classifier

The K-Nearest Neighbors (KNN) Classifier assigns a resume to a job role based on the majority class of its k nearest neighbors in the feature space.

Model Evaluation and Performance

The classifiers were evaluated on a test dataset of resumes, with accuracy scores summarized below:

Table 1: Accuracy scores of the classifiers.

Classifier	Accuracy
Decision Tree	0.50
Random Forest	0.29
XGBoost	0.39
KNN	0.33

Conclusion: The Decision Tree Classifier achieved the highest accuracy of 0.50. However, the low accuracies across all models suggest potential improvements in feature engineering (e.g., using word embeddings), dataset size, or hyperparameter tuning.

Contribution Guidelines

Contributions to improve the AI-Powered Resume Classification System are encouraged. To contribute, follow these steps:

1. Fork the repository at [GitHub](#).
2. Create a new branch (`git checkout -b feature-branch`).
3. Implement changes (e.g., enhance model accuracy, optimize preprocessing).
4. Commit changes (`git commit -am 'Add new feature'`).
5. Push to the branch (`git push origin feature-branch`).
6. Submit a Pull Request.