Resume Classification using Machine Learning Algorithm

Introduction: Resume classification using Machine Learning is a powerful and versatile tool that significantly streamlines the hiring process. By automating the evaluation of resumes, it saves valuable time for HR departments and hiring managers, allowing them to quickly identify the most qualified candidates. This technology enhances efficiency, reduces bias, and helps organizations make data-driven decisions when selecting applicants for interviews.

Machine Learning Overview: We designed this machine learning model that can automatically classify the resumes based on their content using the Machine Learning model. The model can categorize resumes based on predefined unique values such as HR, Design, Engineer and many more. The script reads resumes from a specified directory, processes them, and moves each resume to its corresponding category folder. Additionally, it generates a CSV file named categorized_resume.csv that lists each resume's filename and its assigned category.

Model: Random Forest Classifier: The model used for resume classification is a **Random Forest Classifier**. This model was selected for several reasons:

1. **Robustness:** Random Forest is an ensemble learning method that builds multiple decision trees and merges them to obtain a more accurate and stable prediction. It reduces the risk of overfitting, which is a common issue in decision trees when dealing with complex datasets like resumes.
2. **Performance:** Random Forest is known for its high performance in classification tasks, especially when the dataset contains many features, as in the case of text data represented by TF-IDF vectors.
3. **Feature Importance:** One of the advantages of Random Forest is its ability to provide insights into feature importance. This can be useful for understanding which words or phrases are most indicative of specific resume categories.
4. **Versatility:** Random Forest can handle both binary and multi-class classification tasks, making it suitable for categorizing resumes into multiple categories.

Fig 1: Relevant features of different resume data.

Model Training

The Random Forest model was trained on a dataset of resumes using the following steps:

1. **Data Preparation:** The resumes were cleaned and preprocessed to remove noise and irrelevant information.
2. **Feature Extraction:** Text data was transformed into numerical features using TF-IDF vectorization.
3. **Model Training:** The Random Forest model was trained using these features, and its performance was evaluated using metrics like accuracy, precision, recall, and F1-score.

Output:

Validation Accuracy: 64%

Accuracy: 73%

Macro Average:

- **Precision: 0.69**
- **Recall: 0.67**
- **F1-Score: 0.66**

Weighted Average:

- **Precision: 0.73**
- **Recall: 0.73**
- **F1-Score: 0.71**

Dependencies: We make a requirements.txt file and it just needs to install the file to resolve dependency.

Script Components

Based on the model we made

1. rf_classifier_model.pkl',
2. tfidf_vectorizer.pkl' and using them made script.pt.

Load model and vectorizer (model_path, vectorizer_path):

- **Purpose:** Loads the pre-trained Random Forest model and TF-IDF vectorizer from the specified file paths.
- **Parameters:**
    1. model_path: Path to the saved model file (.pkl).
    2. vectorizer_path: Path to the saved TF-IDF vectorizer file (.pkl).
- **Returns:** The loaded model and vectorizer.

Extract text from pdf(pdf_path):

- **Purpose:** Extracts text content from a PDF file.
- **Parameters:**
    1. pdf_path: Path to the PDF file.
- **Returns:** Extracted text from the PDF file as a string.

Categorize resumes (resume_directory, model, vectorizer)

- **Purpose:** Classifies resumes in the specified directory, moves them into category folders, and returns the categorization data.
- **Parameters:**
    1. resume_directory: Directory containing the resumes to be categorized.
    2. model: Pre-trained Random Forest model for classification.
    3. vectorizer: TF-IDF vectorizer used for transforming text data.
- **Returns:** A list of dictionaries containing filename and category.

To run the script we need to type the following command:

python script.py /path/to/resume_directory

Outputs

- The script creates a folder for each category within the specified resume directory and moves the respective resumes into these folders.
- A file named categorized_resume.csv is generated in the script's directory, containing two columns:
    1. filename: The name of the resume file.
    2. category: The category assigned to the resume.

1	filename	category
2	10030015.	15
3	10186968.	7
4	10219099.	15
5	10554236.	0
6	10624813.	15
7	10712803.	15

Fig 2: Output as CSV format.

Except this, the folder creates some subfolder names as unique values. And inside the folders there the resumes are categorized.

The model can be further enhanced by incorporating advanced techniques such as the T5 Transformer or BERT Transformer for training and evaluation. These models, known for their state-of-the-art performance in natural language processing tasks, could potentially improve the accuracy and robustness of the resume classification process. However, due to time constraints and limited access to necessary resources, implementing these advanced models is not feasible at the moment. Despite this, the current script still offers a highly effective and reliable solution, leveraging proven machine learning methods to deliver robust performance comparable to other tools available in the market.