**Resume Classification using NLP and Machine Learning**

**Overview**

This project focuses on automating the classification of resumes based on various job categories using Natural Language Processing (NLP) techniques and Machine Learning models. The system takes resumes as input, processes the textual data using various NLP techniques, and then classifies them into predefined categories. The goal is to create a robust pipeline for resume parsing and classification to streamline recruitment processes.

**Project Workflow**

1. **Data Collection**:
   * Resumes are collected in different formats (PDF, DOC, DOCX).
   * Each resume is categorized into different job profiles such as Data Scientist, Web Developer, Software Engineer, etc.
2. **Data Preprocessing**:
   * Extract textual data from resumes using libraries like docx2txt and PyPDF2.
   * Clean and preprocess the text, including:
     + Lowercasing
     + Removing stop words
     + Tokenization
     + Lemmatization/Stemming
     + Removing numeric values and URLs
   * Generate additional features such as word count, stopword count, and numeric word count.
3. **Exploratory Data Analysis (EDA)**:
   * Visualize the frequency distribution of profiles using bar charts and pie charts.
   * Analyze the most frequent words and n-grams (unigrams and bigrams) across resumes.
   * Generate word clouds for key insights.
4. **Feature Engineering**:
   * Use TF-IDF vectorizer for text vectorization, which converts raw text into meaningful numeric features.
   * Extract key features like word frequency and n-gram counts to build a comprehensive feature set.
5. **Model Training**:
   * Train multiple Machine Learning models for classification, including:
     + K-Nearest Neighbors (KNN)
     + Decision Tree
     + Random Forest
     + Support Vector Machines (SVM)
     + Logistic Regression
     + Bagging Classifier
     + AdaBoost
     + Gradient Boosting
     + Naive Bayes
   * Evaluate the performance of models based on precision, recall, F1-score, and accuracy.
6. **Model Evaluation**:
   * Compare test and training accuracy across models.
   * Analyze classification performance using confusion matrices and classification reports.
7. **Results and Visualizations**:
   * Display top features and important n-grams using bar plots.
   * Visualize classification results through pie charts and bar graphs.
   * Present model performance comparisons to select the best algorithm for resume classification.

**Technologies Used**

* **NLP Libraries**: NLTK, SpaCy, docx2txt, PyPDF2
* **Machine Learning Libraries**: Scikit-learn, XGBoost, AdaBoost
* **Data Handling**: Pandas, Numpy
* **Data Visualization**: Matplotlib, Seaborn
* **Text Vectorization**: TfidfVectorizer, CountVectorizer
* **Other Tools**: WordCloud for visualization