## **Proposal: AI-Based Job Application Success Predictor and Recommendation Engine**

**Project Contributors:** Oren Moreno, Stanley Huynh, Andrew Kuruvilla

\*Notes from professor:

1. Datasets may be difficult to source
2. NLP is resource intensive so test with small subsets of data
3. Test many different models to compare performance
4. Make sure evaluation metrics are solid (ROC/AUC and confidence matrix suggested).

### **Executive Summary**

This proposal outlines the development of an AI-based job application success predictor and job recommendation engine. The tool will leverage pre-trained machine learning algorithms to analyze various data points, including candidate resumes, cover letters, job descriptions, and historical hiring data. By identifying patterns and correlations, the predictor aims to provide accurate predictions on the likelihood of a candidate's success in a specific job role while the recommendation engine seeks to recommend the most suitable positions for a candidate. This innovative solution can significantly streamline the hiring process, improve candidate selection accuracy, and enhance overall organizational efficiency while also providing insights into other roles for which a candidate may be suitable.

### **Problem Statement**

The traditional job application process often involves manual screening and evaluation, which can be time-consuming, subjective, and prone to bias. This leads to inefficient hiring practices, increased costs, and potentially missed opportunities to hire top talent. An AI-based job application success predictor can address these challenges by automating the initial screening process and providing data-driven insights to hiring managers.

### **Proposed Solution**

The predictor will be developed using advanced machine learning techniques, such as natural language processing (NLP) and statistical modeling. Key features of the solution include:

* **Resume and cover letter analysis:** Extraction of relevant keywords, skills, and experiences from candidate documents.
* **Job description analysis:** Identification of essential qualifications, skills, and experiences required for the role.
* **Historical data analysis:** Examination of past hiring decisions and candidate outcomes to identify patterns and correlations.
* **Machine learning model development:** Training of a predictive model using a diverse dataset of resumes, cover letters, job descriptions, and hiring outcomes.
* **Prediction generation:** Scoring of candidate applications based on their predicted likelihood of success in the role.
* **Recommendations generation:** Recommends other job titles that are likely to match the candidate’s abilities based on the descriptions of those jobs.

### **Benefits**

* **Improved hiring efficiency:** Automation of the initial screening process, reducing the time and resources required to review applications.
* **Enhanced candidate selection accuracy:** Identification of candidates who are most likely to be successful in the role, leading to better hiring decisions.
* **Reduced bias:** Objective evaluation of candidates based on data-driven insights, minimizing the impact of personal biases.
* **Data-driven decision-making:** Provision of actionable insights to hiring managers, enabling informed decisions about candidate selection.
* **Insights for Candidates:** Job-seekers can also benefit from this tool by seeing how well they may fit a particular job and can get ideas for other jobs that they may be suitable for.

### **Implementation Overview**

1. **Data collection and preparation:** Gathering and cleaning a diverse dataset of resumes, cover letters, job descriptions, and hiring outcomes.
2. **Feature engineering:** Extraction of relevant features from the data, such as keywords, skills, and experience levels.
3. **Model development:** Training a machine learning model using appropriate algorithms, such as random forest or gradient boosting.
4. **Model evaluation:** Assessing the performance of the model using metrics like accuracy, precision, recall, and F1-score.
5. **Deployment:** Integration of the predictor into the organization's hiring process.

### **Budget and Timeline**

*timeline of completion dates here*:

Proposal Submitted - 9/16/2024

Final Presentation - 12/08/2024

### **Conclusion**

The development of an AI-based job application success predictor and recommendation engine represents a significant opportunity to revolutionize the hiring process. By leveraging advanced machine learning techniques, this tool can provide valuable insights to hiring managers, improve candidate selection accuracy, and enhance overall organizational efficiency.

### **Project Implementation Plan**

The **bold** names in parentheses at the end of each lettered part indicate the person responsible for leading that particular part, though all members may contribute to each part of each numbered step.

1. Data Preparation: This step is crucial for creating a clean, structured dataset for model training.

a) Data Collection: Gather resumes and job descriptions from available sources. Why: Provides the raw material for the project. **(Andrew)**

b) Text Cleaning: Remove irrelevant information and standardize formatting. Why: Ensures consistency and improves data quality for better model performance. **(Stanley)**

c) Information Extraction: Extract key details like skills, experience, and job titles. Why: Transforms unstructured text into structured data for feature engineering. **(Oren)**

1. Model Selection and System Implementation: This step creates meaningful inputs for the machine learning models and d.

a) Model Choice and Data Splitting: Choose an open-source, pre-trained model such as BERT or Llama and create a method for splitting the datasets into training, validation, and test sets. Why: The pre-trained model will allow semantic meanings within the datasets to be understood and splitting is necessary before training. **(Andrew)**

b) Create Training Loops: Defines the hyperparameters for batch processing of the datasets to fine-tune the pre-trained model. Why: This step sets the controls for model fine-tuning. **(Oren)**

c) Fine-tuning: Fine-tune the model using the datasets of resumes and job descriptions. Why: Increases the ability of the model to recognize the type of data found in the datasets. **(Stanley)**

3. Matching System Implementation Rationale: This step creates the core functionality of the project.

a) Similarity Scoring: Develop a method to measure the relevance between a resume and a job description. Why: Provides the basis for matching and recommendations. **(Andrew)**

b) Job Recommendation: Implement a system to suggest relevant jobs based on a given resume. Why: Fulfills the project's objective of recommending jobs. **(Oren)**

c) Performance Evaluation Metrics: Define metrics to assess the quality of matches and recommendations. Why: Enables demonstration of the effectiveness of the implemented system. **(Stanley)**

4. Model Evaluation and Optimization: Assessing and attempting to improve the performance of the fine-tuned model.

a) Performance Evaluation: Calculate relevant metrics and examine and categorize model mistakes. Why: Demonstrates implementation of evaluation metrics and ability to critically assess model performance. **(Stanley)**

b) Hyperparameter Tuning: Optimize model parameters using techniques like grid search or Bayesian optimization. Why: Shows understanding of model optimization techniques and possibly improves model performance. **(Andrew)**

c) Performance Visualization: Create charts and graphs to illustrate model performance. Why: Shows data visualization skills and ability to communicate results effectively. **(Oren)**

5. Documentation and Presentation Rationale: This step demonstrates ability to communicate technical work clearly.

a) Code Documentation: Write clear comments and function docstrings. Why: Shows ability to create maintainable, understandable code. **(Oren)**

b) README Creation: Write a comprehensive README explaining the project, its setup, and usage. Why: Demonstrates ability to document a project for other developers or users. **(Andrew)**

c) Results Summary: Create a concise report or presentation of the project's outcomes. Why: Shows ability to summarize technical work for a potentially non-technical audience. **(Stanley)**