**Job Recommendations Optimization and Candidate Interest Prediction Report.**

|  |
| --- |
| **Project Members** |
| 1.    Isaac Munyaka. |
| 2.    Phemina Wambui. |
| 3.    Otiende Ogada. |
| 4.    Caroline Gesaka. |
| 5.    Ann Njoroge. |
| 6.    Joan Maina. |

**Submission Date**: 10/08/2024

**GithubLink**:

Contents

[Overview 3](#_Toc174022752)

[Introduction 3](#_Toc174022753)

[Business Understanding 3](#_Toc174022754)

[Problem Statement 3](#_Toc174022755)

[Objectives 4](#_Toc174022756)

[Success Metrics 4](#_Toc174022757)

[Data Understanding 5](#_Toc174022758)

[Data Inspection 6](#_Toc174022759)

[Data Cleaning 7](#_Toc174022760)

[Exploratory Data Analysis (EDA) 8](#_Toc174022761)

[Modeling 11](#_Toc174022762)

[Feature Selection 13](#_Toc174022763)

[Job Recommendation Model 14](#_Toc174022764)

[Filtering and Recommending Top Jobs 14](#_Toc174022765)

[Conclusion 15](#_Toc174022766)

[Recommendation 15](#_Toc174022767)

### Overview

The employment sector is continuously evolving, with job seekers and employers facing numerous challenges. Job seekers often struggle to find suitable job opportunities that match their skills and preferences, while employers find it difficult to attract qualified candidates for their job postings. With the advent of advanced data analytics and machine learning, there is an opportunity to enhance the job matching process, making it more efficient and effective for both parties.

### Introduction

In today's competitive job market, the ability to efficiently match job seekers with relevant job opportunities is crucial. Leveraging data from job postings and job seeker profiles, machine learning models can significantly improve the job search experience and the quality of job applications received by employers. This proposal outlines a plan to develop two machine learning models: one for optimizing job recommendations for job seekers and another for predicting the likelihood of job postings receiving a high or low number of applications.

### Business Understanding

For job seekers, finding the right job that matches their skills, preferences, and career aspirations is a challenging task. Similarly, employers face difficulties in creating job postings that attract the right candidates. By addressing these challenges through advanced machine learning models, we can create a more efficient job market, benefiting both job seekers and employers.

### Problem Statement

Despite the vast amount of job postings available online, job seekers often find it challenging to identify the most relevant opportunities. Conversely, employers struggle to understand what factors contribute to the attractiveness of their job postings, leading to a mismatch between job offers and applications. This proposal aims to solve these issues by:

1. Providing personalized job recommendations to job seekers to match them with the most suitable job postings.
2. Developing a machine learning model that predicts whether a job posting will receive a high or low number of applications, helping employers improve their job postings.

**Research Questions**

The following are the questions the project intended to answer:

How can job seekers' preferences and qualifications be effectively matched with available job postings to provide personalized recommendations?

Which factors most significantly influence the number of applications a job posting receives?

What specific improvements can employers make to their job postings to increase their attractiveness based on model predictions?

### Objectives

1. **Optimize Job Recommendations**: Provide personalized job recommendations to job seekers to match them with the most suitable job postings.
2. **Predict Candidate Interest**: Develop a machine learning model to predict whether a job posting will receive a high or low number of applications, enabling companies to understand which factors attract candidates the most and improve their job postings.

### Success Metrics

**Application Rate**: Percentage of job postings receiving applications**.**

**Qualified Application Rate**: Percentage of applications meeting job requirements.

**Precision**: Measure the proportion of recommended jobs that are relevant to the job seeker's input skills.

**Recall**: Measure the proportion of relevant jobs that are successfully recommended to the job seeker.

**F1 Score**: Balance precision and recall to provide a single metric that evaluates the model's performance.

**AUC (Area Under the Curve)**: Measure the model's ability to distinguish between high and low application likelihoods.

**User Satisfaction**: Evaluate the job matching system based on user feedback and satisfaction with the recommendations provided.

### Data Understanding

**Loaded the dataset**: Loaded the job postings and application data into a DataFrame. The dataset has **123849** rows and **28** columns. The columns are as follows;  
Company name: Provides the name of the company offering the job, which helps in understanding the job context and employer branding.

**Title:** The job title is critical for identifying the nature of the job and matching it with user preferences.

**Description:** Contains detailed information about job responsibilities and requirements, which is crucial for matching the job with user skills and interests.

**Max salary**: Indicates the highest salary offered for the job, helping to match job seekers' salary expectations.

**Min salary**: Shows the lowest salary offered, providing a range for matching job seekers' financial requirements.

**Location**: Geographic location of the job, which is important for matching based on job seekers' preferred or available locations.

**Views:** Number of views the job posting has received, which can indicate the popularity or competitiveness of the job.

**Med salary:** Median salary for the job, providing a central measure of compensation, useful for understanding typical earnings.

**Applies:** Number of applications received, which can reflect job demand and help gauge the job's attractiveness.

**Remote allowed:** Indicates if remote work is an option, which is increasingly relevant for job seekers preferring or needing remote work arrangements.

**Formatted experience level**: Specifies the required experience level (e.g., entry-level, senior), aiding in matching jobs with job seekers' experience.

**Skills desc:** Describes the skills required for the job, crucial for aligning job seekers' skills with job requirements.

**Listed time:** Timestamp of when the job was posted, helping to understand the job’s recency and relevance.

**Posting domain:** Indicates the industry or sector of the job, useful for matching jobs with job seekers’ industry interests.

**Currency:** Specifies the currency in which the salary is offered, important for job seekers in different regions or countries.

Compensation type: Details the type of compensation (e.g., base salary, bonuses), helping job seekers understand the total compensation package.

**Job Id:** Unique identifier for each job posting, essential for tracking and referencing individual jobs

### Data Inspection

**Checked data anomalies:** Examined the dataset for any missing values, duplicates and assessed their potential impact on the analysis.

**Handled missing values**: Addressed any missing values through imputation or removal, depending on the context and significance of the missing data.

### Data Cleaning

**Data Cleaning**

Firstly, the unique values in all the columns are identified, then finding duplicates (there are none in this case), checking for potential placeholders (there are 0 occurrences) and finding missing values.

The following columns are dropped with reasons below:

med\_salary         - 94.93% missing; does not provide additional insights from max/min salary

company\_id         - Redundant if company\_name is sufficient

closed\_time       -99.13% missing; not critical to the core objectives

application\_url     -Not integrating a direct application feature

job\_posting\_url    - Not integrating a direct application feature

posting\_domain     -not essential for our recommender logic

original\_listed\_time - since we have listed time ,its redundant

sponsored           -not essential for our recommender logic

formatted\_work\_type -reduntant since we have formatted work type

compensation\_type -reduntant since we have pay period column,pay period column provides more sufficient info than this

The following columns were kept due to the following reasons:

description: Provides job details, essential for matching job seekers with jobs.

title: Job title, crucial for relevance.

work\_type: Specifies full-time, part-time, etc.

listed\_time: Important for tracking job posting timelines.

location: Job location, crucial for geographical matching.

job\_id: Unique identifier for each job posting.

company\_name, company\_id: Information about the company.

max\_salary,min\_salary: Information on job salary range.

views

expiry              #not essential for our recommender logic

pay\_period

**Additional data cleaning included:**

Converting 'listed\_time' and 'expiry' to datetime

Updating 'listed\_time' with random dates within the range of the current date

Dropping the 'calculated\_expiry\_date' column if it's no longer needed

After the above cleaning steps, there are 0 missing values.

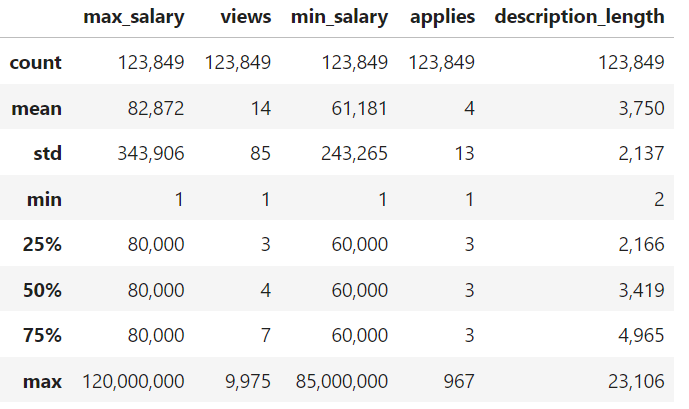
**Data type conversion**: Ensured that all columns had appropriate data types (e.g., converting categorical variables to category type).

### Exploratory Data Analysis (EDA)

The report followed the following key steps for its EDA

Descriptive Statistics

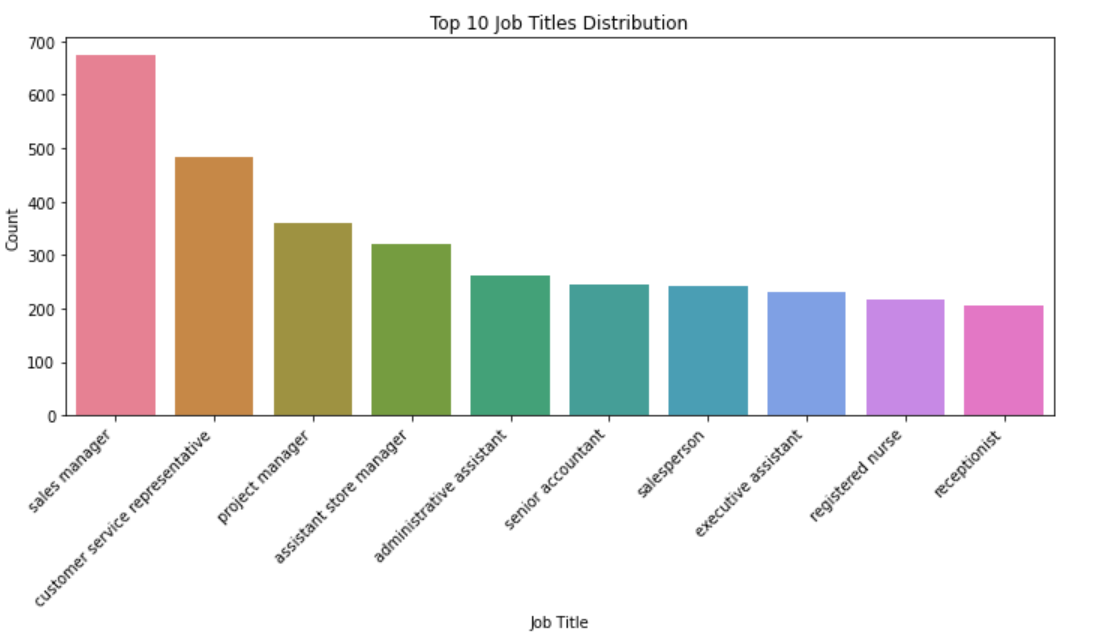
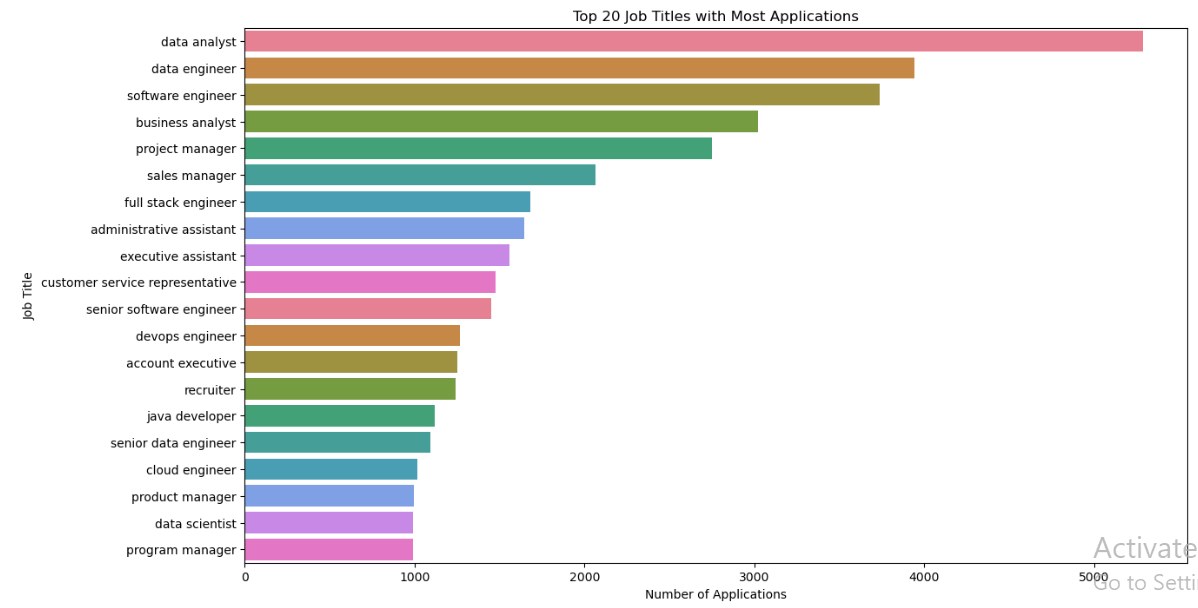
**Summary statistics**: Generated summary statistics (mean, median, standard deviation, etc.) for numerical columns to understand their distribution.



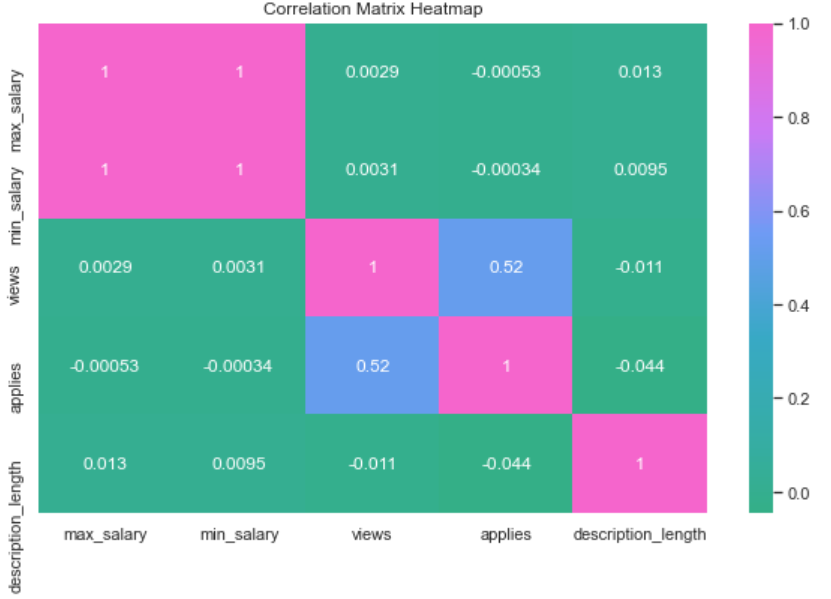
**Frequency distribution**: Examined the frequency distribution of categorical variables.

Data Visualization

Performed various univariant, bivariant and multivariant analysis to establish various relationships. Below are a few higlights.

**Correlation matrix**: Generated a correlation matrix to identify potential relationships between features.



##### Feature Engineering

**Created new features**: Derived new features from existing data to enhance model performance. For instance:

description\_length: Calculated the length of the job description.

days\_since\_listed: Calculated the number of days since the job was listed.

average\_salary: Estimated average salary based on the salary range provided.

formatted\_experience\_level: Transformed experience level into a numerical format if it was originally categorical.

**Data Preprocessing.**Encoded Categorical columns including formatted experience level, work type and currency.

Standardized the numerical columns such as views, applies, description length and average salary.

For the recommender system, tokeniced text columns using punkit, stopword and wordnet specifically the company name, title ,location and description. This was to aid Natural Language Processing.

Finally, reduced dimensionality reduction through Principal Component Analysis (PCA). Tranforming the dataframe into 5 components.

**Target Variable Analysis**

Defined the target variable by through the mean of views column to create application likelihood column. Based on the mean, a threshold was establisted to eith clasfficy an application as either high (1) or low (0).The variables were defined as below**.**

Dependent variable (y) = application likelihood

Independent variable (X) = views, description\_length, average\_salary, formatted\_experience\_level, days\_since\_listed, work\_type.

Class distribution: Checked the distribution of the target variable (high applications) to understand class imbalance.

**Handling Class Imbalance**

Applied SMOTE: Used the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance in the target variable.

**Splitting the Data**

Train-test split: Split the dataset into training and testing sets to evaluate model performance.

These steps provided a thorough understanding of the dataset, helped identify potential issues, and facilitated the creation of meaningful features for the machine learning models. The insights gained during EDA were crucial for informing the modeling process and ensuring robust model development.

Modeling   
**Selected multiple models**: Chose various machine learning models to evaluate their performance, including:

Logistic Regression – Baseline Model

Random Forest

K-Nearest Neighbors (KNN)

XGBoost

##### Model Training and Evaluation

**Cross-validation**: Performed cross-validation to assess the models' performance on the training data.Calculated cross-validation accuracy for each model.

**Model fitting**: Trained each model on the entire training set.

**Prediction and probability estimation**: Predicted the target variable on the test set and, where applicable, estimated probabilities.

Performance Metrics Calculation

**Classification report**: Generated classification reports for each model to evaluate precision, recall, and F1 score.

**ROC-AUC score**: Calculated the ROC-AUC score to assess the models' ability to distinguish between high and low application likelihoods.

Hyperparameter Tuning

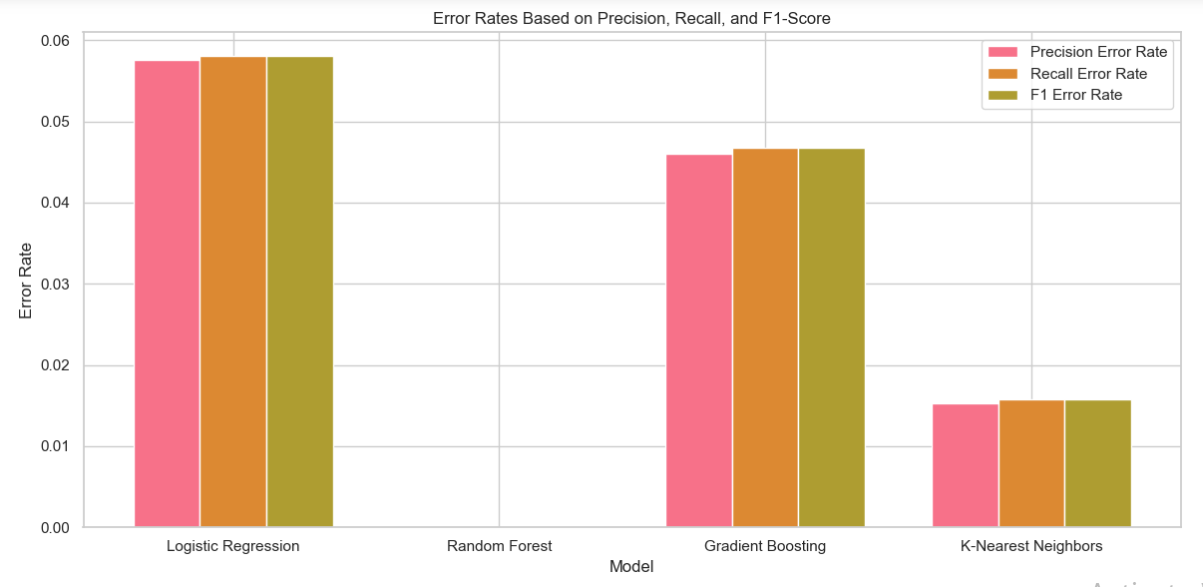
**Grid search**: Used grid search for hyperparameter tuning to find the best parameters for the models. Reported the best parameters and the corresponding F1 score for the tuned models.

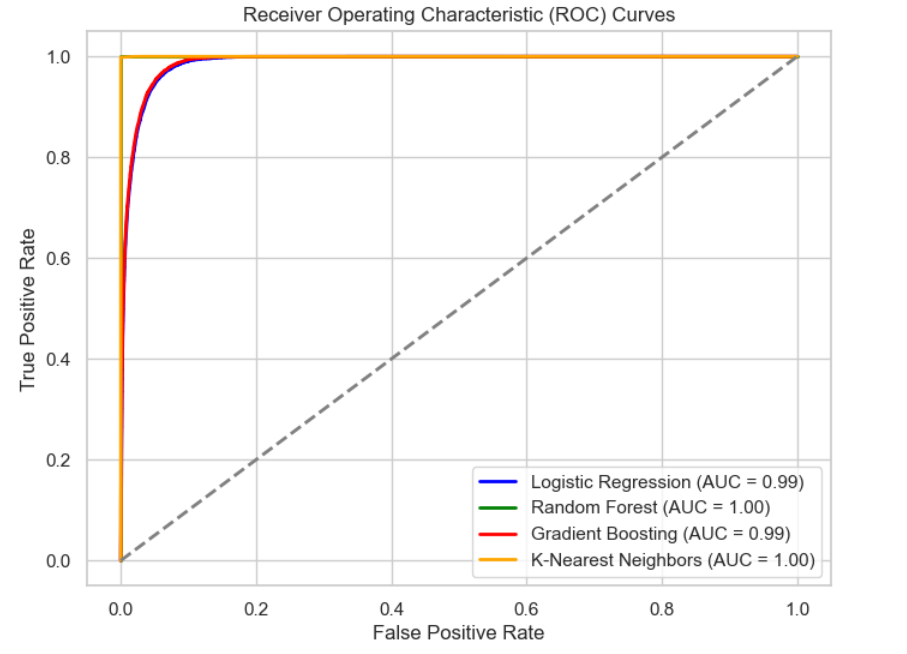
##### Model evaluation and Findings:

**Metrics collection**: Collected key performance metrics (precision, recall, F1 score, ROC-AUC score) for each model. Random Forest Model was the best performing model considered with below metrics.



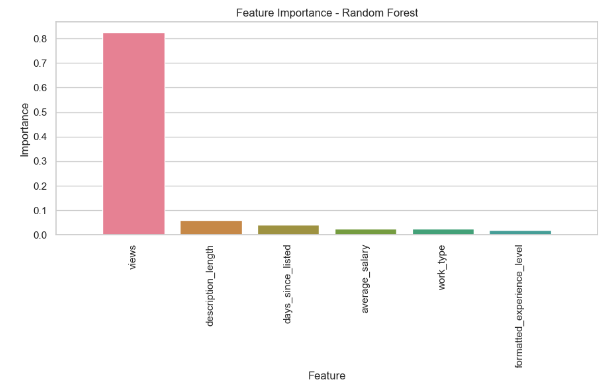
##### Model Performance Comparison

**Bar plot**: Visualized the performance metrics using a bar plots and ROC-AUC curves as belows.



### Feature Selection

Following features were selected as visualized below in their order of importance:



### Job Recommendation Model

Preprocessed the relevant text columns through **TF-IDF Vectorizer initializing** and transformed job descriptions.

**Text Preprocessing**: Improved text preprocessing by converting to lowercase, removing non-word characters, and extra whitespace.

**Job Recommendation Function**: Calculated cosine similarity between input description and job descriptions, returned top recommendations.

**Main Function**: User interface to enter job description and receive job recommendations.

##### K-Nearest Neighbors for Job Recommendations

**Feature Matrix**: Prepared using views, applies, and average\_salary.

**KNN Application**: Implemented KNN for job recommendations based on feature matrix.

**Average Distance Calculation**: Calculated average distance to nearest neighbors.

**Job Recommendations Display**: Displayed KNN recommendations for a specific job ID.

### Filtering and Recommending Top Jobs

**User Input for Top N Jobs**: Prompted user for the number of top jobs to recommend.

**Job Title Filtering**: Filtered DataFrame by job title and retrieved top N jobs based on views.

**TF-IDF Vectorizer** for Job Titles

**TF-IDF Vectorizer**: Initialized and transformed job titles.

**Text Preprocessing**: Improved text preprocessing for job titles.

**Job Recommendation Function**: Calculated cosine similarity between input title and job titles, returned top recommendations.

**Main Function**: User interface to enter job title and receive job recommendations.

**Deployment:**

Random Forest Model was deployed as the primary model due to its flawless performance on the test set.  
Job Recommender Systems was also deployed with input functions to enable users input a job description, title, job id with relevant jobs predictions set as outputs.

The deployment link was:

### Conclusions

The models developed demonstrated strong capabilities in predicting job application likelihood and providing relevant job recommendations, significantly enhancing the job matching process for both job seekers and employers.

### Recommendation

The deployed models were observed to working effectively with recommendations for continuous tuning, regular monitoring, additional features additions, enhancing data preprocessing on for the NLP embedding (like BERT)  and continuous user-interactions for feedback to improve the system.