**Which Resume Attributes Drive Job Callbacks?**

*Kejing Yan*
*Brown University*
https://github.com/anyfruit/Key-Resume-Attributes-Impacting-Job-Callbacks

## Introduction

Getting a job callback is a crucial step for candidates who want to find a job, and resumes often serve as a candidate's first impression. While resumes present various qualifications, skills, and experiences to the employers, not all attributes are valued equally. This project analyzes which resume features have more impact on callback likelihood, helping job seekers present their qualifications more effectively.

The dataset, *Which Resume Attributes Drive Job Callbacks*, includes variables such as education, work experience, skills, job requirements, and whether candidates received a callback (1) or not (0). This allows us to examine how resume content and external factors influence hiring decisions. Previous research, like Bertrand and Mullainathan's (2004) study, showed that implicit biases can also play a role, such as names influencing callback rates regardless of identical qualifications. [1]

This project aims to identify the most impactful resume attributes, uncover unexpected patterns, and offer insights to improve resumes and promote fairer hiring.

## Exploratory Data Analysis

The dataset consists of job application records with various features describing candidate and job-related attributes, such as years of experience, education level, skills, and employment gaps. This dataset provides an opportunity to analyze the factors influencing callback rates and build predictive models to identify the most impactful features.

The distribution of target variables, whether the candidate got job callback, is unbalanced. Majority candidate couldn't receive a job callback which requires additional consideration for the following modeling and analysis.

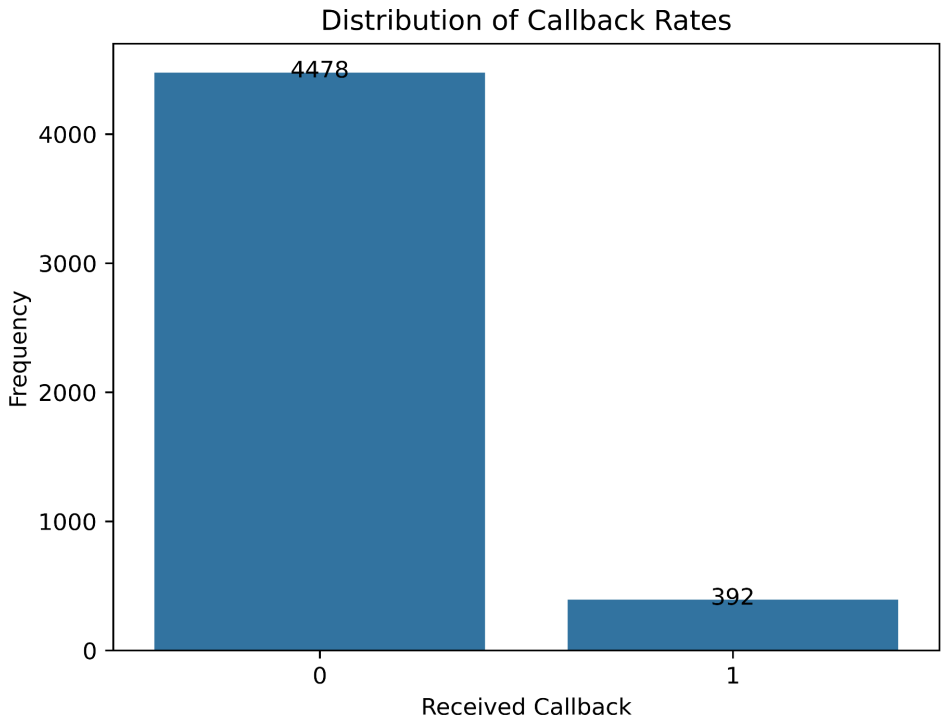


Figure 1**:** Distribution of callback rates showing the imbalance between candidates who did not receive a callback (*0*) and those who did (*1*), with a significantly higher frequency of no callbacks.

The below heatmap shows callback rates for different *job types* in *Boston* and *Chicago*. *Boston* has higher callback rates for *clerical*, *secretary*, and *supervisor* roles, while *sales rep* in *Chicago* also stand out. Other roles, especially in *Chicago*, have lower callback rates.

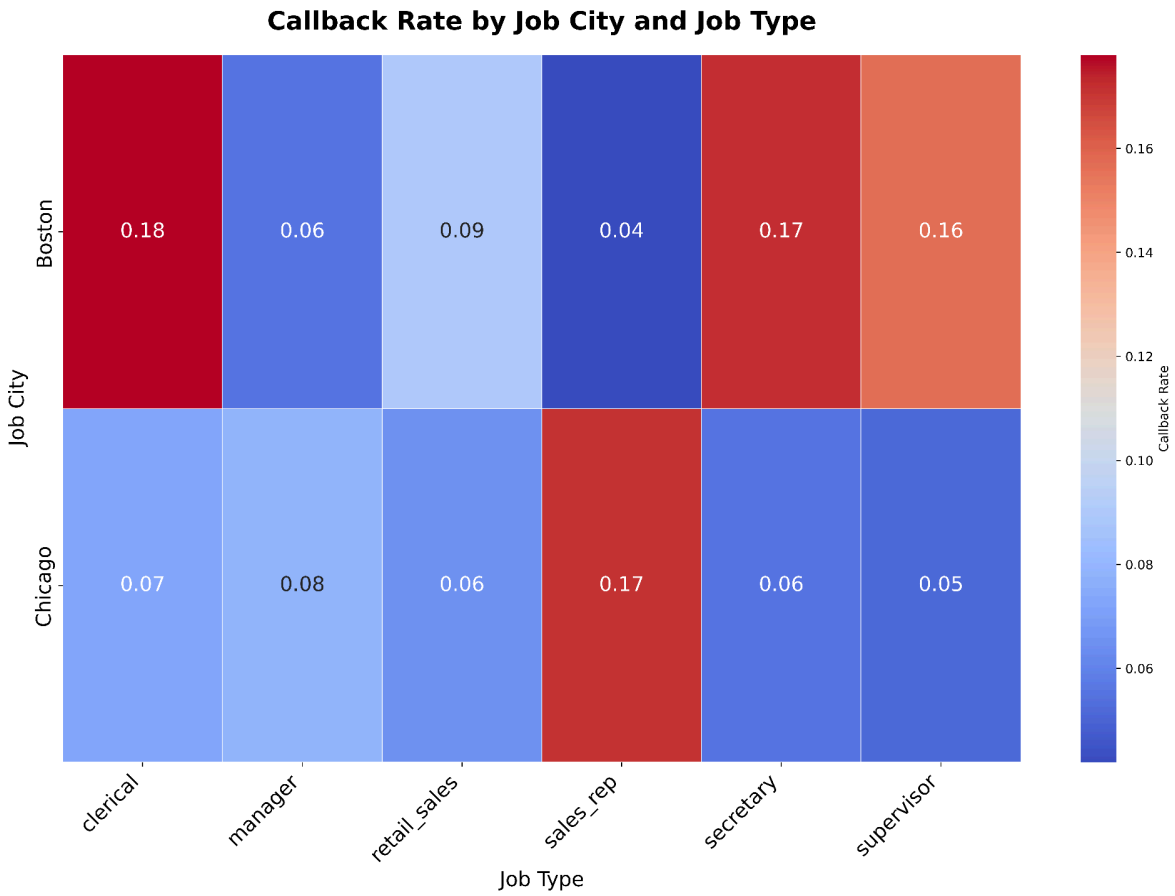


Figure 2: Heatmap showing callback rates by job city and job type, highlighting variations in callback likelihood across locations and positions.

This bar chart displays the callback rates based on the number of years of college education. Candidates with *two years of college* received the highest callback rate, followed by those with *three* and *four years*. Individuals with *zero years* of college had the lowest callback rate. This suggests that partial college education may be perceived more favorably than no college experience but does not show significant improvement after two years.

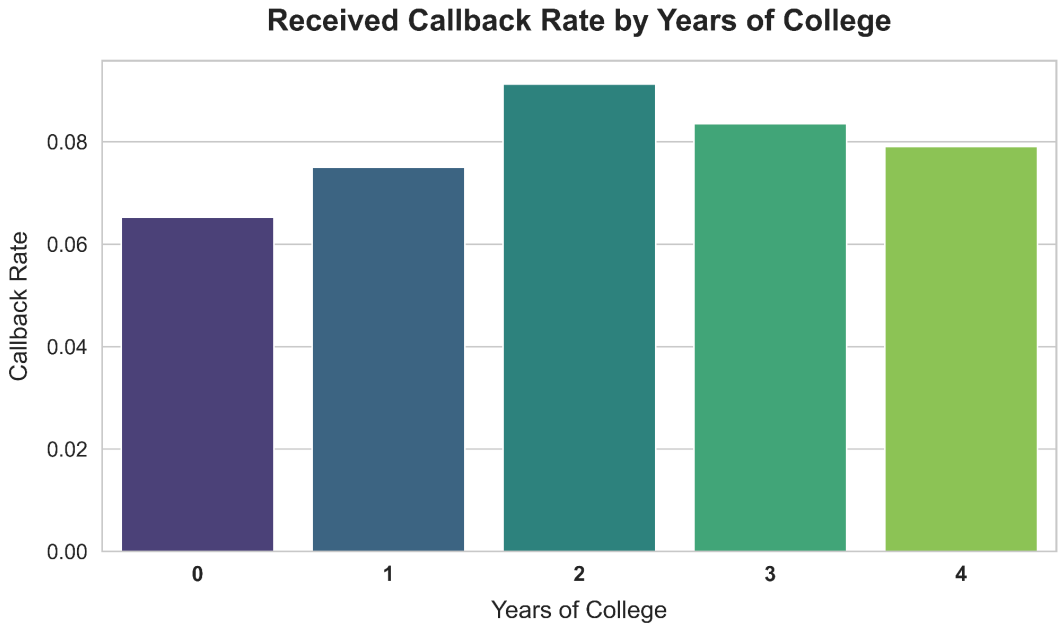


Figure 3: Callback rates for job candidates by years of college education.

Missing values are presented in three independent variables, each containing 56.4%, 40.9%, and 36.3% null values.

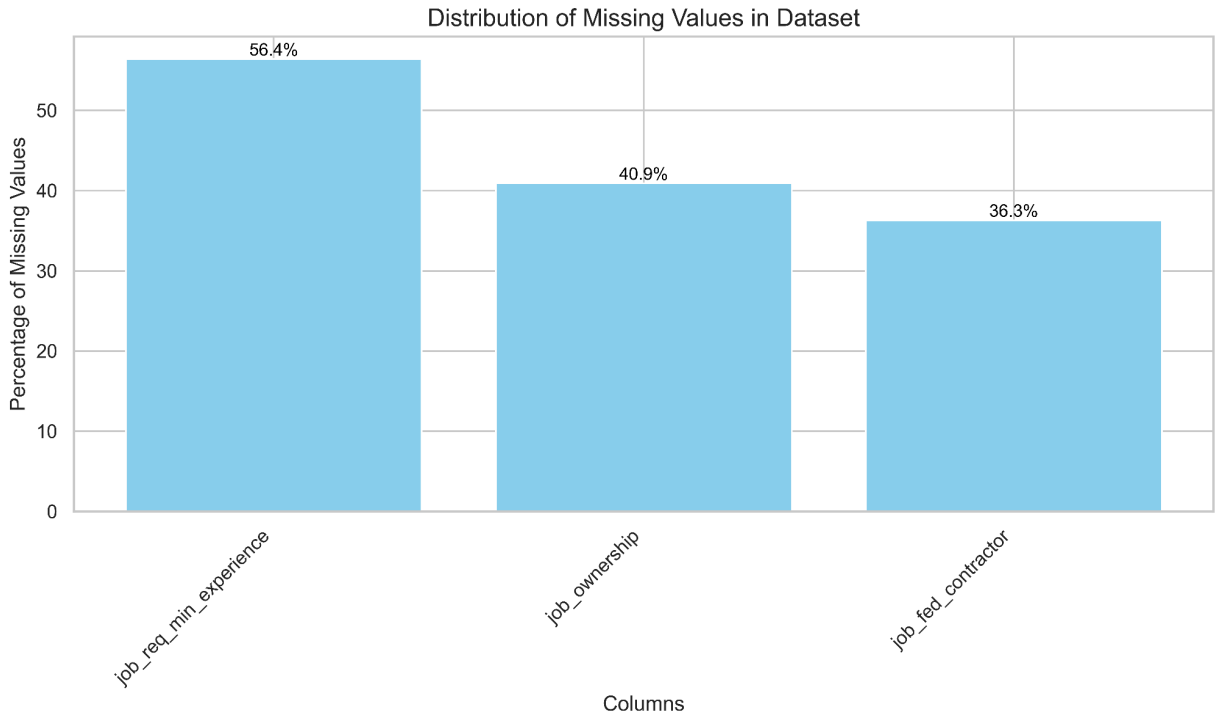


Figure 4: Distribution of missing values across features.

## Methods

As the dataset is unbalanced (91.95% class 0, 8.05% class 1), a stratified splitting strategy is applied throughout the entire splitting process. The initial act used here is to use a stratified train test split to split 20% of the original dataset to a test set. This is the first step we take to avoid data and information leakage and hence affect the evaluation and performance of the model. After this, the remaining 80% of the original data went through a stratified 4-fold KFold split to further split into 60% for training and 20% for validation.

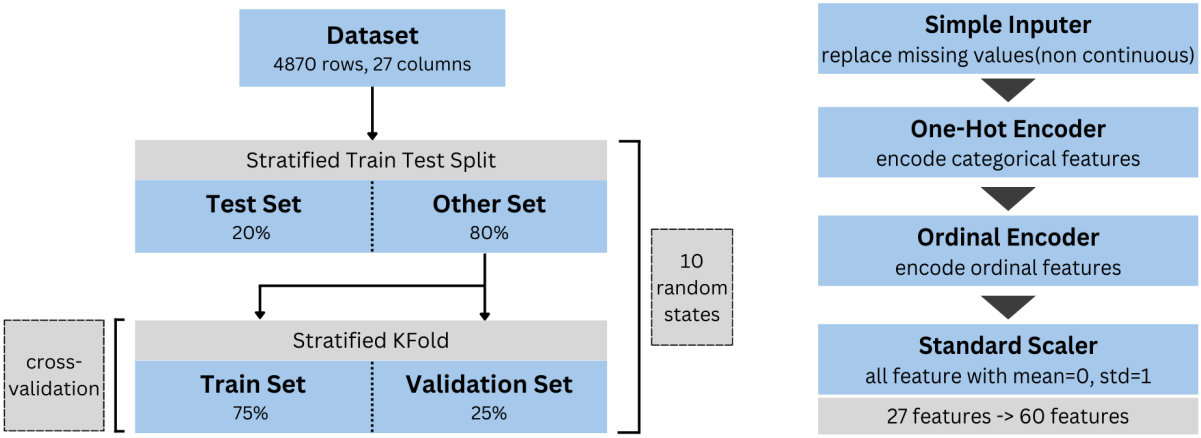


Figure 5**:** Data preprocessing and validation pipeline, including train-test split, stratified KFold cross-validation, and feature transformations like imputation, encoding, and scaling.

For missing data, this dataset only has missing values for categorical and ordinal features, so a simple imputation of 'unknown' to the missing values handled this issue. A pipeline is created using one-hot encoder for categorical features and ordinal encoder for ordinal features, and a standard scaler is applied to shift all features with the same mean and standard deviation. During the feature preprocessing process, 27 features were transformed into 60 features.

Logistic regression, Support Vector classifier, Random Forest classifier, and XGBoost classifier were applied with hyper parameters chosen as shown in the below Table 1. The hyper parameters were tuned in cross-validation to enhance accuracy and avoid bias. The bolded numbers represent the best hyperparameters; if two numbers are bolded for a single parameter, that means they share a similar frequency in 10 random states.

Table 1. Hyperparameter Grid for Model Tuning Across Different Classifiers

| **Models** | **Hyper Parameters** |
|---|---|
| Logistic Regression | C: [**0.001**, **0.01**, 0.1, 1, 10] |
| | class_weight: [**'balanced'**, None] |
| Support Vector Classifier (SVC) | C: [**0.01**, 0.1, 1, 10] |
| | class_weight: [**'balanced'**, None] |
| Random Forest Classifier | max_depth: [1, **5**, 10, 30, 50] |
| | max_features: [**'sqrt'**, **'log2'**, 0.5] |
| | max_samples: [0.5, **0.75**, 1.0] |
| | class_weight: [None, 'balanced'] |
| XGBoost Classifier | learning_rate: [**0.01**, 0.05, 0.1] |
| | max_depth: [**3**, **5**, 10] |
| | subsample: [**0.75**, 0.9, 1] |
| | scale_pos_weight: [scale_pos_weight * 0.5, **scale_pos_weight**, scale_pos_weight * 1.5][1] |

Since the dataset only contains 5k observations, iteration through 10 random states was implemented to avoid uncertainty and possible biases. The evaluation metric I choose to use is the F1 score. As the dataset is considered an imbalance, metrics such as accuracy should be avoided. F1 scores focus on a balance of precision and recall and thus avoiding over-prediction of negative class.

[1] scale_pos_weight = n_negative / n_positive

## Results

The baseline F1 score of the test set is 0.1490. The baseline F1 score is calculated by assuming all predictions are class 1 (as the majority class is class 0 and predicting it will cause precision being undefined), and computed the corresponding precision, recall, and thus F1 score. Among the 4 implemented models, XGBoost derived the highest test F1 score, possibly indicating that it has the best performance. In addition, the small standard deviation of the test score of 10 random states further revealed the consistency and reliability of the XGBoost model.

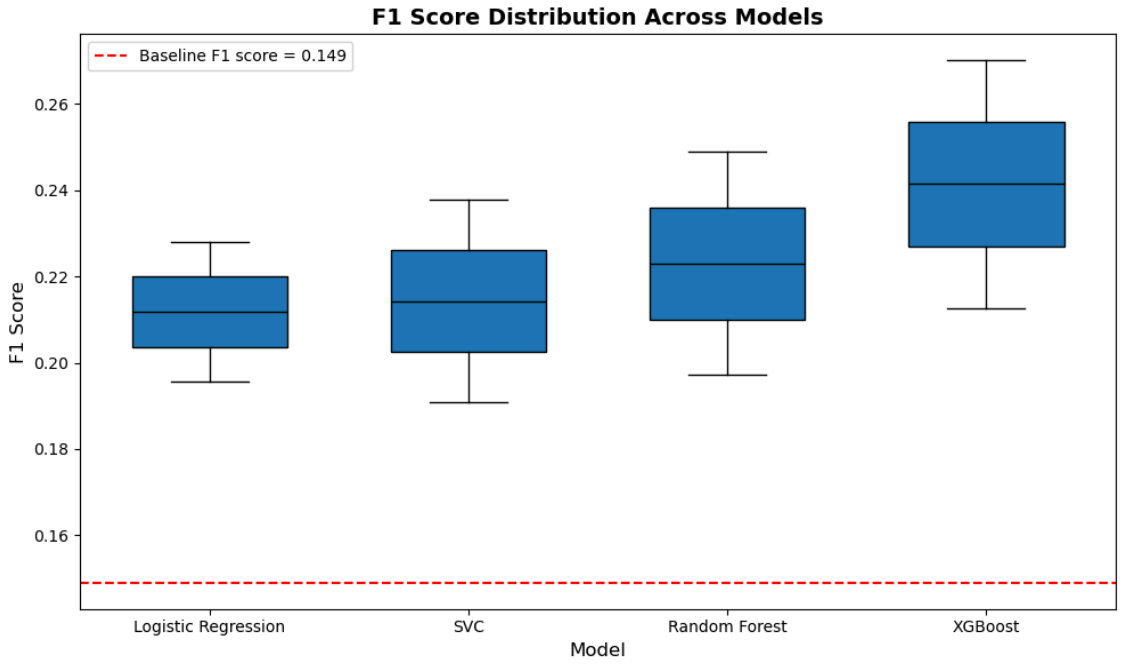


Figure 6**:** Boxplot showing the F1 score distribution for Logistic Regression, SVC, Random Forest, and XGBoost models, with the baseline F1 score as a reference.

Table 2**:** Summary of the mean F1 scores and standard deviations for each model, compared to the baseline score.

| **Models** | **Mean F1 Score** | **Standard Deviation** |
|---|---|---|
| Baseline | 0.1490 | |
| Logistic Regression | 0.2118 | 0.0162 |
| Support Vector Classifier (SVC) | 0.2143 | 0.0234 |
| Random Forest Classifier | 0.2230 | 0.0259 |
| XGBoost Classifier | 0.2414 | 0.0287 |

From the Confusion Matrices, XGBoost performed the best in predicting True Negatives, while two cases were worse than Random Forest Classifier in predicting True Positives. Taking into account that the True Negatives exceed it by a lot and the two cases are acceptable, it is still considered that the XGBoost Classifier is the best model. In comparison, Logistic Regression and SVC struggled with misclassifications, likely due to overlapping features that were harder for those models to separate.

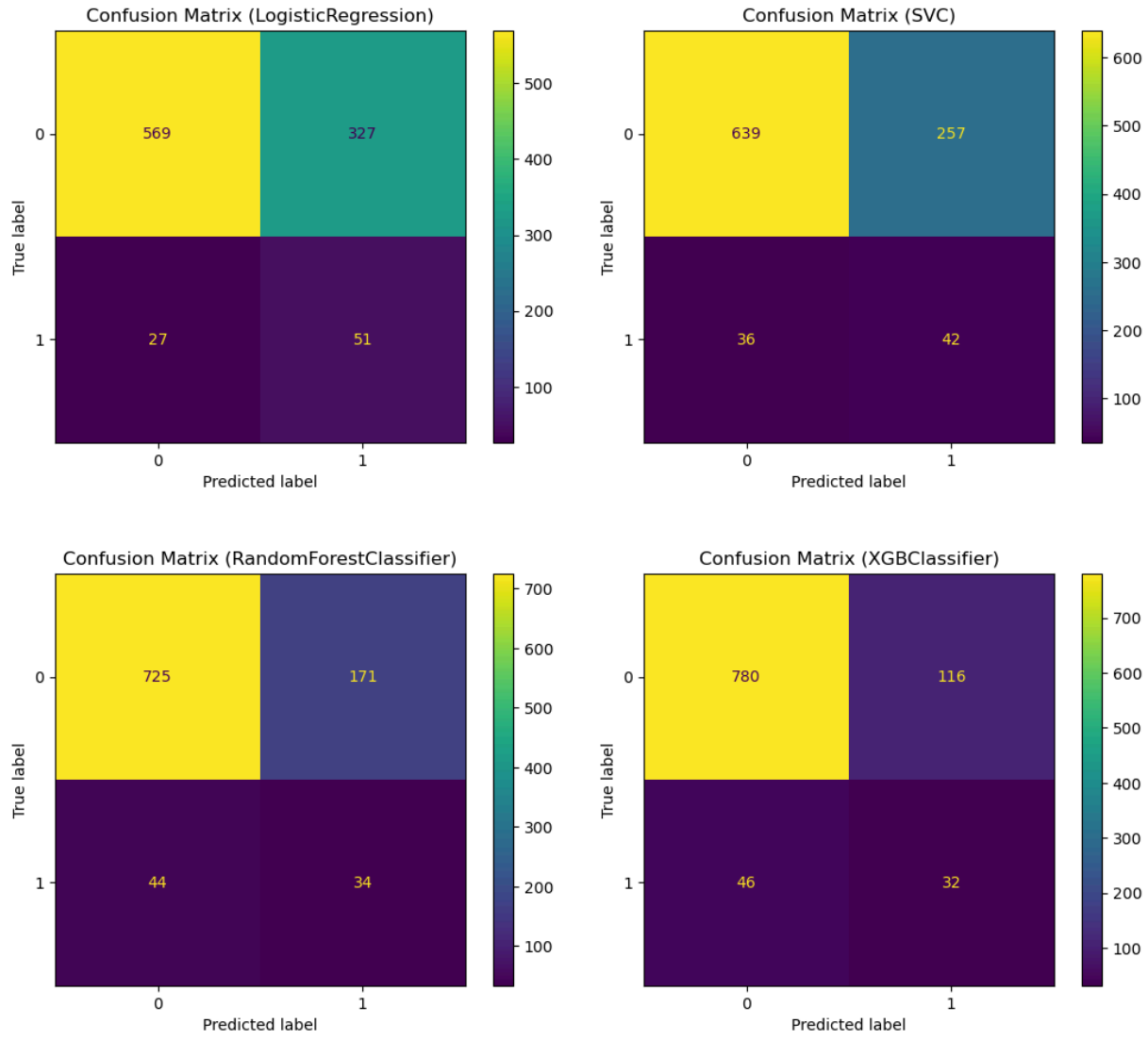


Figure 7: Confusion Matrix of all four models.

The Permutation Importance plot measures how much the F1 score drops when the values of a specific feature are randomly shuffled. The greater the drop, the more important the feature is to the model's performance. The results reveal that resume *quality*, *gender*, and *volunteer experience* have the largest impact on the model when shuffled, which suggests that these features are critical for accurate predictions. Other features like *years of experience* and *job requirements* also appear highly important, though ranked slightly lower compared to the SHAP values plot.

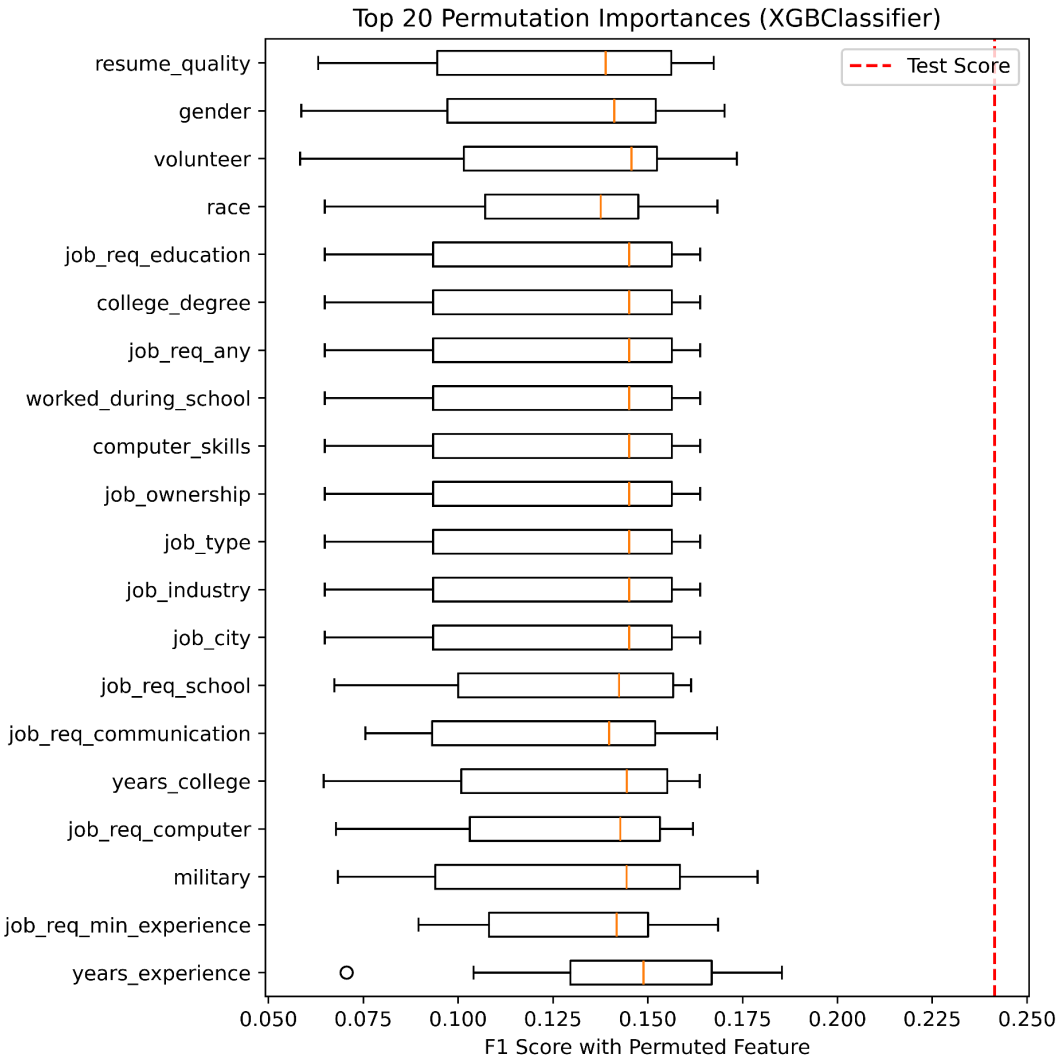


Figure 8**:** Permutation importance ranks the top 20 features based on their effect on the F1 score.

The SHAP Mean Absolute Values plot shows the average impact of each feature on the model's predictions. SHAP values measure how much a feature pushes the prediction higher or lower. The results highlight that feature *years of experience* is making the most influence to predictions, followed by *special skills* and *job equal opportunity employers*. Other features such as *employment gaps* and *race* also have significant impacts. This plot gives a clear view of how much each feature contributes to the model's predictions overall.

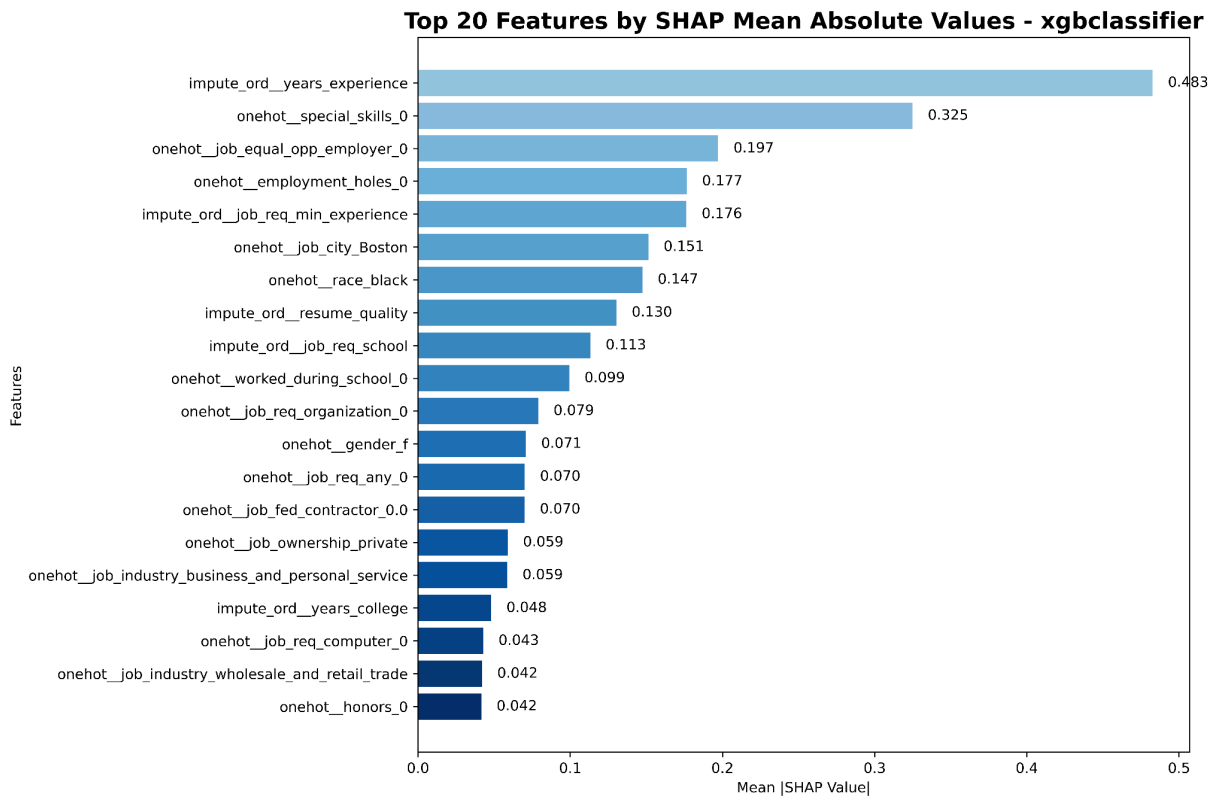


Figure 9: SHAP mean absolute values show the top 20 features with the highest average impact on the model's predictions.

The XGBoost Gain plot uses XGBoost's built-in gain metric to show which features contribute the most to reducing prediction error in the decision trees. The feature *special skills* ranks as the top contributor, followed by *job requirements any* and *job city Boston*. The gain metric indicates that these features are key to improving the model's accuracy. Additional features, such as *employment gaps* and *job type*, also play important roles in reducing the model's error during training.

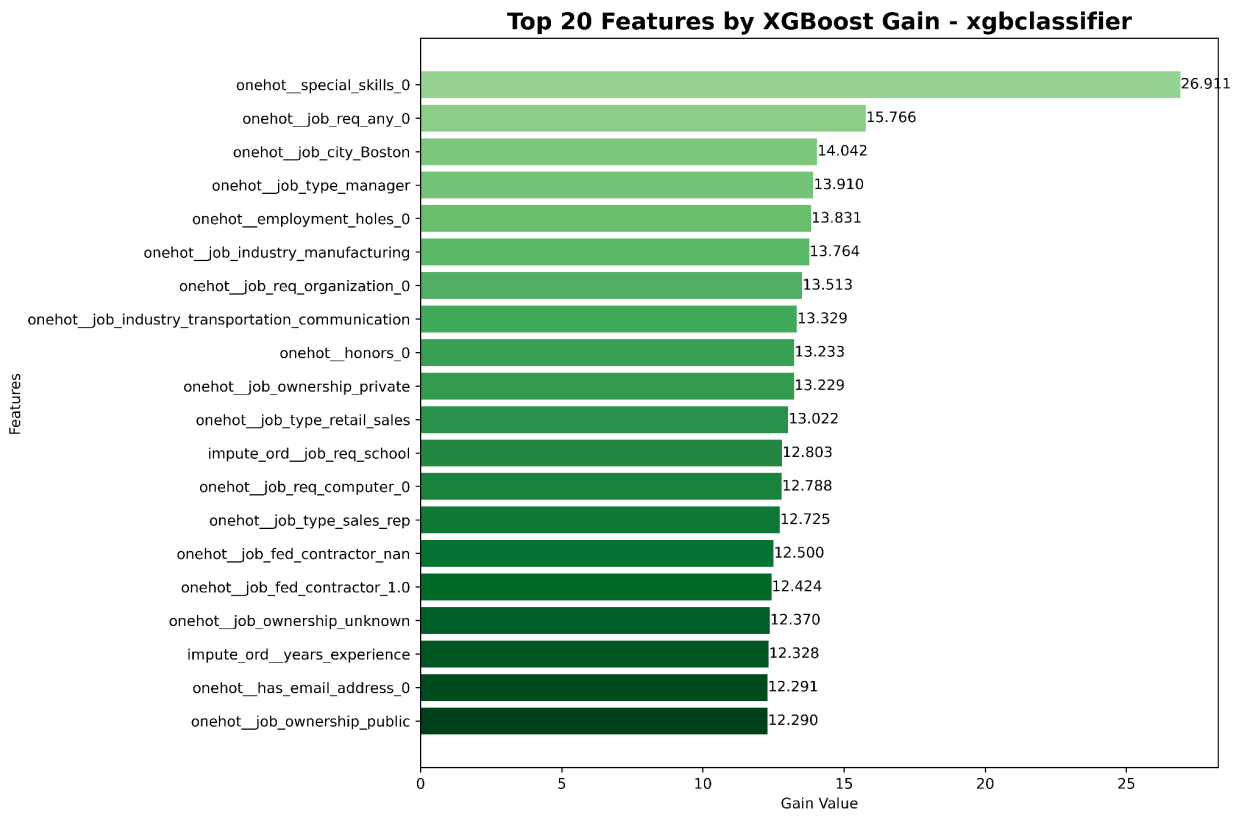


Figure 10**:** XGBoost gain highlights the top 20 features that contribute the most to reducing prediction error in the decision trees.

Despite slight variations in the rankings across the three methods, certain features consistently emerge as highly important. *Years of experience, special skills, employment gaps,* and *resume quality* are among the top features in all three analyses. This consistency reinforces their significance in the model's decision-making process. By combining the insights from SHAP values, permutation importance, and XGBoost gain, it becomes clear which features are the most influential and reliable for the model's predictions.

This SHAP summary plot below shows how individual features impact the predictions of the XGBoost model. Each dot represents a feature's effect on a single prediction, with the color indicating the feature value (red for high, blue for low).

*Years of experience* has the highest impact, with higher values pushing predictions positively and lower values negatively. *Special skills* and *job equal opp employer* also strongly influence predictions in both directions. Features like *employment holes*, *job city Boston*, and *race black* have moderate impacts, while others such as *gender f* and *worked during school* show smaller but consistent effects.

Overall, *years of experience* and *special skills* are the most critical features, with the SHAP values clearly showing their influence on the model's predictions.

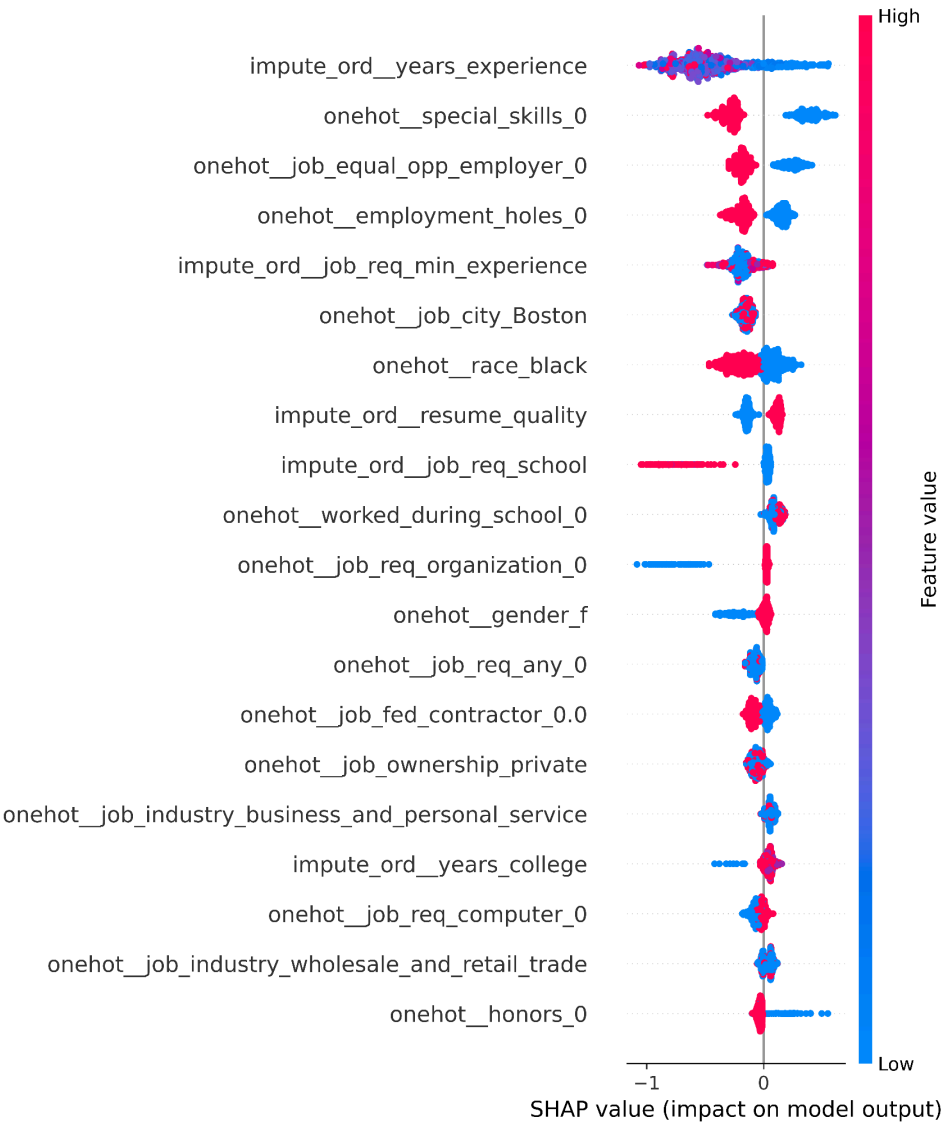


Figure 11: SHAP summary plot showing the distribution of SHAP values for the top features, with feature values color-coded to indicate their influence on the model's output.

For observation 525, this prediction has a log-odds value of -1.27, and the corresponding probability is 0.219 which is being classified in class 0. *Job required school* significantly impacted the prediction negatively. Features such as *race black* and *special skills 0* also contributed negatively to the prediction.

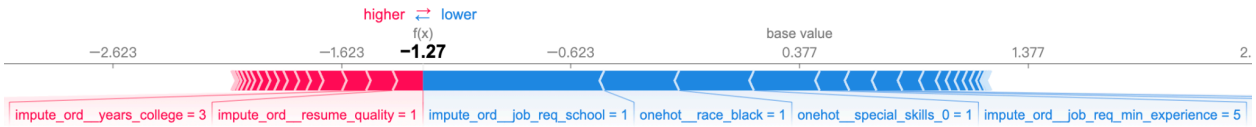


Figure 12: SHAP force plot for observation 525

For observation 777, the corresponding probability predicted is 0.455, which is also in class 0. *special skills 0* made a positive influence on the prediction. However, contribution from feature

*years_experience* outperformed that of the positive side.


Figure 13: SHAP force plot for observation 777

## Outlook

One key limitation of this analysis is that the dataset is a bit outdated. The job market evolves rapidly, and the features influencing callbacks or job success may have shifted. Collecting new data that reflects the current job market trends and requirements would likely improve the model's relevance and predictive accuracy. This updated data could better capture emerging skills or industry shifts that are not present in the current dataset.

Another avenue for improvement lies in expanding the hyperparameter space for model optimization. While the current set of hyperparameters is effective, testing additional parameters or ranges—such as different learning rates, tree depths, or feature sampling methods—might uncover configurations that yield better results. A more thorough search could also help address potential biases introduced by limited parameter choices.

Lastly, principal component analysis (PCA) could be applied to assess whether some features can be reduced without significantly affecting the model's performance. By identifying and removing unnecessary features, PCA can streamline the dataset, making the model more computationally efficient while retaining its predictive power. This approach could also reveal underlying patterns or relationships in the data, adding depth to the analysis.

Combining these strategies—updating the dataset, expanding hyperparameter tuning, and exploring PCA for feature reduction—would make the model both more robust and applicable to real-world scenarios. These improvements would enhance the overall quality, efficiency, and impact of the predictive system.

(Word Count: 1980)

## Reference

Bertrand, M., & Mullainathan, S. (2004). Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *American Economic Review, 94*(4), 991-1013. https://doi.org/10.1257/0002828042002561

Utkarshx27. (n.d.). *Which Resume Attributes Drive Job Callbacks* [Dataset]. Kaggle. From https://www.kaggle.com/datasets/utkarshx27/which-resume-attributes-drive-job-callbacks