JOB CHANGE PREDICTION

Sai Ravi Karthik Mandapaka
Siddharth Gupte
Kalani Paranagama

October 23, 2022

Abstract

During the hiring process, it is useful for an HR department to know if a candidate is truly interested in working for the company. The organisation will save substantial time and money by knowing whether a candidate would stay in the role after being hired. Therefore, it would be bene cial for an HR Department throughout the hiring process if it was possible to forecast whether a candidate would stay or depart based on their past behaviour. This study aims to create a classi er that can predict whether or not a candidate will change their job by using historical data on candidate demographics and details about their education and experience. This study compares the performance of three di erent classi cation models, namely, Logistic Regression, Decision Tree Classi er and XGBoost Gradient Descent classi er, using the evaluation metrics of F1 score, Accuracy, Precision and Recall. Because accuracy is a poor performance indicator in the situation of an imbalanced class label that existed in the raw applicant dataset, the F1 score is our primary evaluation metric. The experimental results show that the XGBoost Gradient Descent classi er had the best performance compared to the other models, with an F1 score of 0.84 and showing a larger area under the ROC curve. Therefore, a machine learning pipeline that could classify unseen, unlabeled applicant data was established using Python, which utilises this XGBoost Gradient Descent classi er with further tuned hyperparameters.

Table of Contents

1. Introduction

A main goal of an HR Department in a company is to ensure the longevity of the candidates they hire within the company. Understanding whether a candidate is likely to remain in the position when hired will save the company a signi cant amount of time and money. Therefore if it is possible to predict an outcome where the candidate will stay or leave using historical records, this would immensely bene t an HR Department during their hiring process.
The goal of this study was to develop a classi er using historical data on candidate demographics and information about their education and experience, that can predict whether or not they will change their job. The dataset in question contains information about Data Scientists who completed courses o ered by the company. Three di erent classi cation models (X, Y and Z) were developed using labelled data (where it was speci ed if yes, the candidate left the job and if no, the candidate did not leave the job) and tested on unlabeled data. The models were then compared using the evaluation metrics of F1 Score, Precision and Recall and the classi er with the best possible predictive outcomes was selected.

2. Pre-processing and features used

2.1 Data Pre-processing

Data pre-processing is the process of extracting useful information from raw data and converting it into a format that can be understood by machine learning models. This section describes the steps that were taken to pre-process the data

and extract a set of useful features prior to the application of classi ers on the dataset..To begin the data pre-processing, the dataset was explored to detect any inconsistencies and outliers. Although there were outliers detected in City Development Index and Training Hours columns, these records were not excluded as removal of these may result in potential loss of critical information (since there was a signi cant number of outliers).

2.1.1 Pre-processing of numerical variables

Binning was performed on the Experience variable to reduce the number of categories from 22 to 5 using a set of appropriate conditions. This allowed for fewer categories when performing label encoding at later stages. However, the original Experience variable was retained in the dataset when feature selection was performed.

2.1.2 Encoding for categorical variables

Categorical Features must be converted to a numeric format to be utilized in machine learning models such as those described in this report, which only take numeric data as input. This can be done by applying various encoding methods to the categorical features such as one-hot encoding and label encoding (Cohen, 2021). As shown in Figure 1, if there was a direct increase or decrease in trend across categories then label encoding was performed with the assumption that there was an inherent ordering within the feature categories. The features for which ordinal label encoding was applied (using LabelEncoder function from Scikit learn) were Enrolled University, Relevant Experience, Last New Job. The target variable was also encoded with 1 indicating 'Yes' and 0 indicating 'No'. One-hot encoding was performed on the remaining variables of Gender, Academic Discipline, Company Type, Company Size and Education Level.

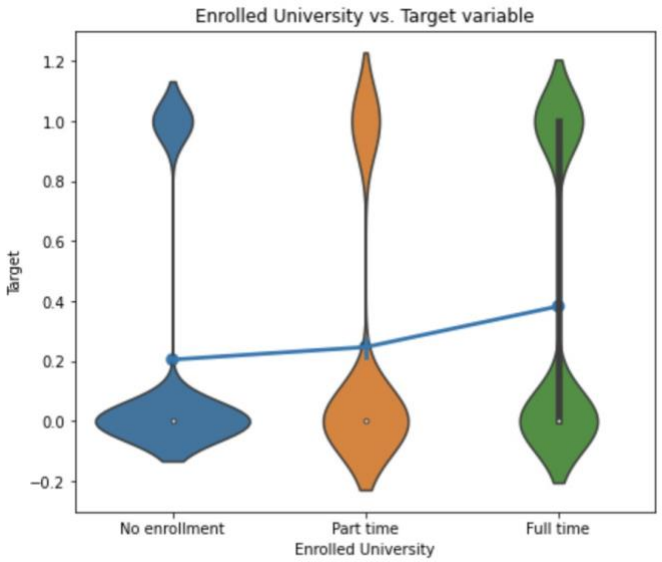


Figure 1: Violin plot and point plot showing the relationship between Enrolled University and Target variable. This indicates that candidates who were enrolled in university full time were more likely to switch jobs. There is an increase in the trend across the three categories of this feature.

2.2 Feature selection

Once encoding was completed, the process of feature selection was carried out to nd the set of most essential features. This allows the removal of redundant features and the training of a model with lower complexity and greater accuracy. Six standard methods of feature selection were applied to the encode data which were; selection based on Correlation Coe cient, Information Gain, Chi-square test, ANOVA F-Test, Variance threshold and Recursive Feature Elimination. Next, the features that appeared most frequently within the resultant feature sets from the above six methods were selected as the nal feature set. There were 30 features in this nal feature subset. This method of selecting

the nal feature set was utilised to maximise the likelihood of singling out the most useful features that will allow us to build the most accurate model, rather than applying just a single feature selection method.

2.3 Application of Sampling methods (SMOTE)

As there is a class imbalance issue with the training dataset, which consists of historical records where the majority of employees reported not changing jobs, the SMOTE oversampling technique was applied to deal with the imbalanced distribution of the target variable. SMOTE rst selects a minority class instance at random and nds its k nearest minority class neighbours. SMOTE works because it generates convincing new synthetic examples from the minority class that are substantially near in feature space to already existing examples from the minority class. SMOTE was performed using the 'smote' function of the imbalance-learn package.

Fig. 2 displays the distribution of of the target variable prior to and after oversmapling. As seen in the dstibuttions application of SMOTE recti ed the class imbalance issue.

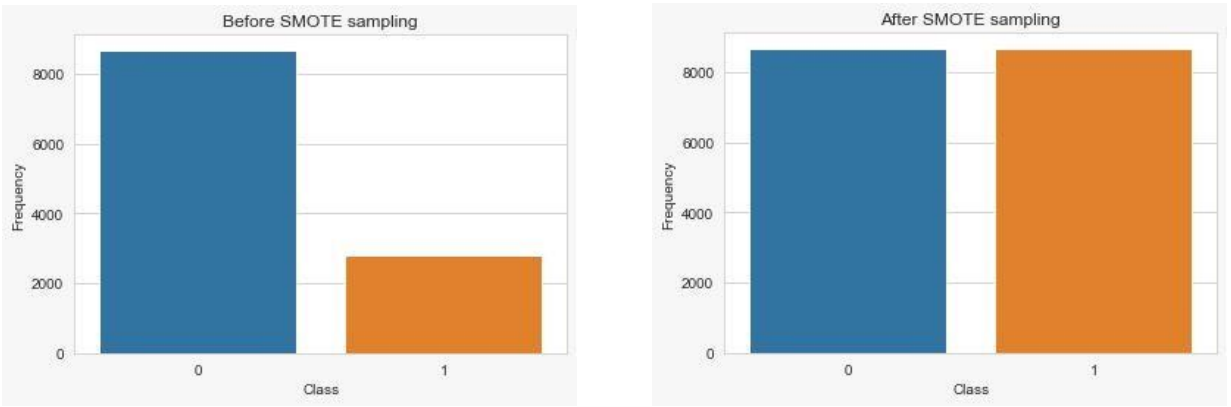


Figure 2: Distribution of target variable before and after SMOTE sampling.

3. Models

The pervious section outlined the methodology that was applied to acquire the set of transformed features. This section discusses the models which were implemented on the wrangled and feature selected data. The three models that were applied to the data were:

- Logistic Regression
- Decision Tree Classi er
- XGBoost Classi er

Each of these models are well suited to a two-class classi cation dataset such as the one we have at play here. We have chosen three models with increasing model complexity from Logistic Regression to XGBoost Classi er and hence increasing degree of freedom, as models with di erent complexities are able to capture di erent levels of relationships.

3.1 Model 1: Logistic Regression

Logistic regression is a supervised learning algorithm that describes the relationship between one binary dependent variable and a set of one or more independent variables. Therefore, the dependent variable will have only two values. For eg, yes and no or 1 and 0.

It uses the sigmoid activation function that takes a certain input and produces an output value which lies within the range of 0 and 1. The activation function can be given as follows:

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

The activation function tests di erent values of the model coe cients through multiple iterations to give the best t of log odds. (*What Is Logistic Regression? | IBM*, n.d.-a)

3.2 Model 2: Decision Tree Classi er

The Decision Tree algorithm uses a set of principles to make a decision. It is typically said to forecast the class of the novel, previously unseen input, but in reality, the algorithm must choose the class to be assigned. The idea is to repeatedly partition the dataset until all the data points that belong to each class are isolated by using the dataset features to produce yes/no questions. The entire process resembles a tree like structure where the rst node of this tree is called the root node.
The result at a particular node divides the dataset according to the importance of a feature and adds additional nodes. The nal nodes formed are known as leaf nodes if you want to terminate the process after a split. The technique attempts to
fully divide the dataset such that each leaf node, or node that doesn't further divide the data, belongs to a single class. We refer to these as pure leaf nodes. (Refer Fig.1 in appendix)

3.3 XGBoost Gradient Descent Classi er

For classi cation and regression, a Gradient Boosted Tree is an ensemble tree based learning approach. Extreme Gradient Boosting (XGBoost) is a distributed, scalable gradient-boosted decision tree machine learning framework. It o ers parallel tree boosting, making it a more regularized version of the gradient boosting algorithm.

3.4 Discussion of model di erence(s)

Logistic Regression	Decision Tree Classi er	XGBoost Classi er
Divides the decision into exactly two regions.	Divides the decision into multiple smaller regions.	Ensemble of decision trees.
Linear boundary.	Non-linear boundary.	Non-Linear Boundary.
Logistic Regression's straightforward linear boundary generalises better when the classes are not properly separated.	Trees are prone to over tting the training data when classes are not clearly divided.	Bad separation between classes is handled better when multiple weak decision trees are combined.

4 Final Model Selection Methodology

The selection process for the nal model had the following objectives:

- Choose the best model out of three known models in terms of performance and computing costs with the set of features obtained from the wrangling and feature selection process.
- Choose the best subset of the top features based on the transformed features acquired from the feature selection process.
- For the chosen model and subset of features, select the best values to be passed as parameters of the model through an optimized process called hyperparameter tuning.

4.1. Training Method

Initially, all features were fed into the three models. For this process, the dataset was split into a training set and a validation set with a split ratio of 80/20. Furthermore, K-fold cross validation was applied to the 80% training set. Cross validation is the process of splitting up the dataset into k equal sized subsets with the model being tted on k-1 subsets and applied to the kth subset. For the Logistic Regression and Decision tree models, we used Sklearn's CV feature. XGBoost has its own CV function that was employed here. Given that the dataset only contains a few 10000 records, the number of folds was set as 5.

4.2. Performance Metrics

Here is the list of performance metrics that were employed to evaluate the performance of each model.

Performance Metric	Description
F1 Score	The harmonic mean of precision and recall to compare performance of a classi er with better precision and a classi er with better recall
Accuracy	The ratio of correct predictions to total predictions.
Precision	A classi cation model's capacity to recognise only the pertinent data points. It is the product of the number of true positives and the sum of the true positives and false positives.
Recall	The number of true positives divided by the total number of true positives and false negatives
AUC	This is a metric for classi ers' propensity to di erentiate between di erent classes.

However, as per the interest of the HR department, we went ahead with the F1 score as the barometer for selecting the nal model.

4.3. Selecting the Top Feature Subset

As discussed in section 2.2, we were given the set of transformed features. These features were gathered from the data wrangling, feature selection and SMOTE sampling process. In this section, we will discuss why it was important to not use all the features for the nal model. There are two main reasons:

- Over tting: The usage of too many features can result in over tting, especially because we have a repetition of features encoded in di erent ways in this set. This may result in a lot of noise being fed to the nal model.

- Computation Power: The more features we have in our nal model, the more complex will its computation be, especially during the hyperparameter tuning process.

4.4. Hyperparameter Tuning Approach

The two main concerns while performing the hyperparameter tuning process were the combination of parameters and Optimization of parameters. The Hyperopt package in Python was utilised to employ a bayesian estimation method to select the best combination of parameters for the nal model. Hyperopt acts as a additional layer on top of the model such that, if the XGBoost model derives the best result from an ensemble of trees, the Hyperopt optimizer derives the best result from an ensemble of XGBoost models built with di erent combinations of parameter values with the given features.

5 Experimental Results

This section describes the results of each step of the nal model selection process and the optimization of the nal model with respect to the given features.

5.1. Model Performance: A Comparative Glance

The table/comparision of evaluation metrics is given as follows based on train-test split of the training data for all features.:

	Model	Accuracy	F1_score	Recall	Precision	AUC
0	1- Logistic Regression	0.829126	0.82637	0.81367	0.839521	0.829112
1	2 -Decision Tree	0.789707	0.790924	0.795718	0.786221	0.789712
2	3 -XGBoost Classifier	0.855373	0.844582	0.786167	0.912375	0.855353

Even though the the XGBoost Classi er is the clear winner based on the F1 score, it also surpassed the other two models in terms of the other performance metrics even though an argument may be made for the logistic model which comes close in terms of performance..

To solidify its dominance, the ROC curve for the three models were compared as shown in g. 3.

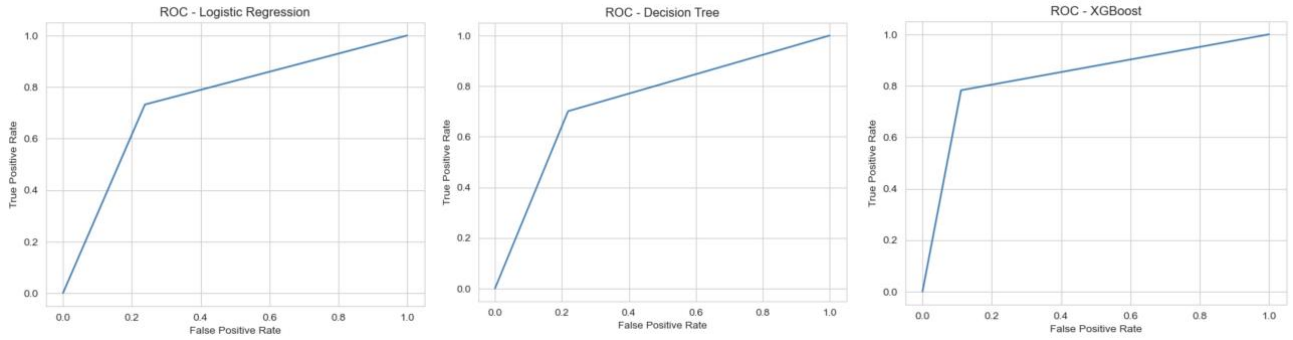


Figure 3: ROC (receiver operating characteristic curve) for the three classification models

The area under the ROC curve for XGBoost is larger than the curves for the other two models, indicating a much better di erentiation between the classes..

5.2. Top Feature Subset Selection

The following plot shows the average impact on the model output indicating how well each feature contributes to the prediction of the target variable.

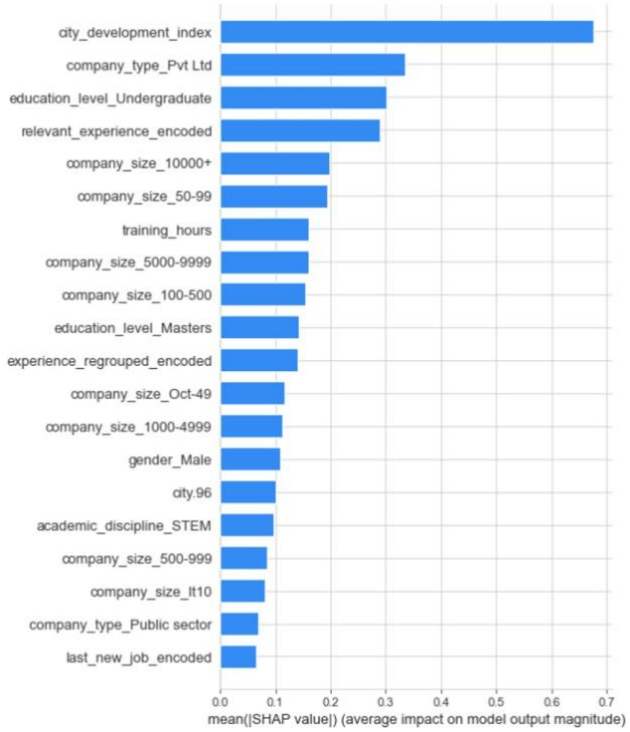


Figure 4: Average impact of features on model performance

The top 5 features in the above plot were included in the nal feature subset, along with the one extra feature (enrolled university encoded) which we took from the top 30 features as it had a large in uence on target feature when we carried out the initial feature selection.

5.3. Final List of Parameters

Through the bayesian optimization technique o ered by Hyperopt, the following combination of parameters and their respective values was selected for the nal XGBoost model:

max_depth : 16,
learning_rate : 0.13
n_estimators : 350
objective :
binary:logistic booster :
gbtree n_jobs : 1
nthread : None

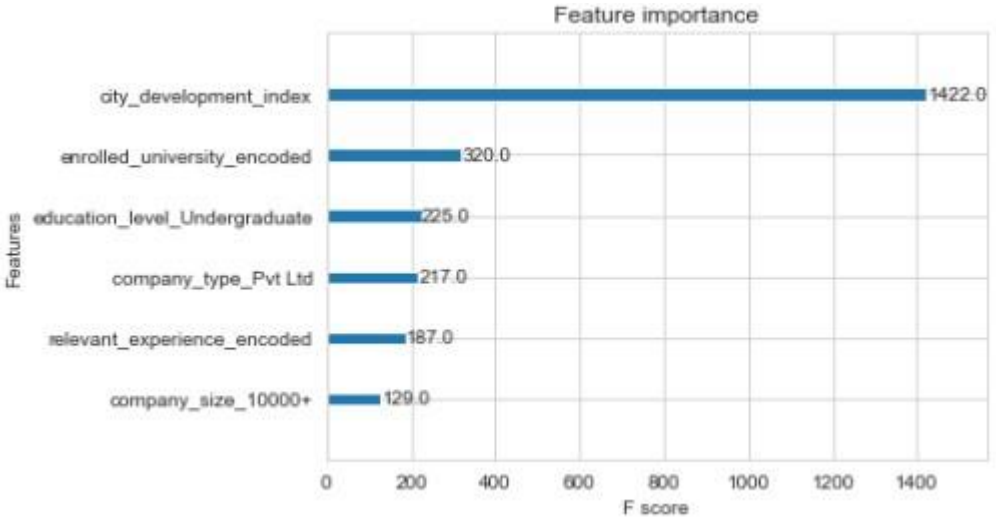
nthread : None
gamma: 7.2
min_child_weight : 1
max_delta_step : 0
subsample : 1
colsample_bytree : 0.7
colsample_bylevel : 1

reg_alpha : 0
reg_lambda : 0.5
scale_pos_weight :
1 base_score : 0.5
random_state : 0
seed : None
missing : 1

5.4. Model Inference

First, let us plot the top ten relevant features with respect to the target variable using the XGBoost feature importance as shown in Fig.5. Generally, importance provides a score that indicates how useful or valuable each feature was in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions in the ensemble, the higher its relative importance.

Figure 5: Feature importance plot derived from the XGBoost model for the top features



5.5. Pipeline

Finally, custom pipeline was built to combine the data-preprocessing, feature selection and resampling steps (as described in section 2) and apply the best performing XGBoost Classi er on the resulting dataset to predict the target variable. This pipeline (which was created using the 'pipeline' class of the Sklearn package) was then applied to the test dataset to obtain the nal set of predictions (pred_labels.csv).

As shown g. 6 (below) a number of custom transformer classes were de ned, extending the functionality of the bases classes (e.g. BaseEstimator, TransformerMixin) to acheive the steps necessary to make nal predictions.

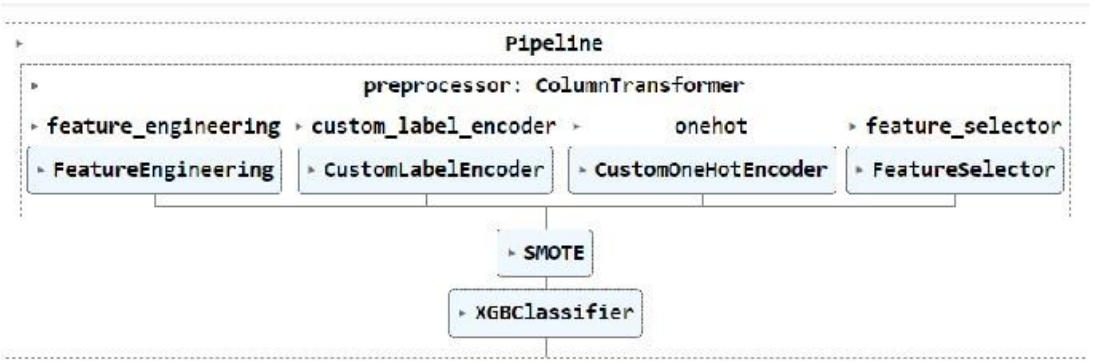


Figure 6: Visualization of the machine learning pipeline

6 Conclusion

As described in section 5, the optimal classi er that couod carry out the job change prediction was the **XGBoost Gradient descent classi er which provided an F1 score of 0.84 on the training dataset,** performing better than the other models that were tested. **The best set of features used by this classi er included City Development Index, Enrolled University, Education Level (Undergraduate class), Relevant Experience and Company Size (>10000+)** which were obtained by examining the feature importance criteria provided by the model. As discussed in section 5.2, city development index was one of the most important features (from the feature importance plot). Upon examining the dataset more closely, it could be see that as city development index increased so did the likelihood of a candidate switching their job. This may be explained by the fact that in areas where development is high would tend to have more job opportunities thereby increasing the likelihood of switching jobs. A similar trend could be observed with some of the other optimal features where a signi cant positive or negative correlation could be seen between those features and the target variable (for example relevant experience and company size).

Some future recommendations would be:

- Provision of a larger dataset that is more representative of the population.
- Include more relevant features in terms of what would motivate a candiate to switch jobs (e.g. features indicating their satisfaction with their current job)
- Using various data science techniques to create new features based on the interaction of these variables.

References

A Guide on XGBoost hyperparameters tuning. (n.d.). Kaggle.com.

https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning/notebook

Cohen, J. (2021, December 25). *Categorical Feature Encoding - Towards Data Science*. Medium.

https://towardsdatascience.com/categorical-feature-encoding-547707acf4e5

What is Logistic regression? | IBM. (n.d.-b). Www.ibm.com.

https://www.ibm.com/au-en/topics/logistic-regression

XGBoost + k-fold CV + Feature Importance. (n.d.). Kaggle.com.

https://www.kaggle.com/code/prashant111/xgboost-k-fold-cv-feature-importance/notebook

Appendices

Fig 1: Decision Tree Implementation

Decision tree trained on the best features