Prediction of Job Change

Data Mining Final report

Group 1

Renping Ge, Yuchen Ma, Zongzhu Li

2021-04-30

1. Introduction

Data science is a new industry. Since it gets more popular in society, more companies need data scientists. The company which is active in Big Data and Data Science wants to hire data scientists among people who successfully pass some courses which are conducted by the company. Many people sign up for their training, but not all people will work for the company after training. Therefore, the company wants to know which of these candidates really wants to work for the company after training or looking for a new employment. That is the reason why we explore this topic. In this project, we aim to explore the probability of a candidate to look for a new job or will work for the company after training, as well as interpreting affected factors on employee decision. This project can help the company to reduce the cost and time as well as the quality of training.

The structure of our report includes the following part: description of dataset, description of algorithm that we use in our model and cleaning, experimental setup, results, summary and conclusions, and references.

2. Description of Dataset

The dataset we select for our analysis is from Kaggle competition website. This dataset is collected by the company and designed to understand the factors that lead a person to work for the company (leaving their current job). The dataset needs to be cleaned since it's imbalanced and contains missing values. Besides, most features are categorical, so it needs to conduct some data cleaning and preprocessing. The dataset has 19158 entries and 14 columns.

The following table explains each column, and we could also see more details of our dataset in the appendix.

Column_name	Description
Enrollee_id	Unique ID for enrollee
city	City code
citydevelopmentindex	Development index of the city
gender	Type of University course enrolled if any
Releven_experience	Relevant experience of enrollee
Enrolled_university	Type of university course enrolled if any
Education_level	Education level of enrollee
Major_discipline	Education major discipline of enrollee
experiences	Enrollee total experience in years
Company_size	Number of employees in the current employer's company
Company_type	Type of current employer
Last_new_job	Difference in years between previous job and current job
Training_hours	Training hours completed
target	0--Not looking for job change 1—Looking for job change

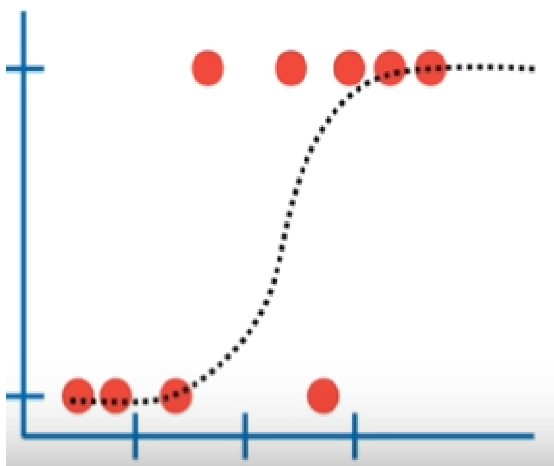
3. Description of Algorithm

3.1 Classification Algorithm

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data. In Classification, a program learns from the given dataset or observations and then classifies new observations into several classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories. Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

3.2 Logistic Regression

In this project, we use logistic regression which is a classification algorithm and is used to predict a binary outcome based on a set of independent variables. The logistic regression is named for the function using logistic function.



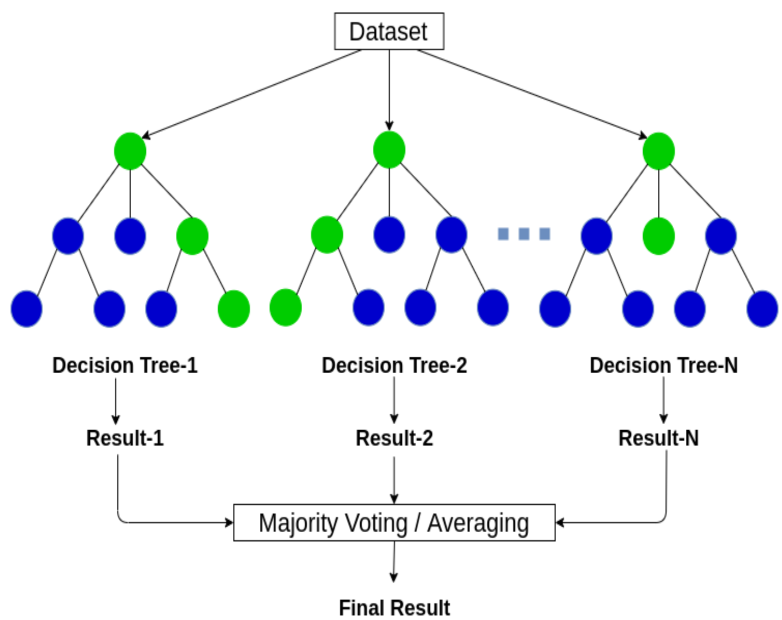
It models the probabilities for binary classification question. The logistic function is defined as below:

$$logistic(\partial) = \frac{1}{1 + \exp{(-\partial)}}$$

3.3 Random Forest

We also use random forest which is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.



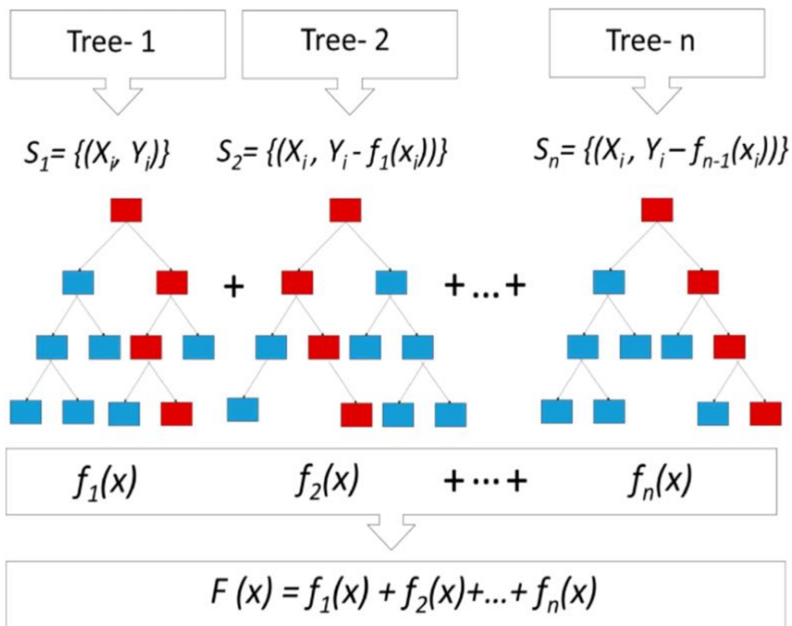
3.4 Gradient Boosting

Gradient boosting is an algorithm that could overfit a training dataset quickly. It gets benefit from regularization and generalizes item by allowing optimization of loss function. The equation of binary cross entropy loss is shown as below. If we have a random variable X, then the pdf of X would show as below:

$$s = \begin{cases} -\int p(x).logp(x)dx & if\ x\ is\ continuous \\ -\sum p(x).logp(x) & if\ x\ is\ discrete \end{cases}$$

In this case, the loss function for the dependent variable y would be:

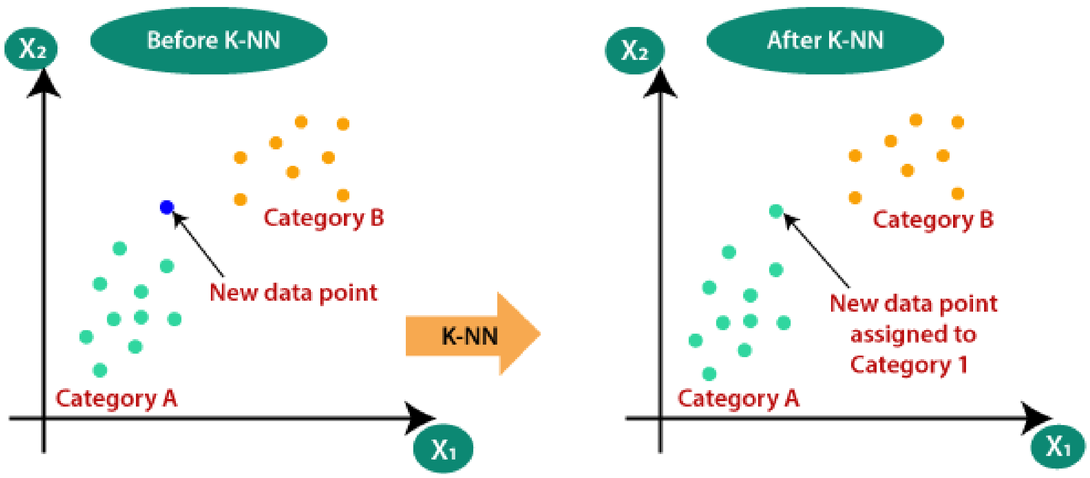
$$L = -y * \log(p) - (1 - y) * \log{(1 - p)}$$



And the above graph shows the principle and the process of gradient boosting.

3.5 K-Nearest Neighbor (KNN) algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.



3.6 Model Evaluation Statistics

3.6.1 Accuracy

The accuracy is the ratios of correctly predicted observations to the total number of observations. Here, TP means true positives, TN means true negative, FP means false positives, and FN means false negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.6.2 Precision

Precision is the ratio of correctly predicted positive samples to the total of predicted positive observations

$$Precision = \frac{TP}{TP + FP}$$

3.6.3 Recall

Recall is the ratio of TP to all the observations in actual area.

$$Recall = \frac{TP}{TP + FN}$$

3.6.4 F1-Score

F-1 score is used to evaluate the classification model based on precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

3.6.5 ROC_AUC Score

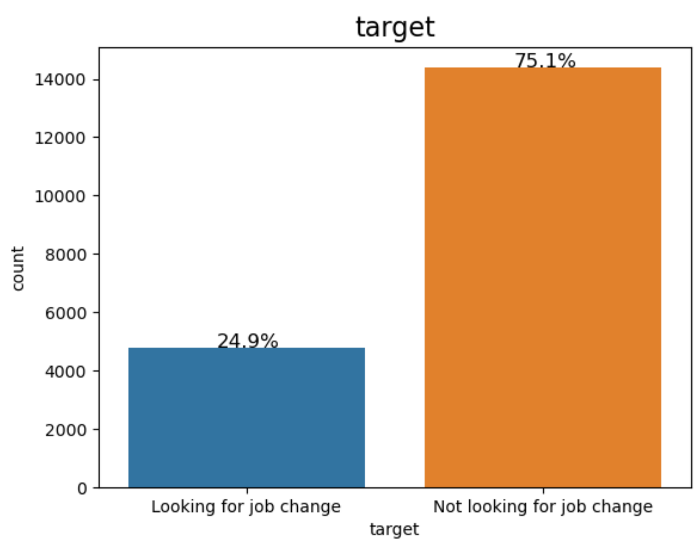
The AUC means the area under the curve. Normally, we would see the model is well when the ROC_AUC score is larger than 0.5

4. Experimental setup

4.1 Data preprocessing

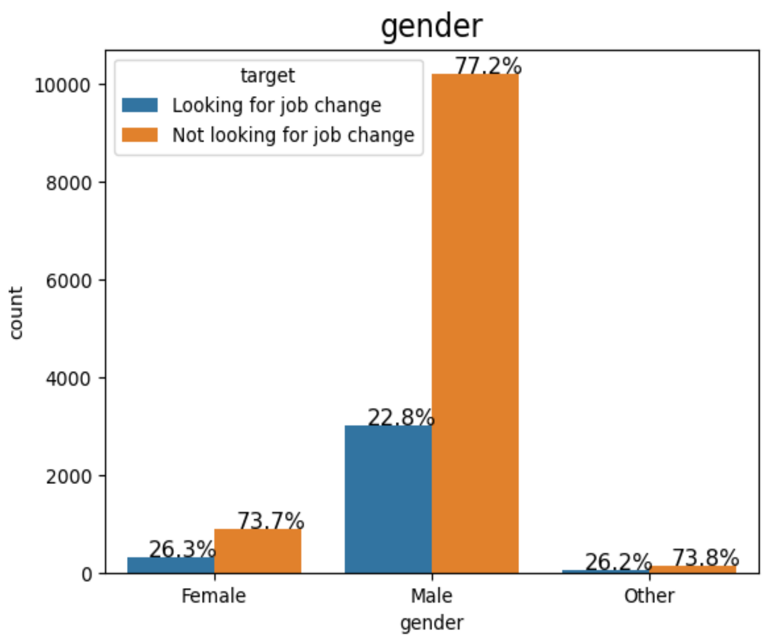
Firstly, we specified the predictor and target variable to make sure they are at the right position. Secondly, filling NA with mean was executed on all numerical variables, the NA in categorical variables were left to be applied in the modeling. Standardization and centralization for numerical variables were executed with StandardScaler(). Encoding categorical features was also done with OneHotEncoder() and the target variable was labeled with LabelEncoder(). Lastly, the dataset was spitted into training and test groups randomly with training taking 70% and test for left 30%.

4.2 Exploratory Data Analysis



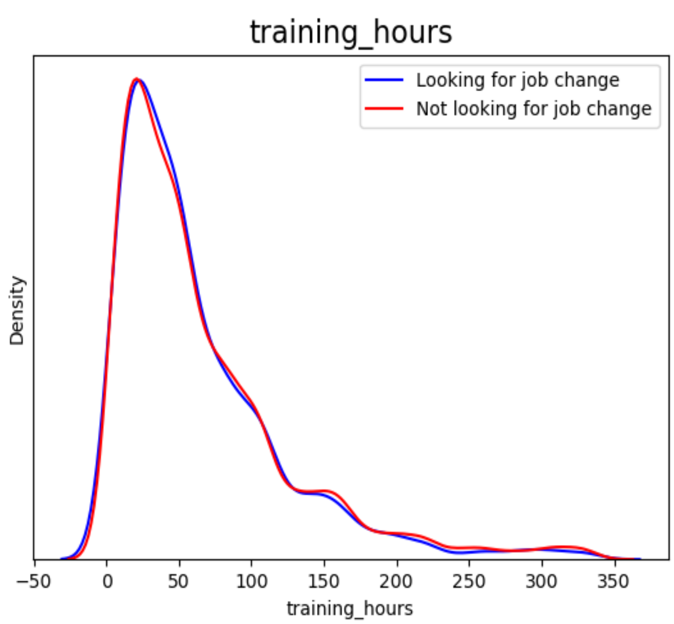
Counts and rates for people looking for job change

Over 19158 enrollees, 24.9% of them are looking for job change and 75.1% of them are not looking for a job. We can see the result of above figure.



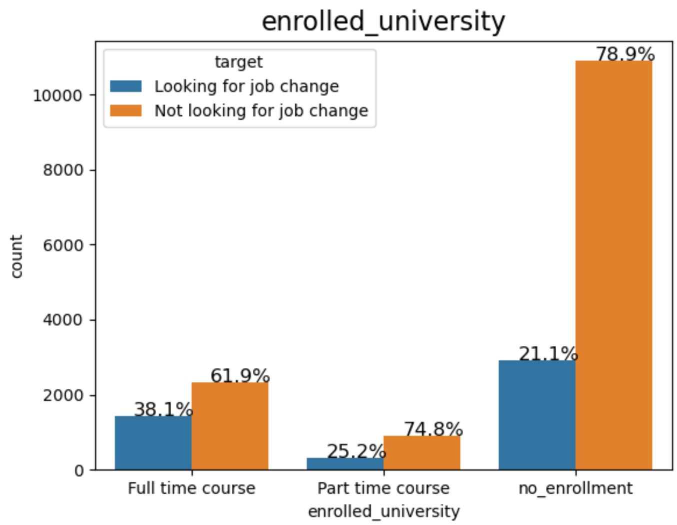
Distribution of job change by gender

With different genders， people shows a comparable rate of looking for a new job. We can see the result from above figure.

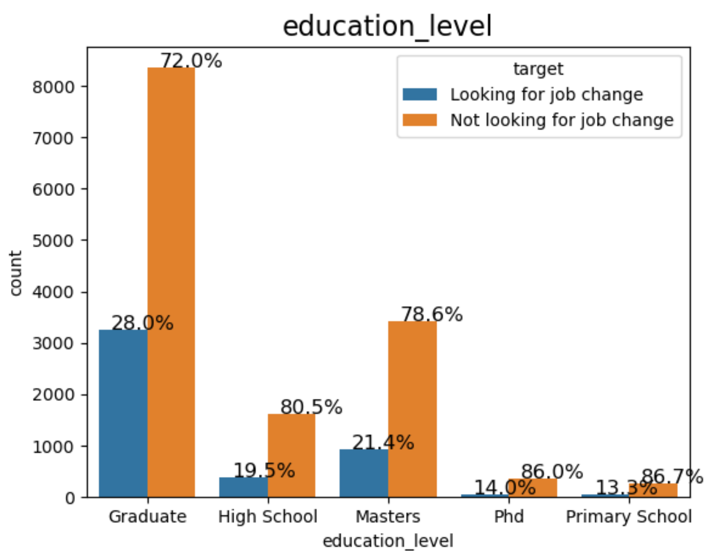


Distribution of job change by training_hours

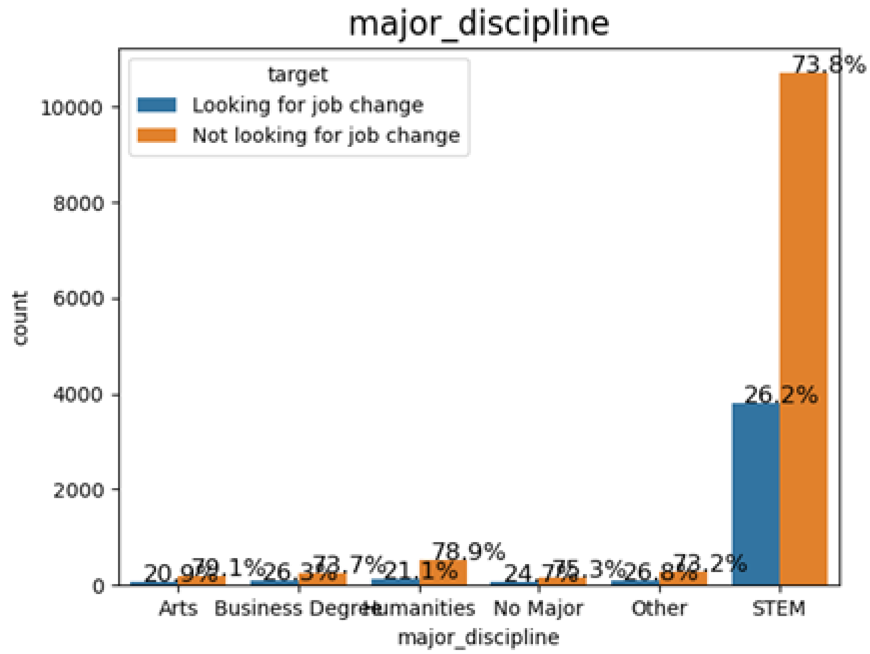
People with different training hours show a comparable rate of looking for a new job. We can see the result from above figure.



Distribution of job change by enrolled_university



Distribution of job change by education_level

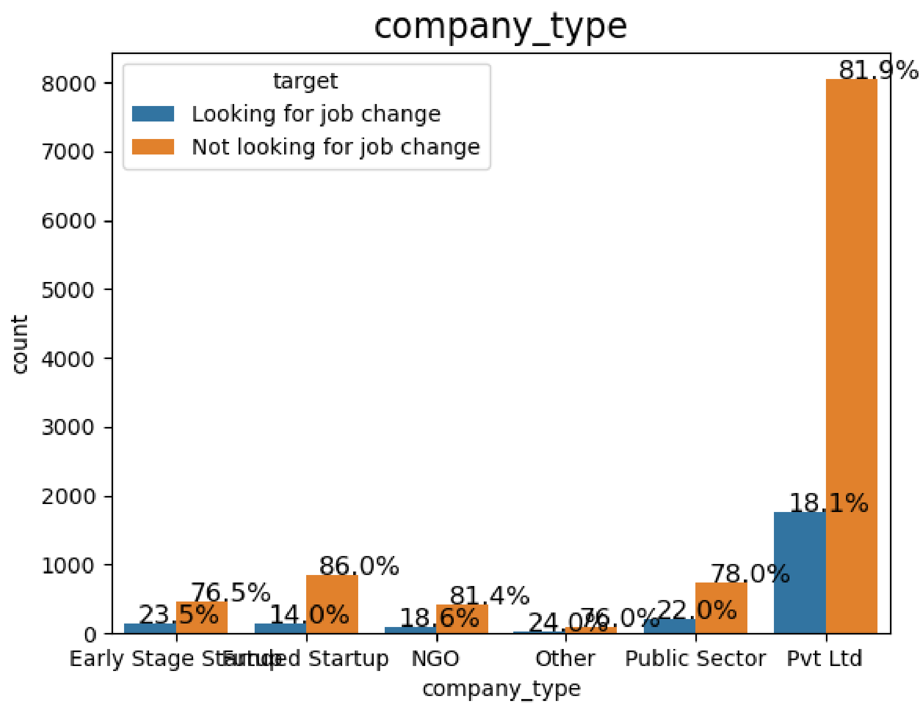


Distribution of job change by major_discipline

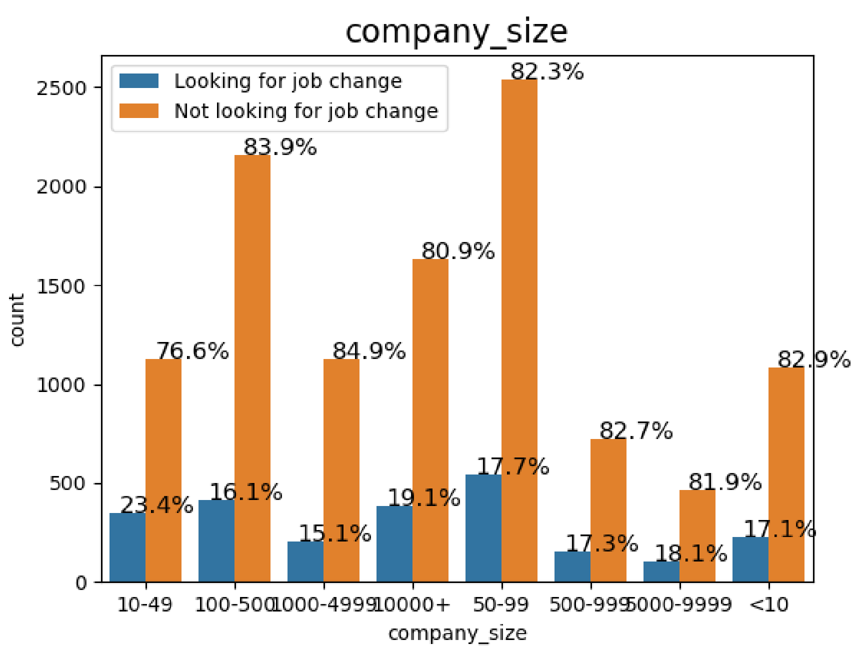
Education plays an important role in the rate of people looking for job change. People who took the full time course are more likely to look for a new job compared to others taking part-time course and no enrollment. People with graduate education level are more inclined to look for a new job compared to high school, masters, Ph.D and primary school. People with discipline of art and humanities are less likely to look for job change compared to the people with discipline of business, STEM and others.

Distribution of job change by city_development_index

City development index stand for the development level and stages, it is very interestingly that in the cities with lower city_development_index, the rate of people looking for a new job is significantly higher than that in the cities with higher city_development_index(Figure.7), which suggests that there are more opportunities in the development cities.

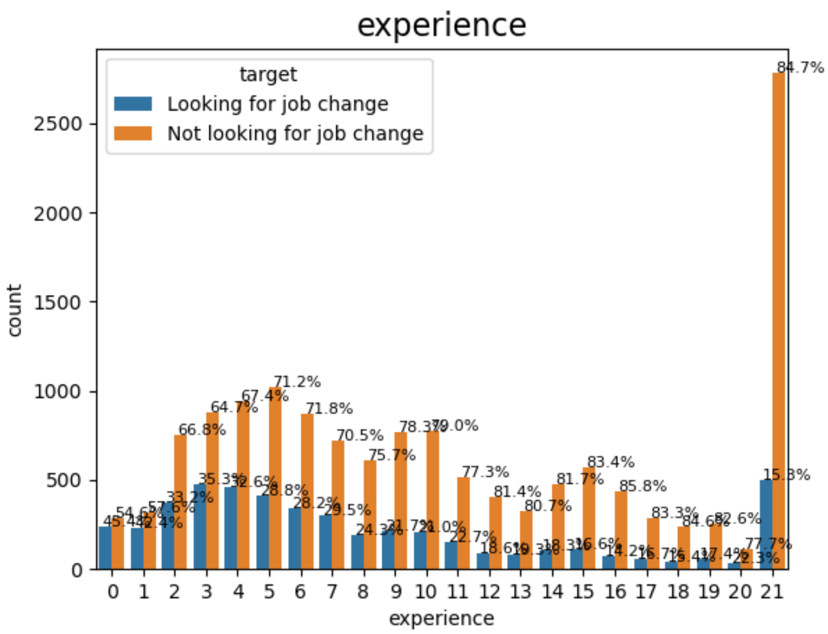


Distribution of job change by company_type

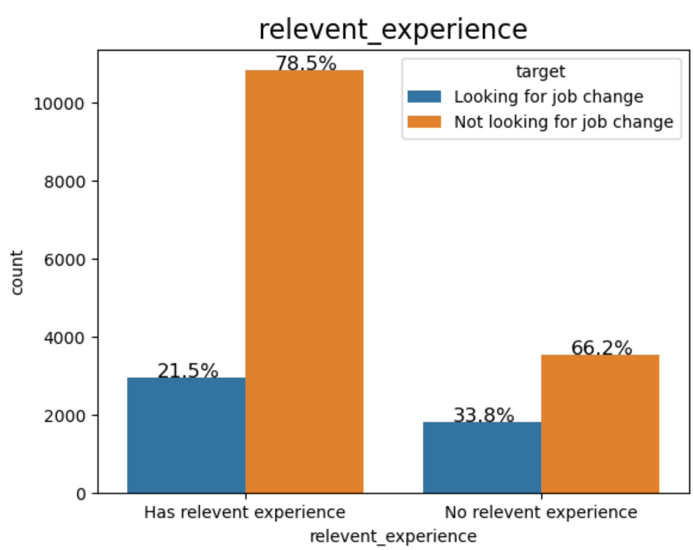


Distribution of job change by company_size

Company type and size also matters. People working in the Pvt Ltd, NGO and Founded Startup are less likely to look for a new job(Figure.8).People working in the company with size

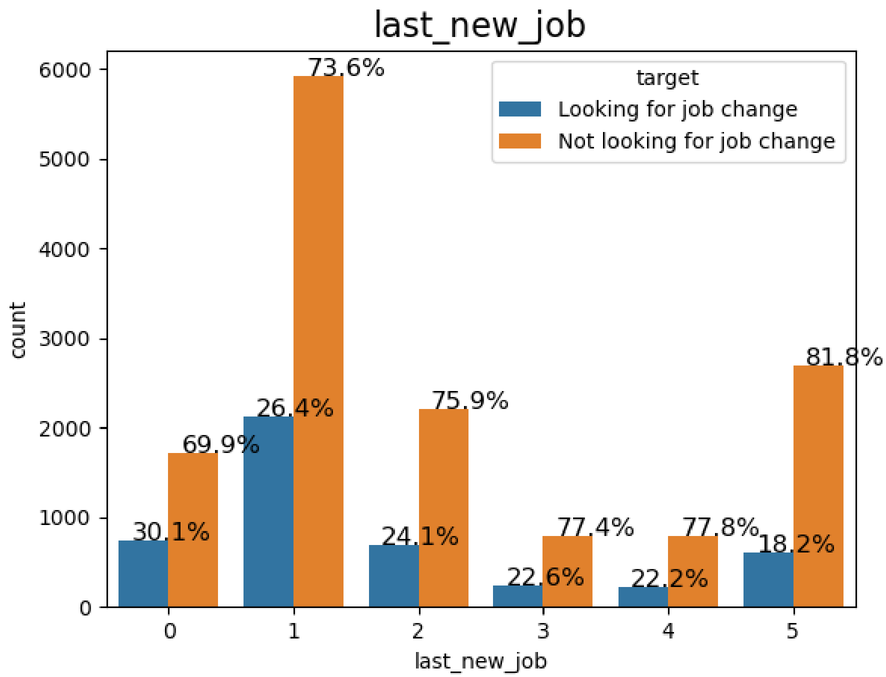


of 10-49 are more inclined to look for a new job.



Distribution of job change by relevent_experience

Working experiences is an important factor affecting the rate of people looking for a new job. People with less working experiences are more likely to look for a new job. The rate of looking for a new job for people with no relevant experience is a little higher.



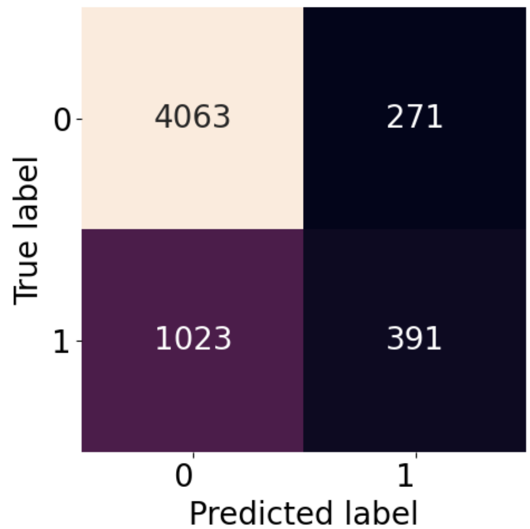
Distribution of job change by last_new_job

The difference of 1 year and zero year shows a significant higher rate of looking for a new job, which indicates that people looking for job change are used to working in different company for same time long.

5. Results

5.1. Logistic Regression Result

From the confusion matrix, we could find that there are 4063 true negatives, and 391 true positives, the result is shown as below:



From the classification report graph below, we can see the accuracy of logistic regression model is 77.48%. It means, for the given test data set, the ratio of the number of samples correctly classified by the classifier to the total number of samples is 77.48%. It is calculated by (TP+TN)/(TP+FP+FN+TN). Here, TP represents the true positive, and the FP represents the false positive, and the FN represents the false negative, and TN means the true negative. We could get those result from confusion matrix above.

```
              precision    recall  f1-score   support

           0       0.80      0.94      0.86      4334
           1       0.59      0.28      0.38      1414

    accuracy                           0.77      5748
   macro avg       0.69      0.61      0.62      5748
weighted avg       0.75      0.77      0.74      5748

Accuracy :  77.48782185107864
ROC_AUC :  78.41463080318185
```

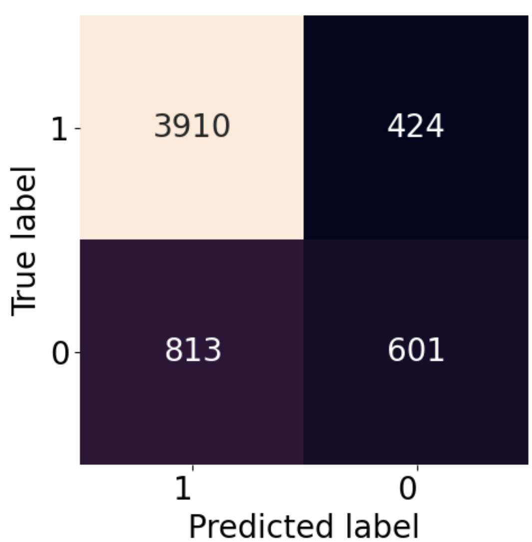
The precision here means the ratio of true positive to the sum of true positive and false positive. And the recall means the ratio of true positive to the sum of true positive and false negative. F-1 score is a combination index, it means the harmonic mean of precision and recall. The equation is like:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

The AUC means the area under the ROC curve. It shows the tradeoff of specificity and sensitivity. And roc_auc sore helps us find the model performances. Normally, the roc_auc score higher than 0.5 would see as a good classifier. I would combine their different index in our summary to interpret the result. All the ROC result is shown is appendix.

5.2.1 Random Forest

From the random forest result, we could see that there are 36910 true positives and 601 true negatives. The result is shown as below.

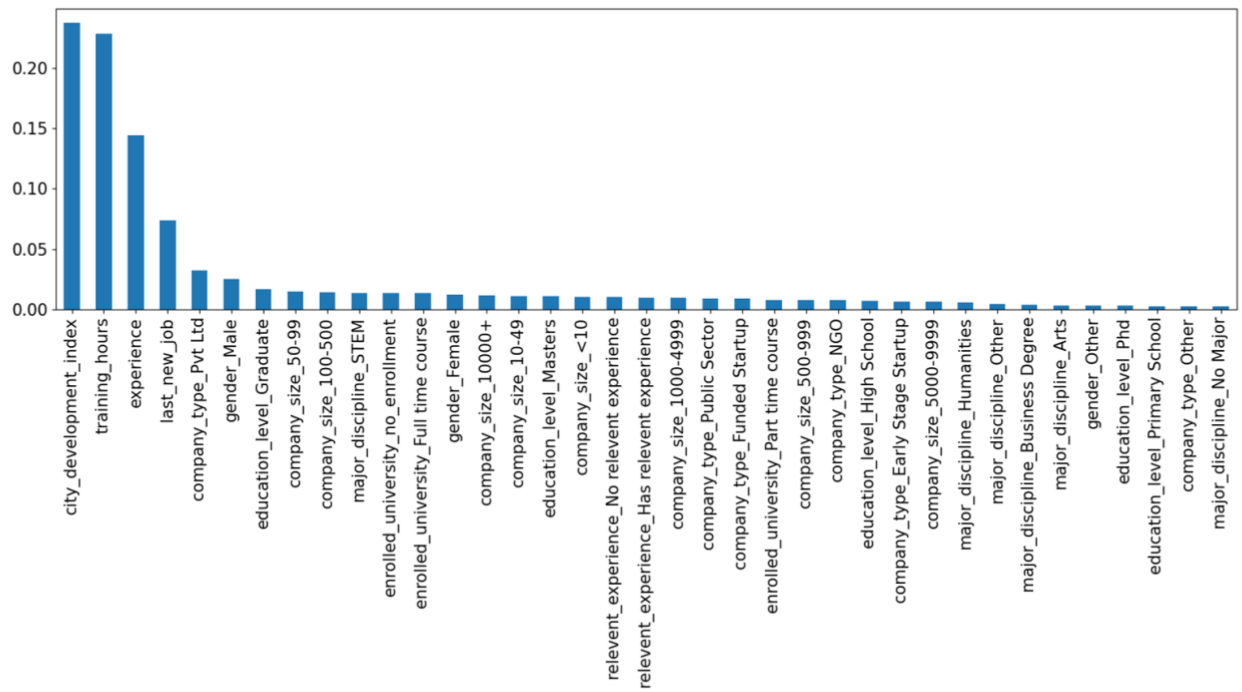


The accuracy for random forest model is 78.47%%, which means that the fitting model could explain 78.47% of the test dataset.

```
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      4334
           1       0.59      0.43      0.49      1414

    accuracy                           0.78      5748
   macro avg       0.71      0.66      0.68      5748
weighted avg       0.77      0.78      0.77      5748

Accuracy: 78.4794711203897
ROC_AUC: 78.6315678340858
```

5.2.2 Feature Importance

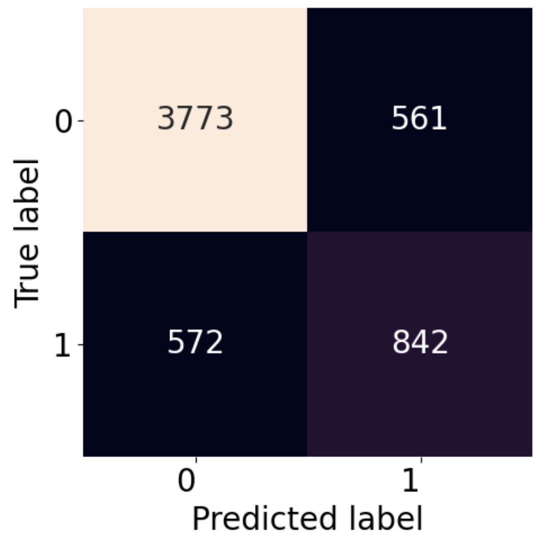
We use the random forest to calculate feature importance, and below shows the result.



From the graph, we could find that the top 5 important features are city development index, training hours, experience, last_new_job and company_type.

5.3. Gradient Boosting Result

From the gradient boosting result, we could see that there are 3663 true negatives and 842 true positives. The result is shown as below.



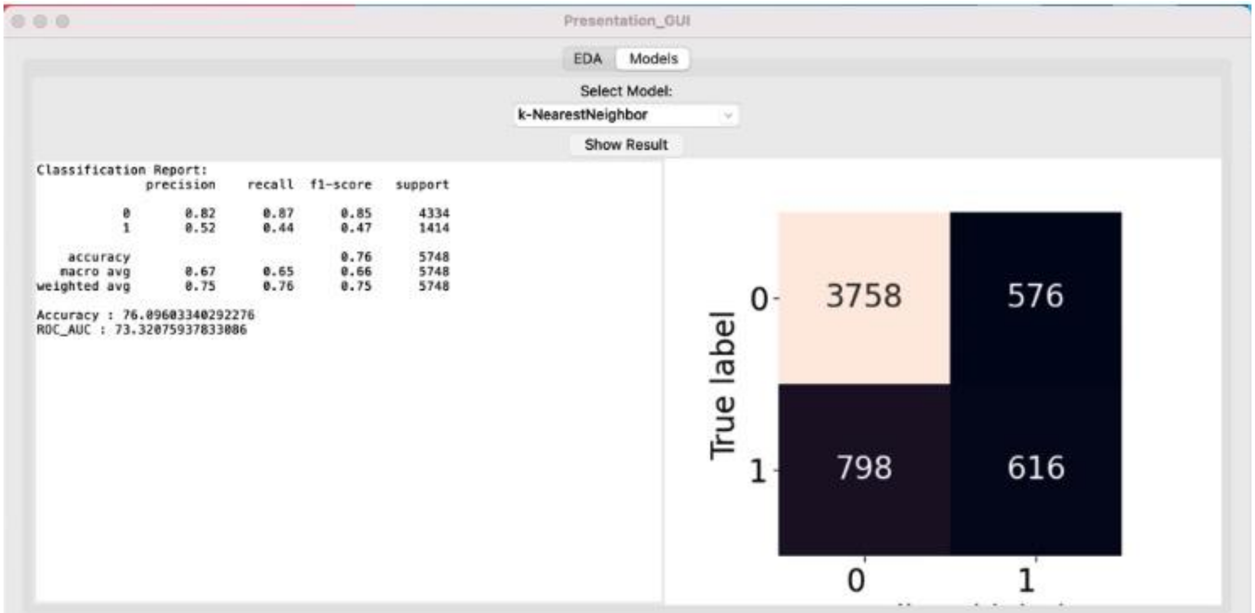
The accuracy for gradient boosting model is 80.28%, which means that the fitting model could explain 80.28% of the test dataset.

	precision	recall	f1-score	support
0	0.87	0.87	0.87	4334
1	0.60	0.60	0.60	1414
accuracy			0.80	5748
macro avg	0.73	0.73	0.73	5748
weighted avg	0.80	0.80	0.80	5748

```
Accuracy: 80.28879610299235
ROC_AUC :  80.1568010318073
```

5.4. GUI

Finally, I tried to implement GUI. We divided it into two parts, EDA part and Model part. I built up the model part. By clicking the different model options, we could directly get the result of our classification model.



6. Summary and conclusions

Based on our results and the index we have discussed before; we could define that gradient boosting is the best model. So, we choose it to do prediction. This model has 80.28% accuracy and ROC_AUC score is 0.80, which means it could explain the 80.28% of the test data. And it has the highest ROC_AUC score.

Besides the models we talked about, I also conduct several other classifiers, like XGB boosting, KNN, etc.… But none of them has the better result than the models we selected.

The improvement of our project may mainly implement in GUI part and data preprocess. Maybe trying another way to deal with our data would have a better result.

7. References

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, 21.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 3146-3154.

Menard, S. (2002). Applied logistic regression analysis (Vol. 106). Sage

https://christophm.github.io/interpretable-ml-book/logistic.html

https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists?select=aug_train.csv

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

https://www.javatpoint.com/classification-algorithm-in-machine-learning

https://en.wikipedia.org/wiki/Gradient_boosting

Appendix

1. Data head

2. Data information

3. Null values

4. ROC_AUC for Logistic

5. ROC_AUC for Random Forest

6.ROC_AUC for HGradient Boosting

```
Dataset first few rows:

   enrollee_id      city  ...  training_hours target
0         8949  city_103  ...              36    1.0
1        29725   city_40  ...              47    0.0
2        11561   city_21  ...              83    0.0
3        33241  city_115  ...              52    1.0
4          666  city_162  ...               8    0.0

[5 rows x 14 columns]
Dataset info:
```

Figure 1. data head

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19158 entries, 0 to 19157
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   enrollee_id             19158 non-null  int64
 1   city                    19158 non-null  object
 2   city_development_index  19158 non-null  float64
 3   gender                  14650 non-null  object
 4   relevent_experience     19158 non-null  object
 5   enrolled_university     18772 non-null  object
 6   education_level         18698 non-null  object
 7   major_discipline        16345 non-null  object
 8   experience              19093 non-null  object
 9   company_size            13220 non-null  object
 10  company_type            13018 non-null  object
 11  last_new_job            18735 non-null  object
 12  training_hours          19158 non-null  int64
 13  target                  19158 non-null  float64
dtypes: float64(2), int64(2), object(10)
memory usage: 2.0+ MB
```

Figure 2. data information

```
Sum of NULL values in each column.
city_development_index       0
gender                    4508
relevent_experience          0
enrolled_university        386
education_level            460
major_discipline          2813
experience                  65
company_size              5938
company_type              6140
last_new_job               423
training_hours               0
target                       0
dtype: int64
```

Figure 3. Null values

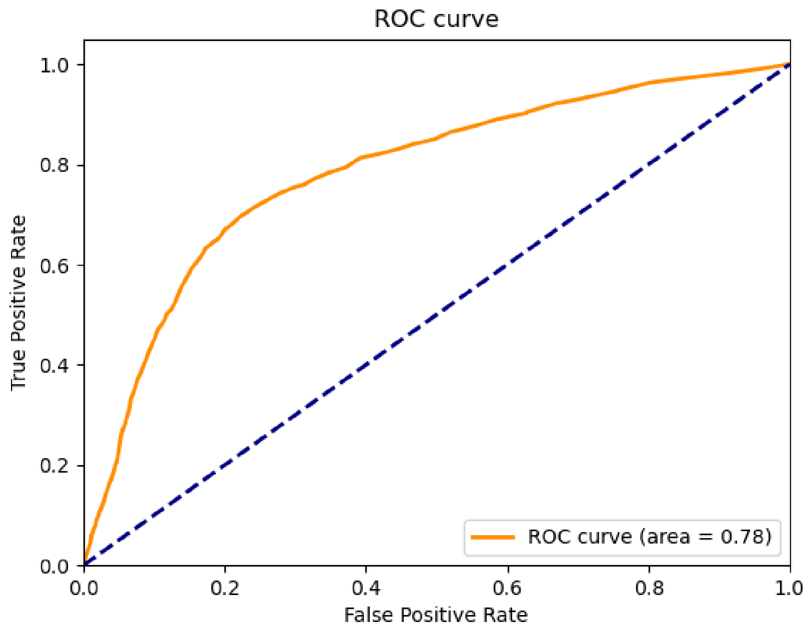


Figure 4.ROC_AUC for Logistic

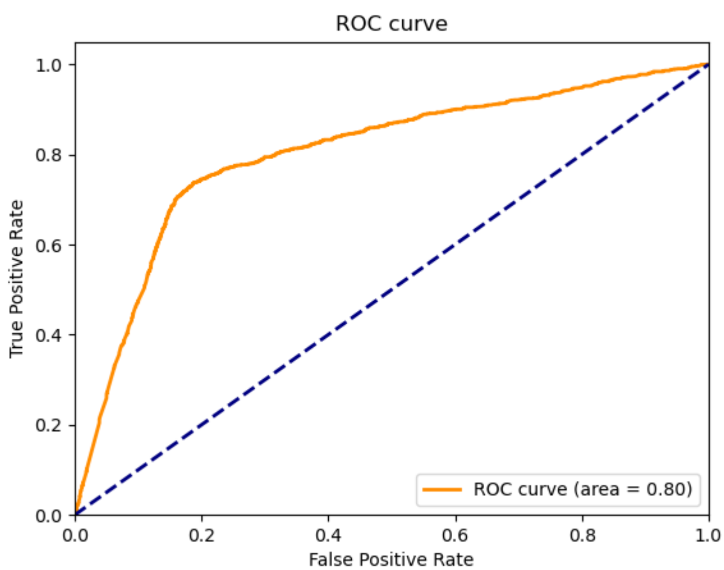


Figure 5.ROC_AUC for Random Forest