**HR Analytics: Job Change of Data Scientists: A Statistical Study**

Document By:

Srishti Gupta

Universita Degli Studi Di Milano

Department of Economics, Management and Quantitative Methods

**Table of Contents**

***Chapter-1 Introduction ………………………………………………………………………………………………………………*3**

***Chapter -2 Data Description……………………………………………………………………………………………………….4***

**2.1 Data Description ……………………………………………………………………………………………………………… 4**

**2.2 Data Preprocessing………………………………………………………………………………………………………….10**

***Chapter-3 Statistical Analysis……………………………………………………………………………………………………13***

**3.1 Logistic Regression …………………………………………………………………………………………………………..13**

**3.2 Linear Discriminant Analysis……………………………………………………………………………………………..14**

**3.3 Quadratic Discriminant Analysis……………………………………………………………………………………….15**

**3.4 K-Nearest Neighbor…………………………………………………………………………………………………………..19**

**3.5 Bagging……………………………………………………………………………………………………………………………..21**

**3.6 Random Forest………………………………………………………………………………………………………………….22**

***Chapter-4 Limitations & Conclusions………………………………………………………………………………………..25***

**4.1 Limitations………………………………………………………………………………………………………………………..25**

**4.2 Conclusions………………………………………………………………………………………………………………………25**

***Appendix***

**Chapter 1**

**INTRODUCTION**

In this chapter you will find Introduction and objective of this study.

**1.1 Introduction**

This dataset is from a company which is active in Data Science wants to hire data scientists among people who successfully pass some courses which is conducted by the company. Many people signup for their training. These training courses were conducted by the Company itself. Company wants to know which of these candidates are really wants to work for the company after training or looking for a new employment because **it helps to reduce the cost and time as well as the quality of training or planning the courses and categorization of candidates**. Dataset contains the information related to demographics, education, experience is in hands from candidate’s signup and enrollment.

This dataset designed to understand the factors that lead a person to leave current job for HR research too. By model(s) that uses the current credentials, demographics, experience data you will **predict the probability of a candidate to look for a new job or will work for the company, as well as interpreting affected factors on employee decision.**

The aim of doing this is to help reduce training costs and time, specifically as it relates to the quality of training, the planning of courses, and the categorization of candidates. Evaluation of this data set can also help provide the Human Resource department with essential information regarding the factors that lead candidates to leave current job positions.

Objective

The objective of this study is to:

* **predict that whether a candidate will look for a new job or will work for the company,**
* **Interpreting affected factors on employee decision.**
* **Suggest company how to improve their candidate selection strategies (i.e whom they should recruit based on certain characteristic etc.)**

**Chapter – 2**

**Data Description**

In this section, we will present main description of the data set. The data is collected from Kaggle.

**2.1 Data Description**

The entire data set is composed of two data files: (1) a train data set with 19,158 observations and 14 variables (2) a test data set with 2,129 observations and 13 variables. Since the outcome variable is only in the train data, this analysis utilizes the train data set as the “full” data set and the test data is disregarded. Thus, the full data for this analysis has a sample size of 19,158 observations and 14 variables. The predictors for this data deal with important information regarding candidate credentials, demographics, and work and education experience. Ten predictors are categorical, two are continuous numeric, there is an “ID” variable, and the outcome variable is binary and indicates if a candidate is looking for a job change or not.

The Summary of the data is as follows in the Fig1. below:

A picture containing table

Description automatically generated

**Fig1. Summary**

**2.1.1 Independent Variables**

In the dataset we have 13 independent variables. The details of the variables with their pictorial representation are as follows:

1. Enrollee ID: Unique candidate ID number
2. City: City code identification (123 unique cities)
3. City Development Index: Scaled development index which is a measure of the development level of the city (values range from 0.448 to 0.949)

Chart, histogram

Description automatically generated

**Fig2. City Development Index**

1. Gender: Candidate gender (Male, Female, or Other)

Chart, pie chart

Description automatically generated

**Fig3. Gender**

1. Relevant Experience: Indicates if a candidate has relevant work experience or not (0 = Candidate has no relevant work experience, 1 = Candidate has relevant work experience)

Chart, bar chart

Description automatically generated

**Fig4. Relevant Experience**

1. University Enrollment: Type of University course enrollment (Full time, Part time, or No Enrollment)

Chart

Description automatically generated

**Fig5. University Enrollment**

1. Education Level: Candidate education level (Graduate, High School, Masters, PhD, or Primary School)

Chart, radar chart

Description automatically generated

**Fig6. Education Level**

1. Major Discipline: Candidate education major (STEM, Business, Humanities, Arts, Other, or No Major)

Chart

Description automatically generated

**Fig7. Major Discipline**

1. Experience: Candidate total work experience in years (categorized in groups between less than 1 and greater than 20)

Chart, histogram

Description automatically generated

**Fig8. Experience**

1. Company Size: Number of employees in current employer's company (categorized in groups between less than 10 and ≥ 10,000 employees)

Chart, bar chart

Description automatically generated

**Fig9. Company Size**

1. Company Type: Current employer's company type (Early-Stage Startup, Funded Startup, NGO, Public Sector, Private Ltd, or Other)

Chart, histogram

Description automatically generated

**Fig10. Company Type**

1. Last New Job: Difference between previous job and current job in years (categorized in groups between 1 and greater than 4 years, or never)

Chart, bar chart

Description automatically generated

**Fig11. Last New Job**

1. Training Hours: Number of completed training hours (values range from 1 to 336 hours)

Chart, bar chart, histogram

Description automatically generated

**Fig12. Training Hours**

* + 1. Dependent Variable
  1. Target: Indicates if candidate is looking for a job change or not after having taken training courses (0 = Not looking for job change, 1 = Looking for job change)

Chart, pie chart

Description automatically generated

**Fig13. Target Variable (0 = Not looking for Job Change, 1= Looking for Job Change)**

* 1. **Data Pre- processing**

**2.2.1 Variables Dropped**

The variables which are dropped are:

Enrollment Id, Candidate City Code, Major Discipline.

Since Enrollment Id is a unique identification number for the candidates who are pursuing Training, hence it will not be required in the analysis. Candidate City Code is also an identifier which is not related to target variable therefore it is also dropped. The variable Major Discipline is also dropped because it is near zero variance as major candidates are from STEM.

**2.2.2 Rescaled Variables**

The variable City Development Index was rescaled to represent the value out of 100. This is done just for interpretability.

**2.2.3 Variables with Missing Values**

Missing values for candidate gender were incorporated in the “Other” category and candidates with missing values for university enrollment were considered to not be enrolled. Mode imputation method is utilized on 5 other variables to deal with missing values. The below Table 1 demonstrates variables with corresponding missing value amounts.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable** | Gender | University Enrollment | Education Level | Experience | Company Size | Company Type | Last New Job |
| **NA** | 4,508 | 386 | 460 | 65 | 5,938 | 6,140 | 423 |

**Table 1: Details of Missing Values**

**2.2.4 Re-leveled Variables**

There were a few variables that had many different categories, or levels, and needed to be condensed as well. For example, candidate major and company type had 6 different groups, candidate major and education level had 5, and candidate experience had over 20 different groups. A total of five variables were re-leveled, and the original and new levels for these variables can be seen in below Table2. Under the “New Levels” column, whatever’s underlined represents the new level name for a given variable, and whatever’s in parenthesis is the original level name.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Original Levels** | **New Levels** |
| Education Level | Graduate, High School, Masters, PhD, Primary School | University (Graduate, Masters, PhD), Other (High School, Primary School) |
| Experience | <1, 1-20 (1, 2, etc.), >20 | Some Experience (<1,1-4), Experience (5-14), A lot of Experience (15-20, >20) |
| Company Size | <10, 10-49, 50-99, 100-500, 500-999, 1000-4999, 5000-9999, 10000+ | Small (<10, 10-49, 50-99, 100-500), Medium (500-999, 1000-4999), Large (5000-9999, 10000+) |
| Company Type | Early-Stage Startup, Funded Startup, NGO, Public Sector, Pvt Ltd, Other | Public (Public Sector), Startup (Early-Stage Startup, Funded Startup), Other (NGO, Pvt Ltd, Other) |
| Last New Job | Never, 1-4, >4 | Never (Never), Recent (1, 2), Kind of Recent (3, 4), Not Recent (>4) |

**Table2: Details of Re-leveled Variables**

**2.3 Correlation Plot**

After re-leveling of some features, imputing the missing values using mode, and dropping some variables the Correlation Plot is computed to understand the relationship between the features. The correlation plot can be seen below in Fig14.

Chart

Description automatically generated

**Fig14. Correlation Plot**

The above correlation plot depicts that the Candidates who will not look for Job change only if the city development index of their city is less and doesn’t not have lot of experience.

**Chapter – 3**

**Statistical Analysis**

In this chapter you will find description of the methodology and models used.

Various models have been used to predict whether candidate will look for Job change or not. In the end we will compare the results of these models to select which model has performed better.

**3.1 Models**

**3.1.1 Logistic Regression**

As we are interested in predicting that whether a candidate will look for Job Change or not. First model used is Logistic Regression which provides the probability that Y belongs to category.

**Data Splitting**

The data set had a sample size of 19,158 observations and 11 variables (10 predictors and the outcome/dependent variable). To perform the analysis the data set is split into 70% train and 30% test.

The output of the Logistic Regression is as follows:

A screenshot of a computer

Description automatically generated with medium confidence

**Table3. Coefficients of Features: Logistic Regression**

From above Table we can clearly see that Features which are important for prediction are City Development Index, Relevant Experience, Enrolled University, Experience, Company Type, Last New Job. It states that the feature which is not useful for the model is company size.

**Confusion Matrix (Training Data)**

Confusion Matrix of Logistic Regression on Training Data is shown below. The Misclassification error on Training set is 23.2%

A picture containing text

Description automatically generated

**Table4. Confusion Matrix (Training Dataset) : Logistic Regression**

**Confusion Matrix (Test Data)**

Confusion Matrix of Logistic Regression on Test Data is shown below. The Misclassification Error on Test set is 22.9%.

A picture containing chart

Description automatically generated

**Table5. Confusion Matrix (Test Dataset): Logistic Regression**

The model is predicting correctly that 3986 candidates will not look for Job change and 368 candidates will look for job change. However, it is predicting incorrectly that 281 candidates will look for job change, and 1014 candidates will not look for Job change. Accuracy of the model is 1-0.23 = 77%. The Model is able to classify 77% of the observations correctly.

**3.1.2 Linear Discriminant Analysis**

The output of the LDA Model is shown in the Table below.

A screenshot of a computer

Description automatically generated with medium confidence

**Table6. Coefficients of Features: Linear Discriminant Analysis**

**Confusion Matrix (Training Data)**

The Confusion Matrix of LDA is shown below in the Table. The Model has predicted correctly that 9284 candidates will not look for Job change and 1157 candidates will look for Job change. However, it has wrongly predicted 2238 candidates will not look for Job Change and 830 candidates will look for Job change. The misclassification error on Training data is 23% which is like Logistic Regression.

A picture containing text

Description automatically generated

**Table7. Confusion Matrix: Training Data (Linear Discriminant Analysis)**

**Confusion Matrix (Test Data)**

The confusion Matrix computed for Test Data is shown below in the Table. The Model has predicted correctly that 3895 candidates will not look for Job Change and 473 will look for change. However, it has wrongly predicted that 909 candidates will not look for Job change and 372 candidates will look for Change. The Accuracy of the Model is 77.3%

Text

Description automatically generated with low confidence

**Table8. Confusion Matrix: Test Data (Linear Discriminant Analysis)**

**3.1.3 Quadratic Discriminant Analysis (QDA)**

**Confusion Matrix (Training Data)**

The Confusion Matrix of QDA is computed on Training Data and it is shown below in the table. The Model has predicted correctly that 8383 candidates will not look for Job change and 697 will look for Jo change. However, it has predicted wrongly that 1731 candidates will look for Job change and 1617 candidates will not look for job change. The Misclassification error on Training Data is 1 - 0.75 = 25 %

A picture containing text

Description automatically generated

**Table9. Confusion Matrix (Training Data): QDA**

**Confusion Matrix (Test Data)**

The Confusion Matrix of QDA is computed on Test Dataset to check the Model performance and it is shown in the table below. The Model is predicting correctly that 3492 candidates will not look for Job change and 697 candidates will look for Job change. The Accuracy of the Model is 74%

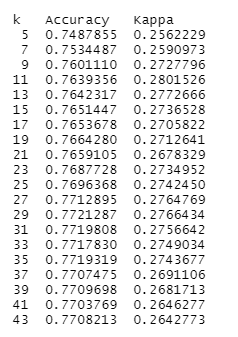
Text, letter

Description automatically generated

**Table10. Confusion Matrix (Test Data): QDA**

**3.1.6 K-Nearest Neighbor**

The K-Nearest Neighbor is computed using 10 folds Cross Validation and repeated 3 times. The accuracy for various values of K is shown in the Table below. The highest accuracy is 77.21% when K = 29. Therefore, K= 29 is the optimal selection for the model.



**Table13: Optimal K Selection**

The plot of the optimal K selection is demonstrated in the below figure. It is clearly visible that the accuracy is highest when K= 29.

Chart, line chart

Description automatically generated

**Fig22. Plot – K Nearest Neighbors**

The Table below demonstrate the variables important for the prediction. As suggested by the model Training Hours is the only variable which is not relevant for the prediction whether candidate will look for Job Change or not.

Text

Description automatically generated with low confidence

**Table14: Features Importance: K Nearest Neighbors**

The confusion Matrix of K Nearest Neighbor on Test dataset is demonstrated below in the Table. The KNN model has predicted correctly that 3945 candidates will not look for Job Change while 419 candidates will look for Job change. However, it has predicted wrongly that 963 candidates will not look for Job change and 322 will look for change. The misclassification error of the Model is 22%.

Text

Description automatically generated with medium confidence

**Table15: Confusion Matrix(Train Dataset): K Nearest Neighbors**

The Performance Table of K Nearest Neighbors is demonstrated below. The accuracy of the Model is 77.25%

Text

Description automatically generated

**Table16: Accuracy & Confidence Interval: K Nearest Neighbors**

**3.1.7 Bagging**

The confusion Matrix of Bagging is demonstrated in the table below. The model has predicted correctly that 3686 candidates will not look for Job Change while 533 candidates will look for Job change. However, it has misclassified 849 candidates will not looking for Job change and 581 candidates will look for Job Change. The misclassification error is 1 -0.74 = 26%.

A picture containing text

Description automatically generated

**Table17: Confusion Matrix (Test Data): Bagging**

The accuracy of the Model is shown in table below. Model Accuracy is 74% and using Binomial distribution we can compute 95% Confidence Interval which indicates that the accuracy value likely to lie between 73 – 75%. Then we have No Information Rate value it is the largest proportion of the observed class. In the dataset the largest proportion is for class 0 therefore the larger class is (3686+581)/5649 = 77% which means if no model is developed and simply classify every candidate not looking for change which means belongs to Category 0 it will be right 75% of the time. The Accuracy of the model is lesser than the No-Information Rate obviously it is not a good model. If we are interested in predicting class 1 rather than class 0 than this model doesn’t perform very well. Sensitivity tells us that how often we can correctly predict the Class 0 which is 86% and specificity tells us that how often we are able to correctly predict Class 1 which is 38%.

Sensitivity is very high, and Specificity is very low. One of the reasons for the wide difference between the Sensitivity and Specificity is that there are few data points for Class 1 but much more for class 0. Therefore, this model is dominated by Class 0 and indeed it is performing good for predicting Class 0 but not good so job for predicting Class 1.

A screenshot of a computer

Description automatically generated with low confidence

**Table18: Accuracy: Bagging**

The Bagging variable importance is demonstrated in the fig. below. Bagging Model has suggested the variables important for the prediction are same as suggested by Ridge, LASSO and Logistic Regression.

Table

Description automatically generated

**Fig23: Plot Variable Importance: Bagging**

**3.1.8 Random Forest**

The Confusion Matrix of Random Forest is demonstrated in the table below. The Model has predicted correctly that 3823 candidates will not look for Job change while 575 candidates will look for Job change. However, it has misclassified 807 candidates to Class 0 i.e Not looking for Job Change and 444 candidates to Class 1 i.e Looking for Job Change. The Misclassification error is 1-0.78 = 22%

A picture containing text

Description automatically generated

**Table19: Confusion Matrix(Test Data): Random Forest**

The accuracy of the Model is shown in table below. Model Accuracy is 77.8% and using Binomial distribution we can compute 95% Confidence Interval which indicates that the accuracy value likely to lie between 76– 78%. Then we have No Information Rate value it is the largest proportion of the observed class. In the dataset the largest proportion is for class 0 therefore the larger class is 75% which means if no model is developed and simply classify every candidate not looking for change which means belongs to Category 0 it will be right 75% of the time. The Accuracy of the model is better than the No-Information Rate obviously it is a good model. If we are interested in predicting class 1 rather than class 0 than this model doesn’t perform very well. Sensitivity tells us that how often we can correctly predict the Class 0 which is 89% and specificity tells us that how often we are able to correctly predict Class 1 which is 41%.

Sensitivity is very high, and Specificity is very low. One of the reasons for the wide difference between the Sensitivity and Specificity is that there are few data points for Class 1 but much more for class 0. Therefore, this model is dominated by Class 0 and indeed it is performing good for predicting Class 0 but not good so job for predicting Class 1.

A picture containing text, receipt

Description automatically generated

**Table20: Performance of Model: Random Forest**

**3.2 Algorithms and Percentage of Accuracy**

|  |  |
| --- | --- |
| **Algorithms** | **Performance** |
| Logistic Regression | 77% |
| Linear Discriminant Analysis | 77.30% |
| Quadratic Discriminant Analysis | 74% |
| K-Nearest Neighbors | 77.25% |
| Ridge Regression | 76.80% |
| LASSO | 77% |
| Bagging | 74.70% |
| Random Forest | 77.80% |

The Random Forest Model has performed best among all other models for the dataset.

**Chapter – 4**

**Limitations & Conclusions**

**4.1 Limitations**

The dependent variable Target is a categorical variable. It has two classes 0 and 1. 0 refers to Candidate not looking for Job change and 1 refers to candidate looking for Job change. There are few data points /observations available in the dataset for class 1 and due to this there is an imbalance in the dataset. The data imbalance issue could be dealt by performing oversampling or undersampling of the data by using ROSE library in R. For future research it would be great to compare the results of the Models after using oversampling or undersampling technique.

Another Limitation was

**4.2 Conclusions**

Random Forest performed the best between the other models for this analysis, and it found city development index, training hours, and not being enrolled in a university to be the most significant, or “important,” factors for predicting candidate odds of looking for a job change. City development index, working for a large or “other” company, and not being enrolled in a university were found to be significant factors for predictions as well. Candidates that are from cities with higher city development indexes, aren’t enrolled in a university, or completed more training hours are more likely to, on average, not be looking for a job change. Meanwhile, the data also shows that candidates that work for large companies or have no relevant work experience in data science are more likely to look for a job change, on average.

A suggestion for the Talent Acquisition Team of the organization is to search for the bright minds which have lower odds of Looking for a Job change.

From the analysis we find that candidates from cities with higher city development indexes, aren’t enrolled in a university, and/or work for smaller to midsize companies have less chances that they will look for Job change. The organization could filter their search criteria and may avoid looking for candidates from cities with low city development index, work for large startup or public companies and/ or don’t have work experience.

**Appendix**

**R-code**

#Load packages

library(dplyr)

library(tidyr)

library(plyr)

library(ggplot2)

library(lme4)

library(AICcmodavg)

library(magrittr)

library(scales)

library(gmodels)

library(agricolae)

library(multcomp)

library(Sleuth2)

library(MASS)

library(car)

library(glmnet)

library(caret)

library(leaps)

library(bestglm)

library(VIM)

library("VIM")

library(forcats)

library(stringr)

library(formattable)

library(AppliedPredictiveModeling)

library(klaR)

library(randomForest)

library(pROC)

library(ISLR)

library(naniar)

library(factoextra)

library(lsmeans)

library(tidyverse)

library(Sleuth3)

library(tree)

library(ranger)

library(partykit)

library(parallel)

library(doParallel)

library(kernlab)

library(MLmetrics)

library(rminer)

library(foreach)

#Load dataset( we are loading train dataset because in Test dataset Target variable is not present)

#Train data

aug = read.csv("aug\_train.csv", header=T, na.strings=c(""," ","NA"))

aug = data.frame(aug)

head(aug)

str(aug)#19158 obs & 14 variables

summary(aug)

## plot city development index ##

ggplot(data = aug) + geom\_histogram(mapping = aes(x = city\_development\_index), fill = "blue", col = "black", bins=20 )

## plot of gender ##

mytable = table(aug$gender)

lbls = paste(names(mytable), "\n", mytable, sep="")

pie(mytable, labels = lbls,main="Gender")

## plot of Relevant Experience ##

ggplot(data = aug) + geom\_bar(mapping = aes(x = relevent\_experience), fill = "Blue", col = "black")

##plot of Enrolled\_University

mytable <- table(aug$enrolled\_university)

lbls <- paste(names(mytable), "\n", mytable, sep="")

pie(mytable, labels = lbls, main="enrolled university")

## plot of education level

ggplot(data = aug) + geom\_bar(mapping = aes(x = education\_level ), fill = "blue", col = "black")

table\_edulevel <- table(aug$education\_level)

level\_edu <- paste(names(table\_edulevel), "\n", table\_edulevel, sep="")

pie(table\_edulevel, labels = level\_edu, main="Education Level")

##plot of Major Discipline

table\_mjdscpl <- table(aug$major\_discipline)

level\_mjdscpl <- paste(names(table\_mjdscpl), "\n", table\_mjdscpl, sep="")

pie(table\_mjdscpl, labels = level\_mjdscpl, main="Major Discpline")

ggplot(data = aug) + geom\_bar(mapping = aes(x = major\_discipline ), fill = "blue", col = "black", width = 0.6)

## plot of experience

ggplot(data = aug) + geom\_bar(mapping = aes(x = experience ), fill = "blue", col = "black", width = 0.6)

## plot of company size

ggplot(data = aug) + geom\_bar(mapping = aes(x = company\_size ), fill = "blue", col = "black", width = 0.6)

## plot of company type

ggplot(data = aug) + geom\_bar(mapping = aes(x = company\_type ), fill = "blue", col = "black", width = 0.6)

## plot of last new job

ggplot(data = aug) + geom\_bar(mapping = aes(x = last\_new\_job ), fill = "blue", col = "black", width = 0.6)

## training hours

ggplot(data = aug) + geom\_bar(mapping = aes(x = training\_hours ), fill = "blue", col = "black", width = 0.6)

# plot of TARGET

tabletarget <- table(aug$target)

trgt <- paste(names(tabletarget), "\n", tabletarget, sep="")

pie(tabletarget, labels = trgt,main="TARGET")

### plot of Target Variable(0- Not looking for change, 1- Looking for Job Change)

library(ggplot2)

ggplot(data = aug, aes(target)) +

geom\_histogram(aes(y = ..density..), fill = 'orange') +

geom\_density()

#Explore the City Variable

aug$city = as.factor(aug$city)

aug %>% count('city') #categorical w/123 unique levels (or cities)

#Explore 'city\_development\_index' variable

range(aug$city\_development\_index) #values range from 0.448 to 0.949

#Explore 'Gender' feature

aug$gender = as.factor(aug$gender)

levels(aug$gender)

#Explore 'relevant\_experience' variable

aug$relevent\_experience = as.factor(aug$relevent\_experience)

levels(aug$relevent\_experience) #relevant work experience or not; 0 = no experience / 1 = experience

#Inspect 'enrolled\_university' variable

aug$enrolled\_university = as.factor(aug$enrolled\_university)

levels(aug$enrolled\_university) #full time, part time, or no enrollment

#Inspect 'education\_level' variable

aug$education\_level = as.factor(aug$education\_level)

table(aug$education\_level) #Graduate, High School, Masters, PhD, or Primary School

#Inspect 'major\_discipline' variable

aug$major\_discipline = as.factor(aug$major\_discipline)

aug %>% count('major\_discipline') #STEM, Business, Humanities, Arts, Other, or No Major

#Inspect 'experience' variable

aug$experience = as.factor(aug$experience)

table(aug$experience) #grouped between less than 1 and greater than 20

#Inspect 'company\_size' variable

aug$company\_size = as.factor(aug$company\_size)

aug %>% count('company\_size') #grouped between less than 10 and 5000-9999 employees

#Inspect 'company\_type' variable

aug$company\_type = as.factor(aug$company\_type)

table(aug$company\_type) #Early-Stage Startup, Funded Startup, NGO, Public Sector, Private Ltd, or Other

#Inspect 'last\_new\_job' variable

aug$last\_new\_job = as.factor(aug$last\_new\_job)

table(aug$last\_new\_job) #grouped between 1 and greater than 4 years, as well as never

#Inspect 'training\_hours' variable

range(aug$training\_hours) #values range from 1 to 336 hours

#Inspect 'target' variable (target variable)

aug$target = as.factor(aug$target)

levels(aug$target) #0 = Not looking for a job change, 1 = Looking for a job change (predicting '1')

#Drop 'enrollee\_id' and 'city' variables

aug = aug[,-c(1,2)]

#check the data

str(aug)

#Near-zero variance variables

near\_zero = nearZeroVar(aug)

near\_zero

#The 'major\_discipline' variable has near-zero variance- remove it

aug = aug[,-6]

str(aug) #19158 obs & 11 variables

#Check to see if any class imbalances exist- there are some class imbalances but we'll proceed w/caution

table(aug$gender)

table(aug$relevent\_experience)

table(aug$enrolled\_university)

#Barplots to show variables w/class imbalances

par(mfrow=c(2,2))

#Gender

barplot(table(aug$gender),main="Barplot of Gender",

xlab="Gender Type",

ylab="Count",

col="black")

#Relevant Experience

barplot(table(aug$relevent\_experience),main="Barplot of Relevant Experience",

xlab="Relevant Experience",

ylab="Count",

col="black")

#Education Level

barplot(table(aug$education\_level),main="Barplot of Education Level",

xlab="Education Level",

ylab="Count",

col="black")

#University Enrollment

barplot(table(aug$enrolled\_university),main="Barplot of University Enrollment",

xlab="Enrollment Type",

ylab="Count",

names.arg=c("Full time course"="Full time", "no\_enrollment"="No enrollment", "Part time course"="Part time"),

col="black")

#Change graph fit back

par(mfrow=c(1,1))

#See if variables have NAs

sapply(aug, function(x) sum(is.na(x)))

#Change NAs to 'Other' for 'gender' variable

aug$gender[is.na(aug$gender)] <- "Other"

table(aug$gender)

#Change NAs to 'no\_enrollment' for 'enrolled\_university' variable

aug$enrolled\_university[is.na(aug$enrolled\_university)] <- "no\_enrollment"

table(aug$enrolled\_university)

#Use mode imputation for 'education\_level', 'experience', 'company\_size', 'company\_type', &

#'last\_new\_job' variables

c1<-makePSOCKcluster(3)

registerDoParallel(c1)

aug = VIM::kNN(aug, k = 10, numFun = mode)

aug = aug[,-c(12:22)]

str(aug)

sapply(aug, function(x) sum(is.na(x))) #no more NAs

#Change levels for certain variables

#Change 'education\_level' levels

aug$education\_level = ifelse(aug$education\_level == "Graduate" | aug$education\_level == "Masters" | aug$education\_level == "Phd", 1, 0)

aug$education\_level[aug$education\_level == 1] <- "University"

aug$education\_level[aug$education\_level == 0] <- "Other"

aug$education\_level = as.factor(aug$education\_level)

levels(aug$education\_level)

table(aug$education\_level)

#count: 1/uni = 16668, 0/other = 2490

#Change 'experience' levels

levels(aug$experience) <- list("Some Experience"=c("<1","1","2","3","4"), "Experience"=c("5","6","7","8","9","10","11","12","13","14"), "A lot of Experience"=c("15","16","17","18","19","20",">20"))

str(aug$experience)

table(aug$experience)

#count: Some Experience = 4986, Experience = 8602, A lot of Experience = 5570

#Change 'company\_size' levels

levels(aug$company\_size) <- list("Small"=c("<10","10/49","50-99","100-500"), "Medium"=c("500-999","1000-4999"), "Large"=c("5000-9999","10000+"))

str(aug$company\_size)

table(aug$company\_size)

#count: Small = 13000, Medium = 2694, Large = 3461

#Change 'company\_type' levels

levels(aug$company\_type) <- list("Public"=c("Public Sector"), "Startup"=c("Early Stage Startup","Funded Startup"), "Other"=c("Other","NGO","Pvt Ltd"))

str(aug$company\_type)

table(aug$company\_type)

#count: Public = 1163, Startup = 1682, Other = 16313

#Change 'last\_new\_job' levels

levels(aug$last\_new\_job) <- list("Never"=c("never"), "Recent"=c("1","2"), "Kind of Recent"=c("3","4"), "Not Recent"=c(">4"))

str(aug$last\_new\_job)

table(aug$last\_new\_job)

#count: Never = 2557, Recent = 11216, Kind of Recent = 2054, Not Recent = 3331

#Make 'city\_development\_index' out of 100 for better coefficient interpretation (logistic reg)

aug$city\_development\_index = 100\*(aug$city\_development\_index)

# Train and Test Split #

#Inspect data before splitting

str(aug) #19158 obs & 11 variables

#Check to see which level we're predicting

levels(aug$target)

#reference = "0" (Not looking for job change), predicting = "1" (Looking for job change)

#Split data into 70% train, 30% test

set.seed(100)

ind <- sample(2,nrow(aug), replace= T, prob = c(0.7,0.3))

train <- aug[ind == 1,]

test <- aug[ind==2,]

#Logistic Linear Model

logistic\_model <- glm(target~ city\_development\_index + gender + relevent\_experience +

enrolled\_university + experience + company\_size + company\_type +

last\_new\_job + training\_hours, data = train, family = 'binomial')

summary(logistic\_model)

#Predictions

p1 <- predict(logistic\_model, train, type = 'response')

head(p1)

head(train)

#Misclassification Error - Train data

pred1 <- ifelse(p1>0.5, 1,0)

tab1 <- table(Predicted = pred1, Actual = train$target)

tab1 #Confusion Matrix ( It has predicted 9468 correctly that candidates will not look for job change an

#d 904 candidates will look for job change is predicted correctly )

1 - sum(diag(tab1))/sum(tab1) #misclassification error for Training data is 23.21%

#Predictions on Test data

p2 <- predict(logistic\_model, test, type = 'response')

#misclassification error on Test data

pred2 <- ifelse(p2>0.5,1,0)

tab2 <- table(Predicted = pred2, Actual = test$target)

tab2

1 - sum(diag(tab2))/sum(tab2) #Misclassification Error is 22.9

summary(aug)

#correlation plot

library(ggcorrplot)

model.matrix(~0+., data=aug) %>%

cor(use="pairwise.complete.obs") %>%

ggcorrplot(show.diag = F, type="lower", lab=TRUE, lab\_size=2)

#Goodness of Fit Test

with(logistic\_model, pchisq(null.devaince - deviance, df.null-df.residual, lower.tail= F))

#Linear discriminant Analysis

library(MASS)

lda.fit <- lda(target~., data = train)

lda.fit

#Confusion Matrix and Accuracy - Training data

p3 <- predict(lda.fit, train)$class

tab3 <- table(Predicted=p3, Actual =train$target)

tab3

sum(diag(tab3))/sum(tab3)

#Confusion Matrix and Accuracy - Test Data

p4 <- predict(lda.fit, test)$class

tab4 <- table(Predicted=p4, Actual =test$target)

tab4

sum(diag(tab4))/sum(tab4)

#Quadratic Discriminant Analysis

qda.fit <- qda(target~., data = train)

qda.fit

#Confusion Matrix and Accuracy - Training data

p5 <- predict(qda.fit, train)$class

tab5 <- table(Predicted=p5, Actual =train$target)

tab5

sum(diag(tab5))/sum(tab5)

#Confusion Matrix and Accuracy - Test Data

p6 <- predict(qda.fit, test)$class

tab6 <- table(Predicted=p6, Actual =test$target)

tab6

sum(diag(tab6))/sum(tab6)

#######################################

#Data Partition for Ridge & LASSO######

#######################################

set.seed(100)

ind <- sample(2,nrow(aug), replace= T, prob = c(0.7,0.3))

train <- aug[ind == 1,]

test <- aug[ind==2,]

library(caret)

#Custom Control Parameter

custom <- trainControl(method = "repeatedcv",

number = 10,

repeats = 5,

verboseIter = T)

#Ridge Regression

set.seed(1234)

ridge <- train(target~.,

train,

method ='glmnet',

tuneGrid = expand.grid(alpha =0,

lambda = seq(0.0001,1,length=5)),

trControl = custom)

plot(ridge) #ridge regression plot

ridge #Summary of Ridge Regression

plot(ridge$finalModel, xvar ="lambda", label = T) #Plot of log lambda and coefficients

plot(ridge$finalModel,xvar='dev', label = T) # Plot of Fraction Deviance Explained

plot(varImp(ridge,scale=T))

#LASSO Regression(It does both Shrinkage and Feature selection, if there are group of variables which are highly correlated variables which are causing multicollinearity then LASSO will select one variable from group and will ignore others )

set.seed(1234)

lasso <- train(target~.,

train,

method ='glmnet',

tuneGrid = expand.grid(alpha =1,

lambda = seq(0.0001,1,length=5)),

trControl = custom)

#Plot Results

plot(lasso)

lasso

plot(lasso$finalModel, xvar= 'lambda', label=T)

plot(lasso$finalModel, xvar='dev', label = T)

plot(varImp(lasso,scale=T))

#K Nearest Neighbors

trControl <- trainControl(method ="repeatedcv",

number = 10,

repeats = 3)

set.seed(222)

fit <- train(target~.,

data = train,

method = 'knn',

tuneLength = 20,

trControl = trControl,

preProc = c("center", "scale"))

#Model Performance(K Nearest Neighbours)

fit

plot(fit) # Variable

varImp(fit) #Variable Importance Table

#Prediction on Test data

pred <- predict(fit, newdata= test)

pred

#Confusion Matrix

confusionMatrix(pred, test$target)

library("randomForest")

library(ISLR)

library(tree)

testing\_outcome <- aug$target[test]

?randomForest

bagging <- randomForest(target~.,

data = train,

mtry = 10,

importance = TRUE)

bagging

names(bagging)

summary(bagging)

##plot of importance of features

varImpPlot(bagging)

##Prediction

predict.bagging <- predict(bagging, newdata = test)

#Confusion Matrix - Test Data

confusionMatrix(predict.bagging, test$target)

#Plot the prediction on Testing Outcome

plot(predict.bagging)

table(train$target) #Check how many observations Class 0 and how many belongs to class 1

## 10114 observations belong to class 0 and 3395 observations belongs to Class 1.

prop.table(table(train$target)) # 74% observation belongs to class 0 and 25% observations belongs to class 1

summary(train)

table(test$target)

#Random Forest

rftrain <- randomForest(target~.,

data = train)

#Random Forest Predictive Model Evaluation with Test data

library(e1071)

##Confusion Matrix Test Data

confusionMatrix(predict(rftrain, test), test$target, positive = '0')

#### Tried to perform Oversampling, undesampling using ROSE Libray

library(ROSE)

over <- ovun.sample(target~.,

data = test,

method = "over", N= 8534)$aug

str(over)

table(over$target)

summary(over)

table

train$target <- as.factor$target

data.rose <- ROSE(target~., data = train, seed = 100)$aug

str(data.rose)

table(data.rose$target)

##########