Salary Prediction Using R

Jerome Joshua

4/25/2020

PART 1 -Defining the problem

The goal of the project is to analyze the data about the various jobs of various companies and able to predict salaries of any future jobs

Importing packages the plot.

```
library(dplyr)
library(e1071)
library(caret)
library(ggplot2)
library(tidyr)
library(readr)
library(rattle)
library(ggpubr)
library(rpart)
library(xgboost)

options(scipen=999)
```

PART 2- Discover

Load the input train and test files and store as dataframe

```
train_features <- read_csv("C:/Users/jjosh/Desktop/train_features.csv")

train_salaries <- read_csv("C:/Users/jjosh/Desktop/train_salaries.csv")

test_features <- read_csv("C:/Users/jjosh/Desktop/test_features.csv")
```

Check for duplicate values for unique key (JobId)

There are no duplicate records of JobId

```
chk_duplicate<-train_features %>% group_by(jobId) %>% filter(n()>1)
chk_duplciate2<-train_features %>% group_by(jobId)  %>% summarize(n=n())
```

Exploratory data analysis

Below is the summary of the train_feature dataset Checking for how the data is structured , years of experience ranges from 0 to 24 years Miles from metro city ranges from 0 to 99 miles

```
summary(train_features)
```

```
##     jobId            companyId           jobType             degree
##  Length:1000000     Length:1000000     Length:1000000     Length:1000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     major              industry         yearsExperience milesFromMetropolis
##  Length:1000000     Length:1000000     Min.   : 0.00   Min.   : 0.00
##  Class :character   Class :character   1st Qu.: 6.00   1st Qu.:25.00
##  Mode  :character   Mode  :character   Median :12.00   Median :50.00
##                                        Mean   :11.99   Mean   :49.53
##                                        3rd Qu.:18.00   3rd Qu.:75.00
##                                        Max.   :24.00   Max.   :99.00
```

Below is the summary of the train_salaries dataset Salary ranges from 0 k to 301 k

```
summary(train_salaries)
```

```
##     jobId               salary
##  Length:1000000     Min.   :  0.0
##  Class :character   1st Qu.: 88.0
##  Mode  :character   Median :114.0
##                     Mean   :116.1
##                     3rd Qu.:141.0
##                     Max.   :301.0
```

Removing outliers

Salary cannot be zero hence removing records where salary is 0k

```
train_salaries<-train_salaries[train_salaries$salary != 0, ]
```

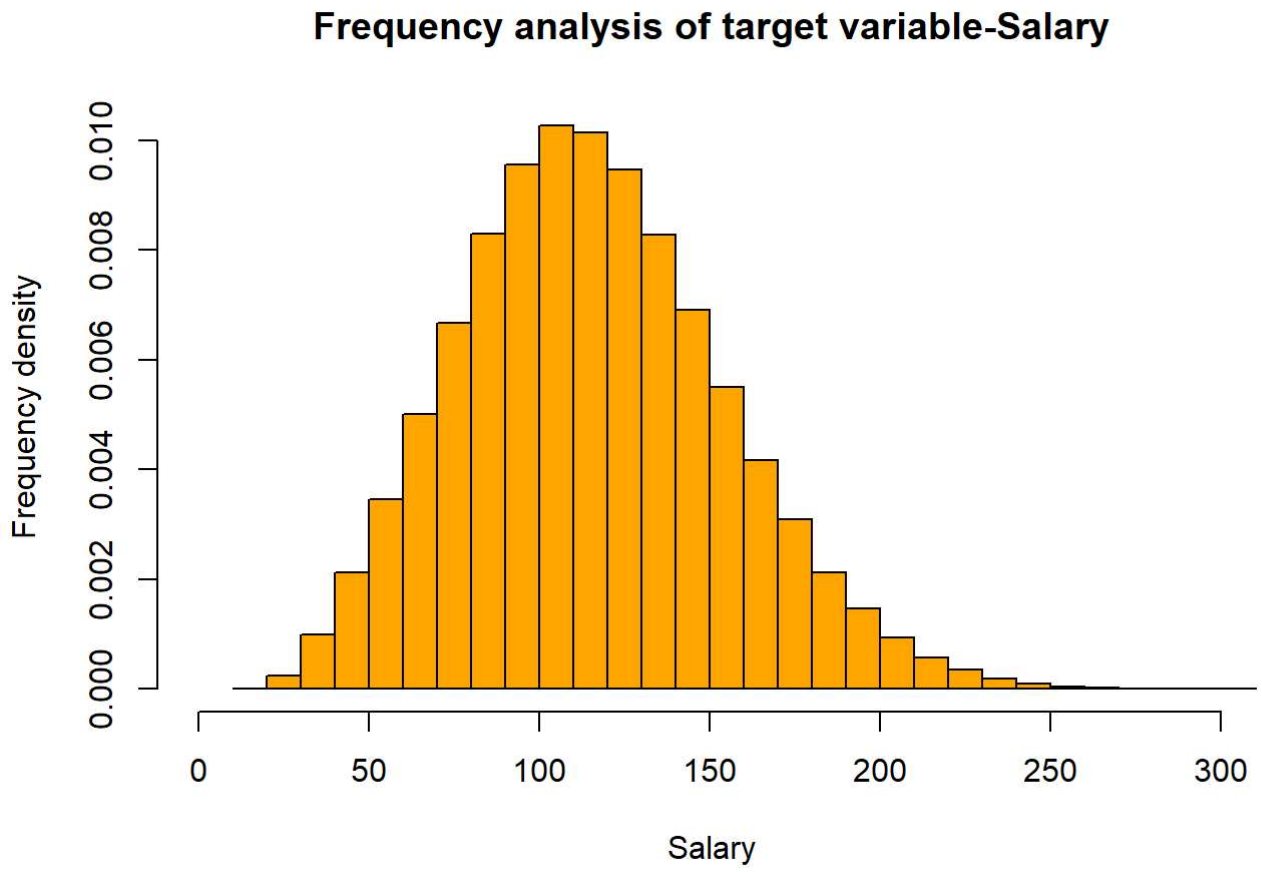
Converting the character type format varaibles into categorical variables

```
train_features$jobType<-as.factor(train_features$jobType)
train_features$degree<-as.factor(train_features$degree)
train_features$major<-as.factor(train_features$major)
train_features$industry<-as.factor(train_features$industry)


test_features$jobType<-as.factor(test_features$jobType)
test_features$degree<-as.factor(test_features$degree)
test_features$major<-as.factor(test_features$major)
test_features$industry<-as.factor(test_features$industry)
```

Analysis of target variable

Histogram of the target variable - Salary Target variable follows the normal distribution

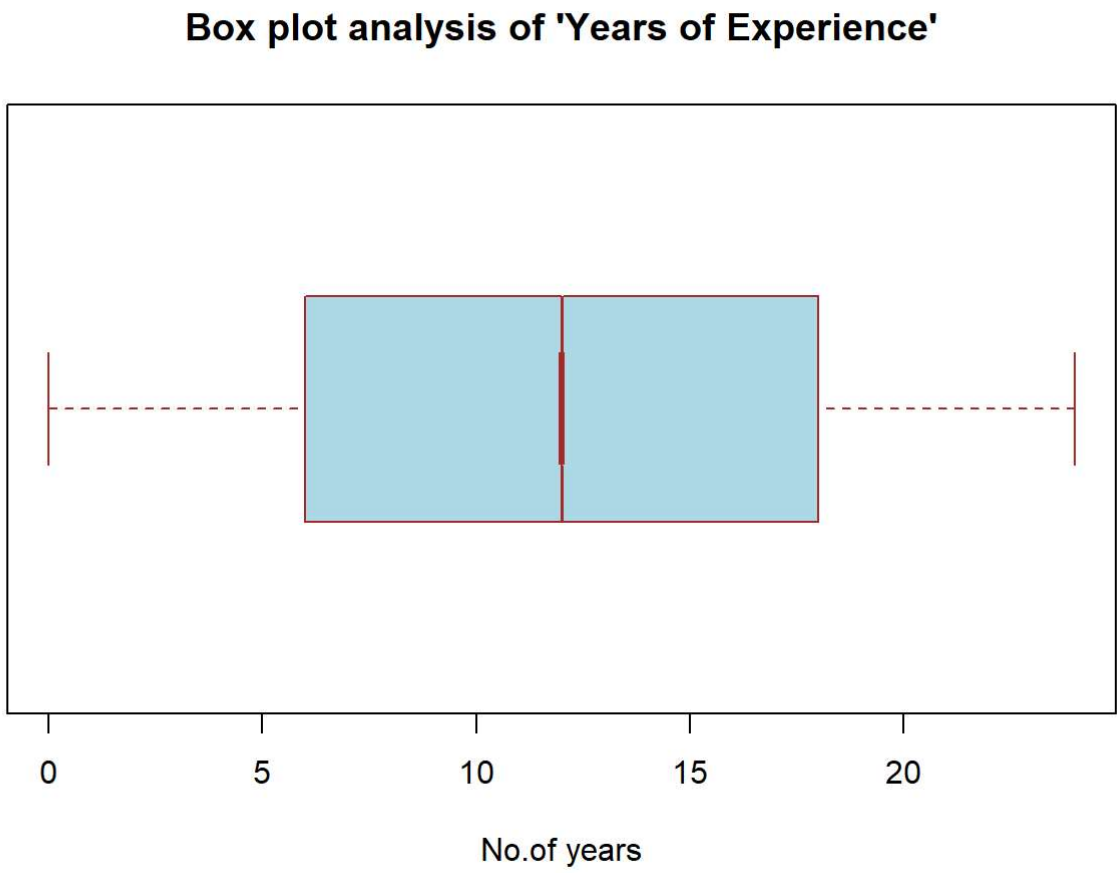
```
hist(train_salaries$salary,
     main="Frequency analysis of target variable-Salary",
     xlab="Salary",
     ylab="Frequency density",
     xlim=c(0,301),
     col="orange",
     freq=FALSE
)
```



Analysis of ‘years of experience’ variable

No outliers in the years of experience

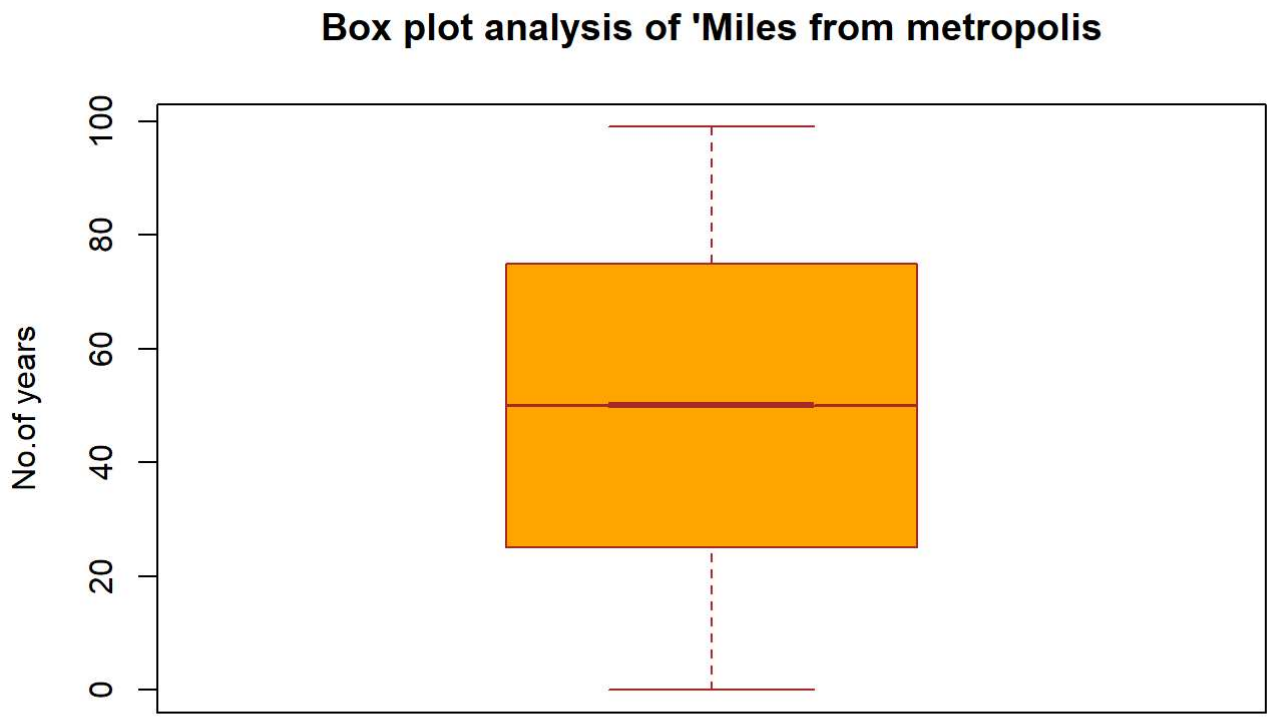
```
boxplot(train_features$yearsExperience,
        main = "Box plot analysis of 'Years of Experience'",
        xlab = "No.of years",
        col = "light blue",
        border = "brown",
        horizontal = TRUE,
        notch = TRUE
)
```



Analysis of ‘miles from metroplois’ variable

No outliers in the distance miles of metropolis variable.

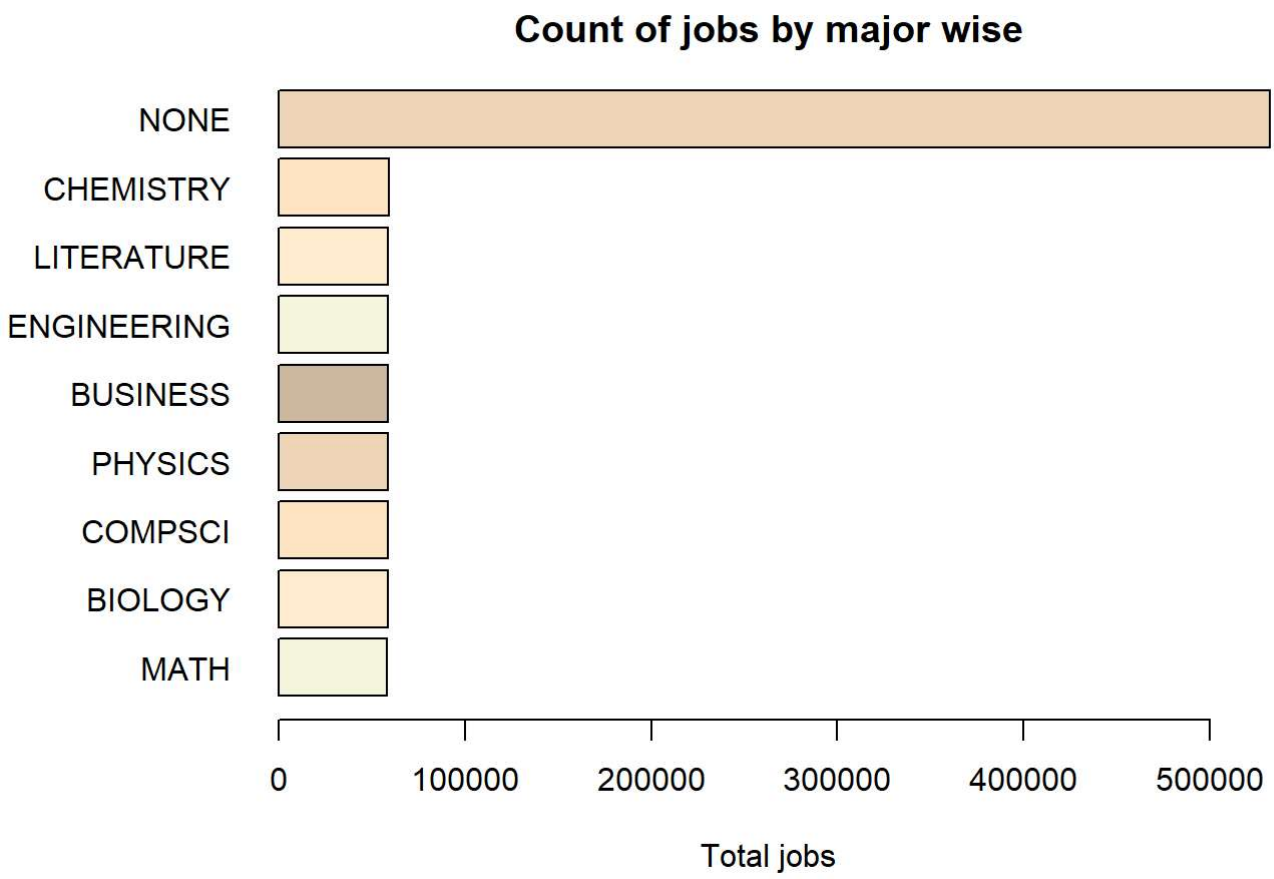
```
boxplot(train_features$milesFromMetropolis,
        main = "Box plot analysis of 'Miles from metropolis",
        ylab = "No.of years",
        col = "orange",
        border = "brown",
        notch = TRUE)
```



Frequency analysis of total jobs by Major

'None' category holds more than 50% of the total jobs

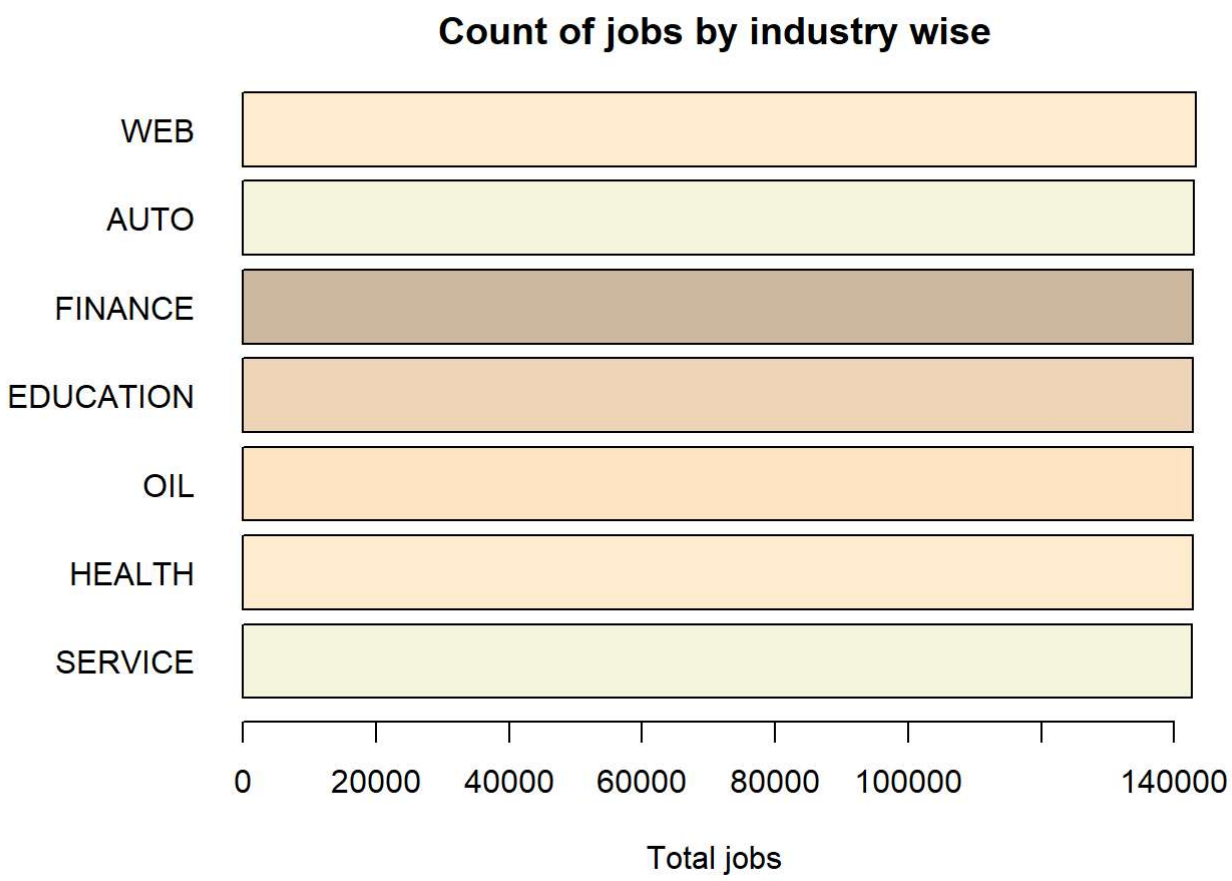
```
table1<-table(train_features$major)
par(oma=c(1,2,1,1))
par(mar=c(4,5,2,1))
barplot(table1[order(table1)],
        horiz = TRUE,
        las = 1,
        col=c("beige","blanchedalmond","bisque1","bisque2","bisque3"),
        main = "Count of jobs by major wise",  xlab = "Total jobs"
)
```



Frequency analysis of total jobs by Industry

The total jobs are distributed almost equally among industries.

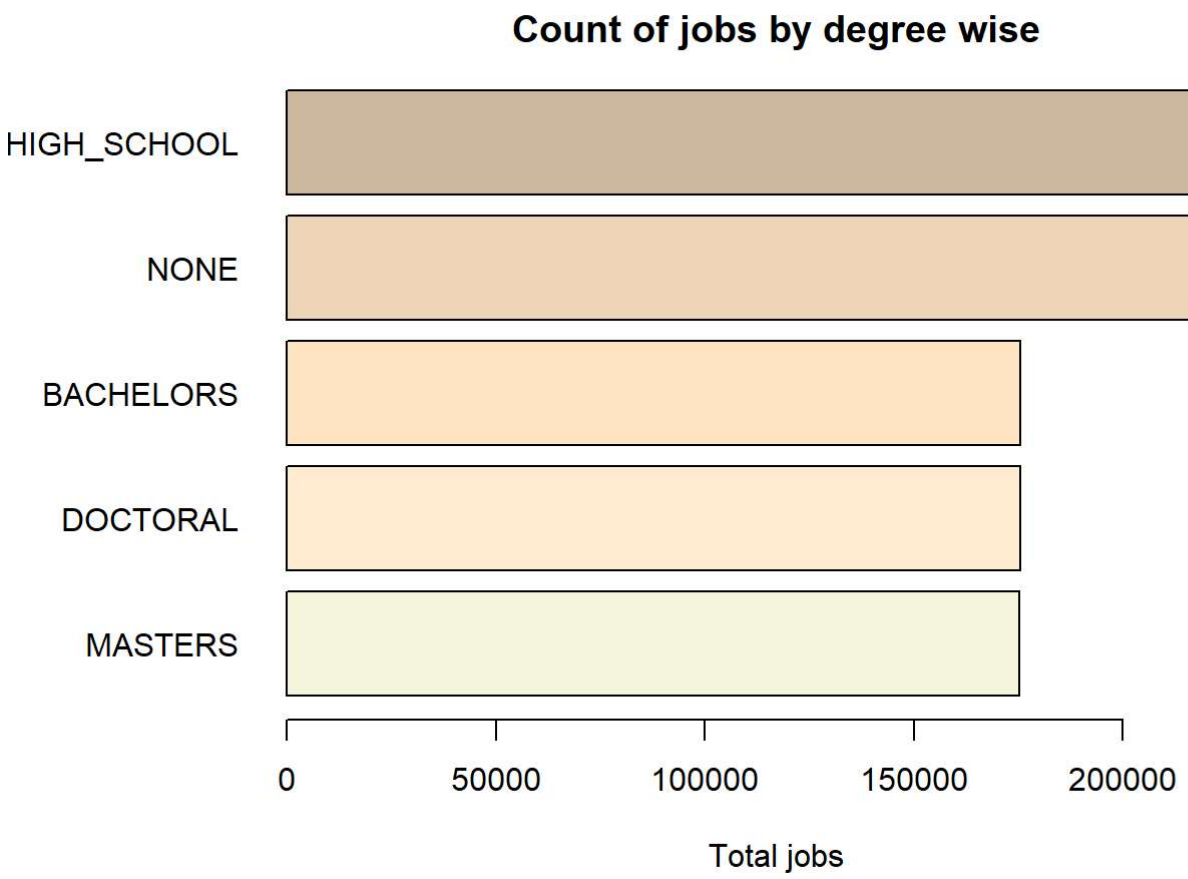
```
table2<-table(train_features$industry)
par(oma=c(1,3,1,1))
par(mar=c(4,5,2,1))
barplot(table2[order(table2)],
        horiz = TRUE,
        las = 1,
        col=c("beige","blanchedalmond","bisque1","bisque2","bisque3"),
        main = "Count of jobs by industry wise",  xlab = "Total jobs"
)
```



Frequency analysis of total jobs by degree

High school and None categories have more jobs than other degrees.

```
table3<-table(train_features$degree)
par(oma=c(1,2,1,1))
par(mar=c(4,5,2,1))
barplot(table3[order(table3)],
        horiz = TRUE,
        las = 1,
        col=c("beige","blanchedalmond","bisque1","bisque2","bisque3"),
        main = "Count of jobs by degree wise",  xlab = "Total jobs"
)
```



Merging dependent and independent variables

```
train_all<-merge(train_features,train_salaries)
```

Correlation analysis of continous variables

Correlation between experience and salary is only 37% which is weakly correlated.

```
cor.test(train_all$yearsExperience, train_all$salary,
         method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  train_all$yearsExperience and train_all$salary
## t = 404.54, df = 999993, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3733278 0.3766965
## sample estimates:
##       cor
## 0.3750134
```

Correlation between Miles from metropolis and salary is only negative 29% which is weakly correlated.

```
cor.test(train_all$milesFromMetropolis, train_all$salary,
         method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  train_all$milesFromMetropolis and train_all$salary
## t = -311.82, df = 999993, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2994717 -0.2958991
## sample estimates:
##       cor
## -0.2976864
```

Correlation matirx for the continous variables There is no strong relation of continous variables with target variable

```
M<-train_all[,c(7,8,9)]
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

M$yearsExperience<-normalize(train_all$yearsExperience)
M$salary<-normalize(train_all$salary)
M$milesFromMetropolis<-normalize(train_all$milesFromMetropolis)
M<-as.matrix(M)

library(corrplot)
```

```
## corrplot 0.84 loaded
```

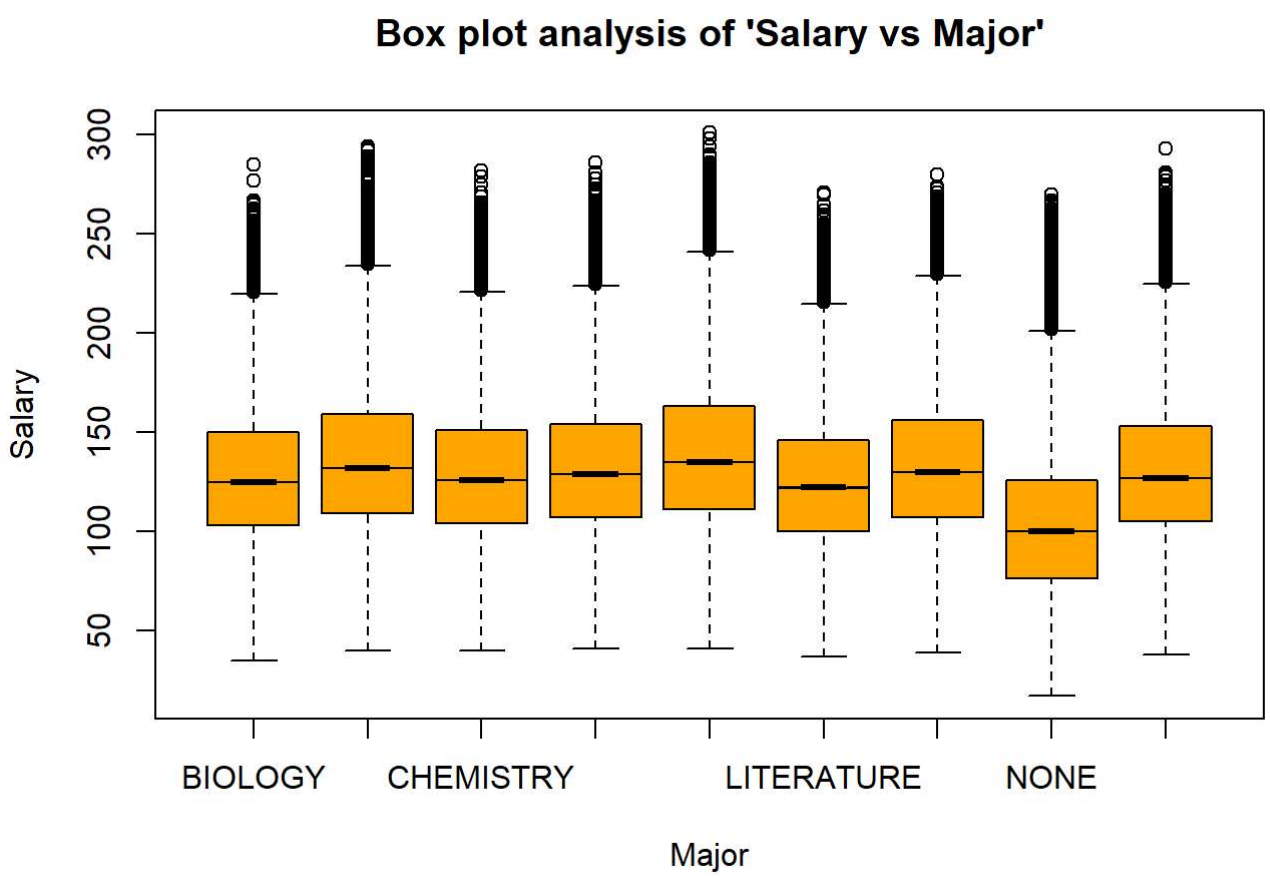
```
Mat <- cor(M)
corrplot(Mat, method = "circle")
```

Correlation analysis of categorical variables

Since correlation cannot be computed directly , we use Analysis of variance method and finding the relation between target and caterogical variables.

Relation between major and salary

```
boxplot(train_all$salary~train_all$major,
        main = "Box plot analysis of 'Salary vs Major'",
        xlab = "Major",
        ylab="Salary",
        col = "orange",
        border = "black",
        notch = TRUE)
```



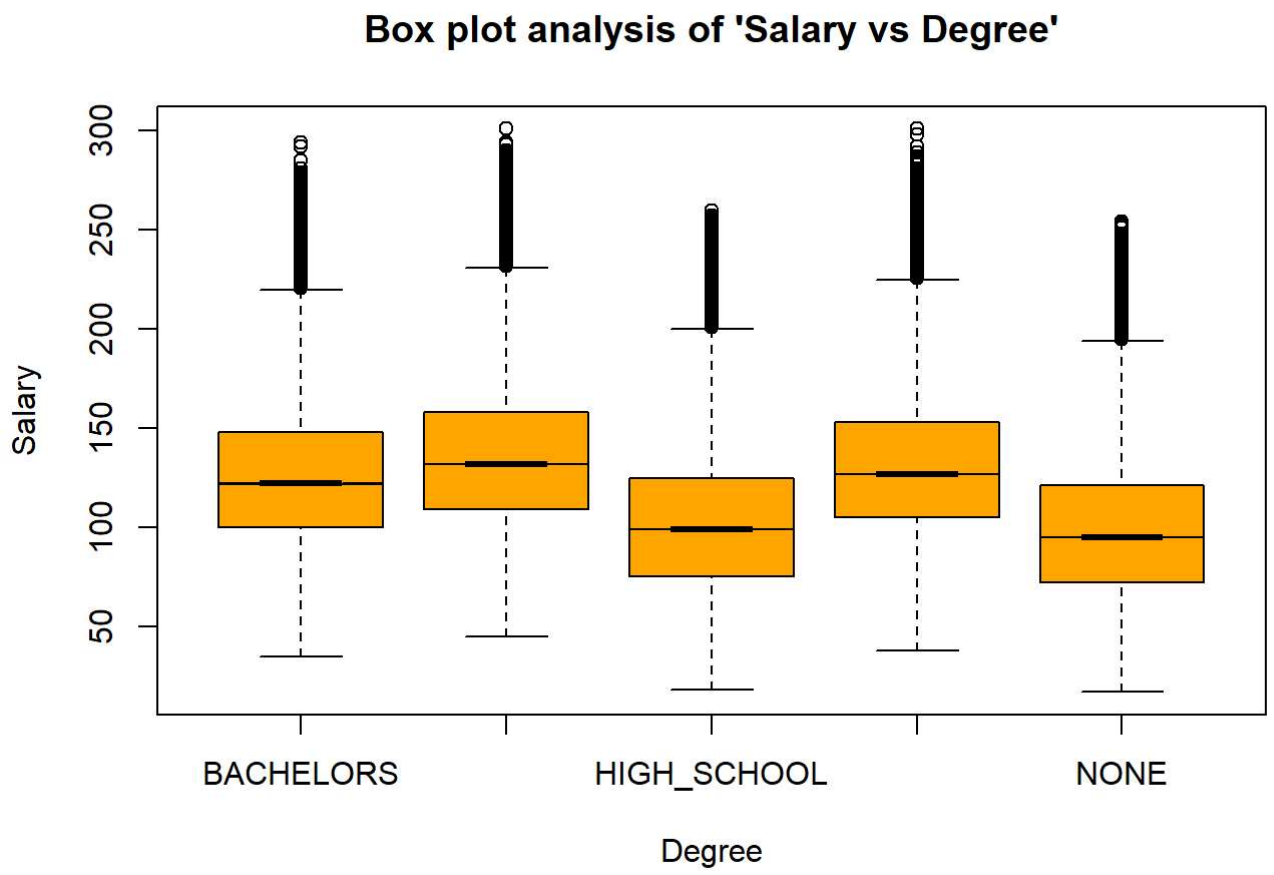

```
aov1<-aov(train_all$salary~train_all$major)
summary(aov1)
```

```
##                     Df     Sum Sq  Mean Sq F value              Pr(>F)
## train_all$major      8  214949861 26868733   20925 <0.0000000000000002 ***
## Residuals       999986 1284059886     1284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p-value is very less and significant. There is no significant difference between the categories of major with respect to salary

Relation between degree and salary

```
boxplot(train_all$salary~train_all$degree,
        main = "Box plot analysis of 'Salary vs Degree'",
        xlab = "Degree",
        ylab="Salary",
        col = "orange",
        border = "black",
        notch = TRUE)
```



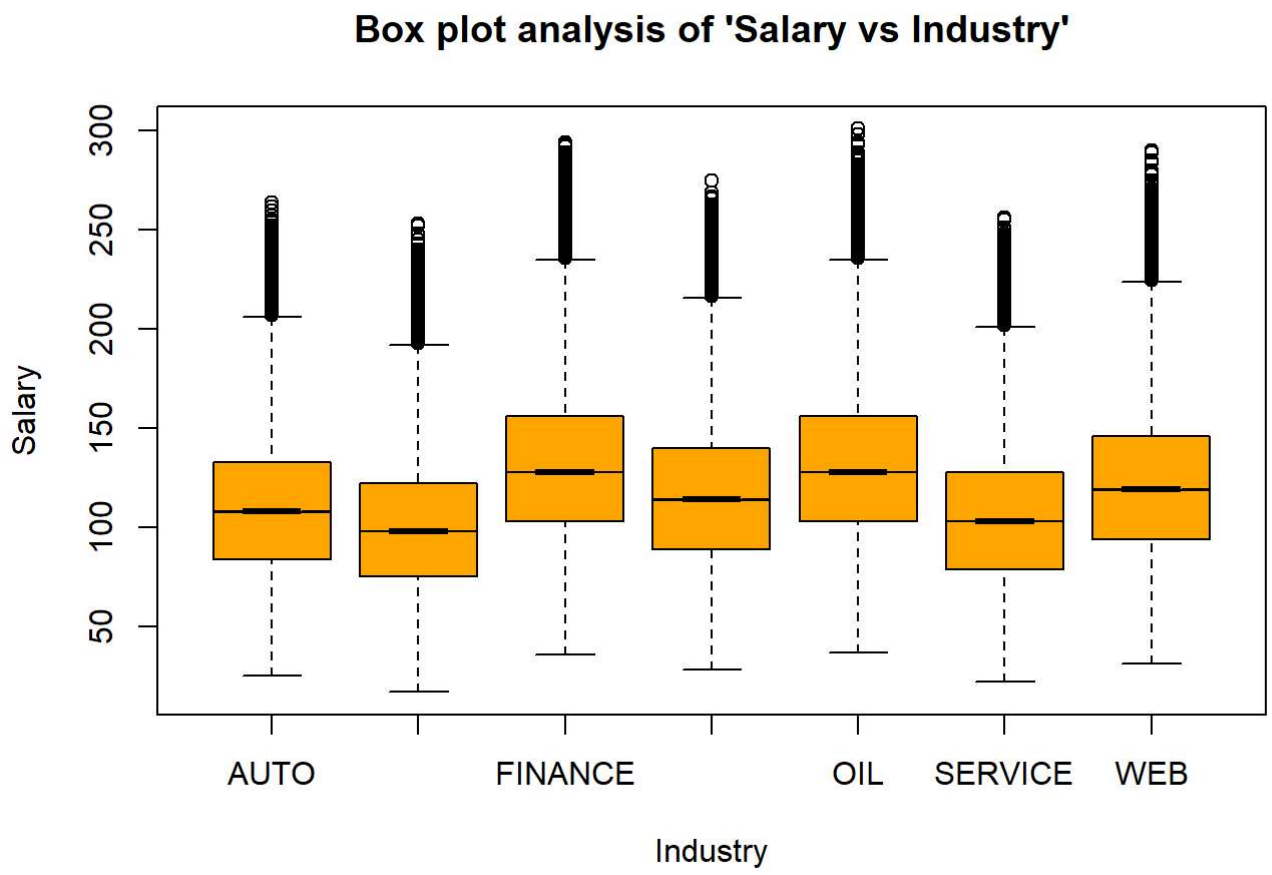

```
aov2<-aov(train_all$salary~train_all$degree)
summary(aov2)
```

```
##                      Df     Sum Sq  Mean Sq F value              Pr(>F)
## train_all$degree      4  241402844 60350711   47988 <0.0000000000000002 ***
## Residuals        999990 1257606903     1258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p-value is very less and significant. There is correlation between the categories of degree with respect to salary

Relation between industry and salary

```
boxplot(train_all$salary~train_all$industry,
        main = "Box plot analysis of 'Salary vs Industry'",
        xlab = "Industry",
        ylab="Salary",
        col = "orange",
        border = "black",
        notch = TRUE)
```




```
aov3<-aov(train_all$salary~train_all$industry)
summary(aov3)
```

```
##                         Df     Sum Sq  Mean Sq F value              Pr(>F)
## train_all$industry       6  131893631 21982272   16079 <0.0000000000000002 ***
## Residuals           999988 1367116115     1367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

p-value is very less and significant. There is correlation between the categories of industry with respect to salary

Baseline model

Baseline model is created by using average salary of industry

```
baseline<-as.data.frame(train_all%>%group_by(industry)%>%summarise(Avg.salary=mean(salary)))

train_all_with_baseline<-merge(train_all,baseline,by.x = "industry",by.y = "industry")

train_all_with_baseline$Square_error<-(train_all_with_baseline$salary-train_all_with_baseline$Av
g.salary)^2

mean(train_all_with_baseline$Square_error)
```

```
## [1] 1367.123
```

Mean square error for predicted salary (baseline) is 1367.123

Since the target variable is continous variable , we will use regression models to predict the salary ### Goal to build 3 models and compare the better performing models

Model 1 using Linear regression.

```
model1_linear_regression<-lm (formula =salary ~ degree+major+industry+yearsExperience+milesFromM
etropolis+jobType , data=train_all)
summary(model1_linear_regression)
```

```
##
## Call:
## lm(formula = salary ~ degree + major + industry + yearsExperience +
##     milesFromMetropolis + jobType, data = train_all)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.642 -14.178  -0.452  13.267  94.751
##
## Coefficients:
##                          Estimate  Std. Error  t value            Pr(>|t|)
## (Intercept)           134.8692441   0.1231437 1095.219 <0.0000000000000002 ***
## degreeDOCTORAL         10.0319623   0.0661998  151.541 <0.0000000000000002 ***
## degreeHIGH_SCHOOL      -5.7169773   0.0997074  -57.338 <0.0000000000000002 ***
## degreeMASTERS           4.9982425   0.0662048   75.497 <0.0000000000000002 ***
## degreeNONE             -9.4015546   0.0997185  -94.281 <0.0000000000000002 ***
## majorBUSINESS           7.6988638   0.1146880   67.129 <0.0000000000000002 ***
## majorCHEMISTRY          1.0803516   0.1145141    9.434 <0.0000000000000002 ***
## majorCOMPSCI            4.0407512   0.1147549   35.212 <0.0000000000000002 ***
## majorENGINEERING       10.6094192   0.1146510   92.537 <0.0000000000000002 ***
## majorLITERATURE        -3.6204892   0.1146081  -31.590 <0.0000000000000002 ***
## majorMATH               5.1443748   0.1150437   44.717 <0.0000000000000002 ***
## majorNONE              -4.9558693   0.1146847  -43.213 <0.0000000000000002 ***
## majorPHYSICS            2.3262112   0.1147412   20.274 <0.0000000000000002 ***
## industryEDUCATION      -9.9916368   0.0733536 -136.212 <0.0000000000000002 ***
## industryFINANCE        21.1425169   0.0733471  288.253 <0.0000000000000002 ***
## industryHEALTH          6.2457135   0.0733620   85.136 <0.0000000000000002 ***
## industryOIL            21.3087897   0.0733600  290.469 <0.0000000000000002 ***
## industrySERVICE        -4.9819100   0.0733767  -67.895 <0.0000000000000002 ***
## industryWEB            12.1098931   0.0733038  165.201 <0.0000000000000002 ***
## yearsExperience         2.0100671   0.0027184  739.435 <0.0000000000000002 ***
## milesFromMetropolis    -0.3995293   0.0006789 -588.464 <0.0000000000000002 ***
## jobTypeCFO             -9.8029600   0.0785587 -124.785 <0.0000000000000002 ***
## jobTypeCTO             -9.7872758   0.0784521 -124.755 <0.0000000000000002 ***
## jobTypeJANITOR        -62.3581863   0.0825914 -755.020 <0.0000000000000002 ***
## jobTypeJUNIOR         -49.7937022   0.0785236 -634.124 <0.0000000000000002 ***
## jobTypeMANAGER        -29.8656534   0.0784403 -380.744 <0.0000000000000002 ***
## jobTypeSENIOR         -39.7857537   0.0783209 -507.984 <0.0000000000000002 ***
## jobTypeVICE_PRESIDENT -19.9445803   0.0784229 -254.321 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.61 on 999967 degrees of freedom
## Multiple R-squared:  0.7436, Adjusted R-squared:  0.7436
## F-statistic: 1.074e+05 on 27 and 999967 DF,  p-value: < 0.00000000000000022
```

```
train_all$model1<-predict (model1_linear_regression ,train_all)

train_all$Square_error<-(train_all$salary-train_all$model1)^2

mean(train_all$Square_error)
```

```
## [1] 384.3803
```

Mean square error for predicted salary using Linear regression is 384

Model 2 using decision trees

```
model2_decisiontree=rpart(salary~degree+major+industry+yearsExperience+milesFromMetropolis+jobTy
pe,data=train_all)

train_all$model2_decisiontree<-predict(model2_decisiontree ,train_all)

train_all$Square_error2<-(train_all$salary-train_all$model2_decisiontree)^2

mean(train_all$Square_error2)
```

```
## [1] 701.934
```

Mean square error for predicted salary using Decision trees is 710

Model 3 using xgboost

Data has to prepared in the format of matrix for xgboost COnverting categorical features into binary varaibles using one hot encoding

```
category_vars = c('jobType', 'degree', 'major','industry')
dummy_vars <- dummyVars(~ jobType +  degree + major+industry, data = train_all)
one_hot_encode_category_vars <- as.data.frame(predict(dummy_vars, newdata = train_all))
```

Combining category features and numeric features

```
xg_model_train_data<-cbind(one_hot_encode_category_vars,train_all[,c(7,8,9)])
y<-as.numeric(xg_model_train_data$salary)
```

Building XG boost Model

```
xgb <- xgboost(data = data.matrix(xg_model_train_data[,-32]),
               label = y,
               booster = "gblinear",
               objective = "reg:squarederror",
               max.depth = 5,
               nround = 50,
               lambda = 0,
               lambda_bias = 0,
               alpha = 0
)
```

```
## [1]  train-rmse:32.563023
## [2]  train-rmse:24.510811
## [3]  train-rmse:21.974651
## [4]  train-rmse:20.974209
## [5]  train-rmse:20.531298
## [6]  train-rmse:20.261835
## [7]  train-rmse:20.072617
## [8]  train-rmse:19.950211
## [9]  train-rmse:19.859789
## [10] train-rmse:19.795593
## [11] train-rmse:19.747232
## [12] train-rmse:19.711535
## [13] train-rmse:19.685266
## [14] train-rmse:19.666195
## [15] train-rmse:19.652025
## [16] train-rmse:19.641500
## [17] train-rmse:19.633698
## [18] train-rmse:19.627802
## [19] train-rmse:19.623323
## [20] train-rmse:19.619926
## [21] train-rmse:19.617109
## [22] train-rmse:19.614946
## [23] train-rmse:19.613310
## [24] train-rmse:19.612062
## [25] train-rmse:19.611052
## [26] train-rmse:19.610237
## [27] train-rmse:19.609570
## [28] train-rmse:19.608978
## [29] train-rmse:19.608498
## [30] train-rmse:19.608116
## [31] train-rmse:19.607738
## [32] train-rmse:19.607433
## [33] train-rmse:19.607210
## [34] train-rmse:19.606983
## [35] train-rmse:19.606783
## [36] train-rmse:19.606615
## [37] train-rmse:19.606445
## [38] train-rmse:19.606308
## [39] train-rmse:19.606205
## [40] train-rmse:19.606102
## [41] train-rmse:19.606022
## [42] train-rmse:19.605946
## [43] train-rmse:19.605877
## [44] train-rmse:19.605829
## [45] train-rmse:19.605761
## [46] train-rmse:19.605726
## [47] train-rmse:19.605661
## [48] train-rmse:19.605640
## [49] train-rmse:19.605625
## [50] train-rmse:19.605597
```

```
train_all$xgboost_model = predict(xgb, newdata = as.matrix(xg_model_train_data[,-32]))

train_all$Square_error3<-(train_all$salary-train_all$xgboost_model)^2

mean(train_all$Square_error3)
```

```
## [1] 384.3877
```

Feature Engineering

Mean ,median ,max and min yearsExperience are computed for each of the category variables(Industry , major, degree,JobType)

```
train_all<-merge(train_features,train_salaries)

jobType_stats<-train_all%>%group_by(jobType)%>%summarise(avg.salary_jobtype=mean(yearsExperienc
e),median.salary_jobtype=median(yearsExperience),min.salary_jobtype=min(yearsExperience),max.sal
ary_jobtype=max(yearsExperience))
industry_stats<-train_all%>%group_by(industry)%>%summarise(avg.salary_industry=mean(yearsExperie
nce),median.salary_industry=median(yearsExperience),min.salary_industry=min(yearsExperience),ma
x.salary_industry=max(yearsExperience))
degree_stats<-train_all%>%group_by(degree)%>%summarise(avg.salary_degree=mean(yearsExperience),m
edian.salary_degree=median(yearsExperience),min.salary_degree=min(yearsExperience),max.salary_de
gree=max(yearsExperience))
major_stats<-train_all%>%group_by(major)%>%summarise(avg.salary_major=mean(yearsExperience),medi
an.salary_major=median(yearsExperience),min.salary_major=min(yearsExperience),max.salary_major=m
ax(yearsExperience))


mean_jobtype<-merge(jobType_stats,train_all[,c(1,3)])
mean2_jobtype<-merge(industry_stats,train_all[,c(1,6)])
mean3_jobtype<-merge(degree_stats,train_all[,c(1,4)])
mean4_jobtype<-merge(major_stats,train_all[,c(1,5)])


train_all_new<-merge(train_all,mean_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean2_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean3_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean4_jobtype,by="jobId")
final_data_<-train_all_new[,c(-1,-2,-10,-15,-20,-25)]
names(final_data_)[1]<-"jobType"
names(final_data_)[2]<-"degree"
names(final_data_)[3]<-"major"
names(final_data_)[4]<-"industry"
```

```
dummy_vars <- dummyVars(~ jobType +  degree + major+industry, data = train_all)
one_hot_encode_category_vars <- as.data.frame(predict(dummy_vars, newdata = final_data_))
xg_model_train_data<-cbind(one_hot_encode_category_vars,final_data_[,c(-1,-2,-3,-4)])
names(xg_model_train_data) <- gsub("[.]", "", names(xg_model_train_data))
y<-as.numeric(xg_model_train_data$salary)
```

Linear regression model after feature engineering

The MSE for linear regression model after feature engineering is 384

```
model1_linear_regression<-lm (formula =salary ~ . , data=final_data_)
summary(model1_linear_regression)
```

```
##
## Call:
## lm(formula = salary ~ ., data = final_data_)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.642 -14.178  -0.452  13.267  94.751
##
## Coefficients: (15 not defined because of singularities)
##                                Estimate         Std. Error  t value
## (Intercept)             530458611.0194679 6450258502.5445757    0.082
## jobTypeCFO                     -9.8029729          0.0785589 -124.785
## jobTypeCTO                     -9.7872759          0.0784521 -124.755
## jobTypeJANITOR                -62.3581882          0.0825915 -755.020
## jobTypeJUNIOR                 -49.7937024          0.0785236 -634.124
## jobTypeMANAGER                -29.8656535          0.0784403 -380.744
## jobTypeSENIOR                 -39.7857537          0.0783209 -507.984
## jobTypeVICE_PRESIDENT         -19.9445802          0.0784229 -254.321
## degreeDOCTORAL                 10.0319623          0.0661998  151.541
## degreeHIGH_SCHOOL              -5.7169803          0.0997074  -57.338
## degreeMASTERS                   4.9982333          0.0662049   75.496
## degreeNONE                     -9.4015576          0.0997185  -94.281
## majorBUSINESS                   7.6988638          0.1146881   67.129
## majorCHEMISTRY                  1.0803516          0.1145142    9.434
## majorCOMPSCI                    4.0407512          0.1147550   35.212
## majorENGINEERING               10.6094191          0.1146510   92.537
## majorLITERATURE                -3.6204892          0.1146082  -31.590
## majorMATH                       5.1443470          0.1150443   44.716
## majorNONE                      -4.9558693          0.1146847  -43.213
## majorPHYSICS                    2.3262112          0.1147412   20.274
## industryEDUCATION         1124641.1442316   13675510.6717848    0.082
## industryFINANCE           1548534.0036648   18829576.1082162    0.082
## industryHEALTH             -82443.7669856    1002574.0361560   -0.082
## industryOIL               1410749.1702840   17154140.7904712    0.082
## industrySERVICE            883656.1682406   10745125.4026394    0.082
## industryWEB                601257.2136889    7311008.3394612    0.082
## yearsExperience                2.0100672          0.0027184  739.434
## milesFromMetropolis           -0.3995294          0.0006789 -588.464
## avg.salary_jobtype                    NA                 NA       NA
## median.salary_jobtype                 NA                 NA       NA
## min.salary_jobtype                    NA                 NA       NA
## max.salary_jobtype                    NA                 NA       NA
## avg.salary_industry     -44298206.2278177  538656453.2669185   -0.082
## median.salary_industry                NA                 NA       NA
## min.salary_industry                   NA                 NA       NA
## max.salary_industry                   NA                 NA       NA
## avg.salary_degree                     NA                 NA       NA
## median.salary_degree                  NA                 NA       NA
## min.salary_degree                     NA                 NA       NA
## max.salary_degree                     NA                 NA       NA
## avg.salary_major                      NA                 NA       NA
## median.salary_major                   NA                 NA       NA
## min.salary_major                      NA                 NA       NA
```

```
## max.salary_major                      NA                 NA       NA
##                                 Pr(>|t|)
## (Intercept)                        0.934
## jobTypeCFO              <0.0000000000000002 ***
## jobTypeCTO              <0.0000000000000002 ***
## jobTypeJANITOR          <0.0000000000000002 ***
## jobTypeJUNIOR           <0.0000000000000002 ***
## jobTypeMANAGER          <0.0000000000000002 ***
## jobTypeSENIOR           <0.0000000000000002 ***
## jobTypeVICE_PRESIDENT   <0.0000000000000002 ***
## degreeDOCTORAL          <0.0000000000000002 ***
## degreeHIGH_SCHOOL       <0.0000000000000002 ***
## degreeMASTERS           <0.0000000000000002 ***
## degreeNONE              <0.0000000000000002 ***
## majorBUSINESS           <0.0000000000000002 ***
## majorCHEMISTRY          <0.0000000000000002 ***
## majorCOMPSCI            <0.0000000000000002 ***
## majorENGINEERING        <0.0000000000000002 ***
## majorLITERATURE         <0.0000000000000002 ***
## majorMATH               <0.0000000000000002 ***
## majorNONE               <0.0000000000000002 ***
## majorPHYSICS            <0.0000000000000002 ***
## industryEDUCATION                  0.934
## industryFINANCE                    0.934
## industryHEALTH                     0.934
## industryOIL                        0.934
## industrySERVICE                    0.934
## industryWEB                        0.934
## yearsExperience         <0.0000000000000002 ***
## milesFromMetropolis     <0.0000000000000002 ***
## avg.salary_jobtype                    NA
## median.salary_jobtype                 NA
## min.salary_jobtype                    NA
## max.salary_jobtype                    NA
## avg.salary_industry                0.934
## median.salary_industry                NA
## min.salary_industry                   NA
## max.salary_industry                   NA
## avg.salary_degree                     NA
## median.salary_degree                  NA
## min.salary_degree                     NA
## max.salary_degree                     NA
## avg.salary_major                      NA
## median.salary_major                   NA
## min.salary_major                      NA
## max.salary_major                      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.61 on 999966 degrees of freedom
## Multiple R-squared:  0.7436, Adjusted R-squared:  0.7436
## F-statistic: 1.036e+05 on 28 and 999966 DF,  p-value: < 0.00000000000000022
```

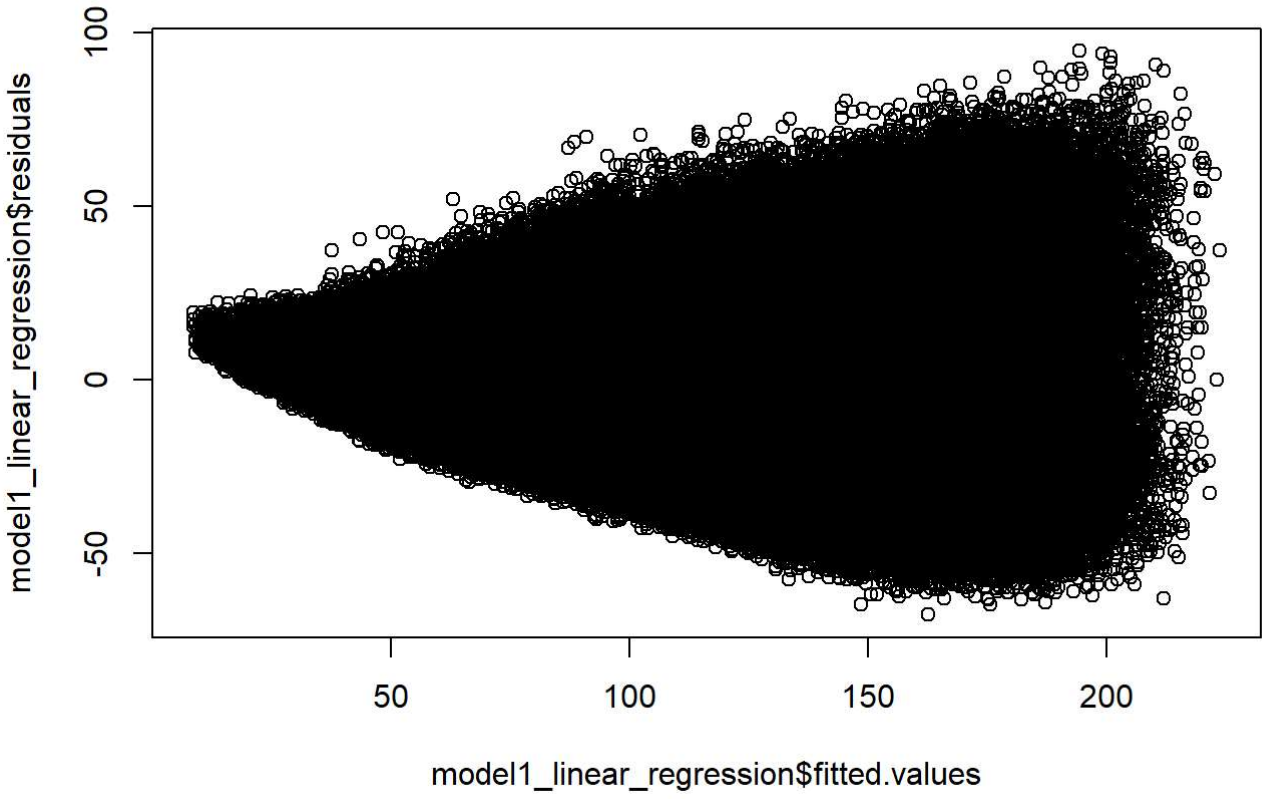
```
train_all$model1<-predict (model1_linear_regression ,final_data_)
```

```
## Warning in predict.lm(model1_linear_regression, final_data_): prediction from a
## rank-deficient fit may be misleading
```

```
train_all$Square_error<-(train_all$salary-train_all$model1)^2

mean(train_all$Square_error)
```

```
## [1] 384.3803
```

```
plot(model1_linear_regression$fitted.values,model1_linear_regression$residuals)
```



From the plot there exists heteroskedasticity ie) larger differences occur when response value is larger

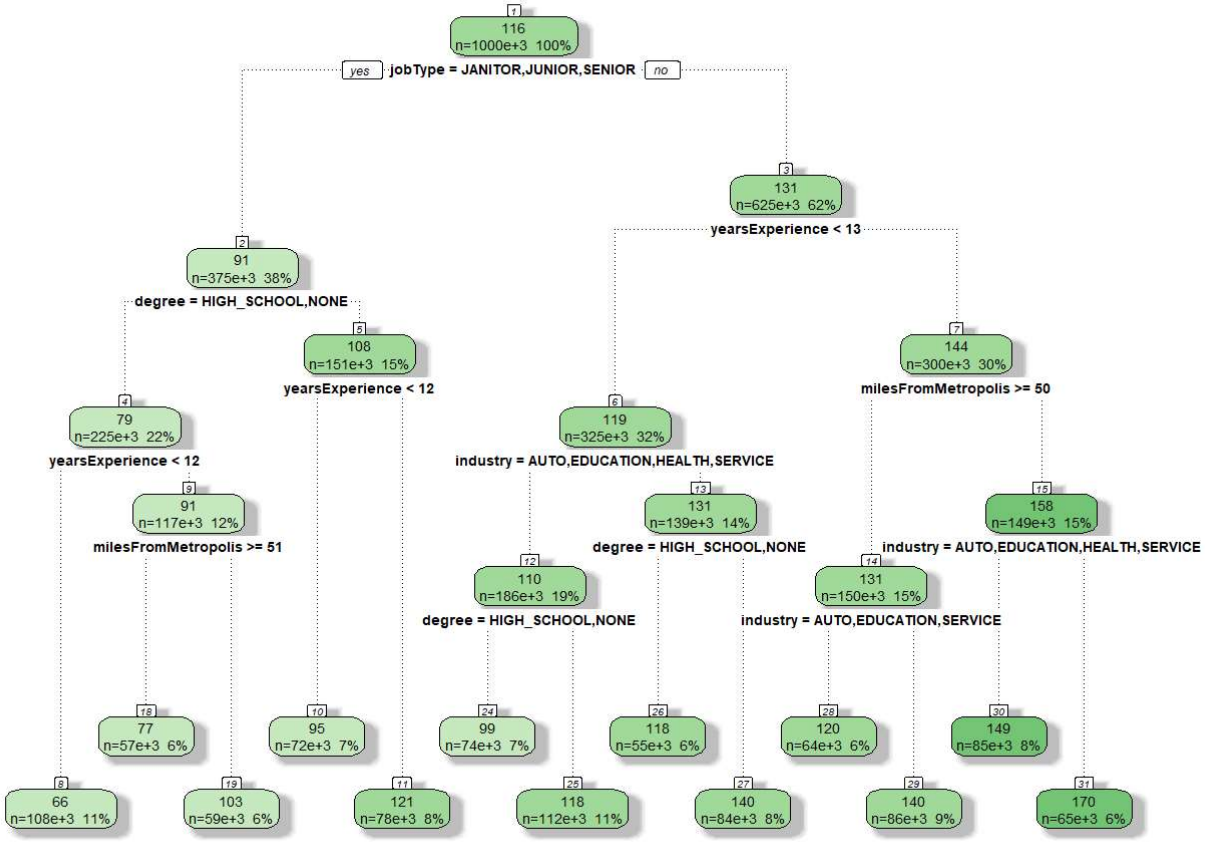
Decision tree model after feature engineering

The MSE for decision tree model after feature engineering is 701

```
model2_decisiontree=rpart(salary~.,data=final_data_)

train_all$model2_decisiontree<-predict(model2_decisiontree ,final_data_)

train_all$Square_error2<-(train_all$salary-train_all$model2_decisiontree)^2

mean(train_all$Square_error2)
```

```
## [1] 701.934
```

```
fancyRpartPlot(model2_decisiontree)
```



XG boost model after feature engineering

Combining category features and numeric features The MSE of xgboost model after feature engineering is 103

Building XG boost Model

```
xg_model_train_data[,c(1:48)] <-round(xg_model_train_data[,c(1:48)],0)
xgb_final <-xgboost(data = data.matrix(xg_model_train_data[,-32]),
                    label = y,
                    booster = "gblinear",
                    objective = "reg:squarederror",
                    max.depth = 5,
                    nround = 50,
                    lambda = 0,
                    lambda_bias = 0,
                    alpha = 0
)
```

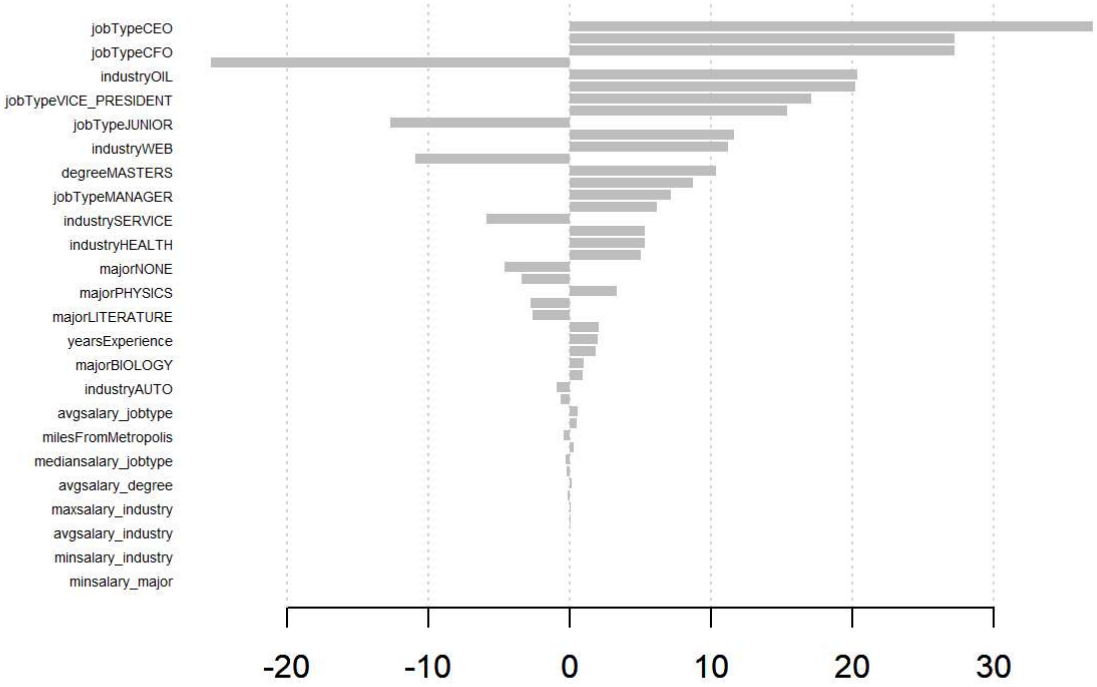
```
## [1]  train-rmse:41.910141
## [2]  train-rmse:31.517422
## [3]  train-rmse:28.105951
## [4]  train-rmse:26.063887
## [5]  train-rmse:24.591965
## [6]  train-rmse:23.352100
## [7]  train-rmse:22.487932
## [8]  train-rmse:21.785324
## [9]  train-rmse:21.237358
## [10] train-rmse:20.846256
## [11] train-rmse:20.546389
## [12] train-rmse:20.313883
## [13] train-rmse:20.139116
## [14] train-rmse:20.005602
## [15] train-rmse:19.910851
## [16] train-rmse:19.839643
## [17] train-rmse:19.785269
## [18] train-rmse:19.743301
## [19] train-rmse:19.713135
## [20] train-rmse:19.690571
## [21] train-rmse:19.672207
## [22] train-rmse:19.658209
## [23] train-rmse:19.647346
## [24] train-rmse:19.639116
## [25] train-rmse:19.632620
## [26] train-rmse:19.627613
## [27] train-rmse:19.623585
## [28] train-rmse:19.620438
## [29] train-rmse:19.617760
## [30] train-rmse:19.615900
## [31] train-rmse:19.614264
## [32] train-rmse:19.612907
## [33] train-rmse:19.611805
## [34] train-rmse:19.610893
## [35] train-rmse:19.610085
## [36] train-rmse:19.609440
## [37] train-rmse:19.608910
## [38] train-rmse:19.608448
## [39] train-rmse:19.608059
## [40] train-rmse:19.607712
## [41] train-rmse:19.607403
## [42] train-rmse:19.607164
## [43] train-rmse:19.606928
## [44] train-rmse:19.606749
## [45] train-rmse:19.606606
## [46] train-rmse:19.606445
## [47] train-rmse:19.606319
## [48] train-rmse:19.606180
## [49] train-rmse:19.606083
## [50] train-rmse:19.605984
```

```
xg_model_train_data$xgboost_model = predict(xgb_final, newdata = as.matrix(xg_model_train_data[,
-32]))

xg_model_train_data$Square_error3<-(xg_model_train_data$salary-xg_model_train_data$xgboost_mode
l)^2

mean(xg_model_train_data$Square_error3)
```

```
## [1] 384.4037
```

Plot of important variables usign XG boost model

```
importance_matrix <- xgb.importance( model = xgb_final)
xgb.plot.importance(importance_matrix)
```



Selection of best model

MSE of linear regression model is 384 MSE of Decisin tree regression is 701 MSE of XG boost model regression is 103.

The best model is XG boost model.

PART 4 -Deployment

Scoring the test data using the final XG boost model.

Importing the test data and extracting features and scoring the test data using the final model

```
test_features <- read_csv("C:/Users/jjosh/Desktop/test_features.csv")
```

```
## Parsed with column specification:
## cols(
##   jobId = col_character(),
##   companyId = col_character(),
##   jobType = col_character(),
##   degree = col_character(),
##   major = col_character(),
##   industry = col_character(),
##   yearsExperience = col_double(),
##   milesFromMetropolis = col_double()
## )
```

```
jobType_stats<-test_features%>%group_by(jobType)%>%summarise(avg.salary_jobtype=mean(yearsExperience),median.salary_jobtype=median(yearsExperience),min.salary_jobtype=min(yearsExperience),max.salary_jobtype=max(yearsExperience))

industry_stats<-test_features%>%group_by(industry)%>%summarise(avg.salary_industry=mean(yearsExperience),median.salary_industry=median(yearsExperience),min.salary_industry=min(yearsExperience),max.salary_industry=max(yearsExperience))

degree_stats<-test_features%>%group_by(degree)%>%summarise(avg.salary_degree=mean(yearsExperience),median.salary_degree=median(yearsExperience),min.salary_degree=min(yearsExperience),max.salary_degree=max(yearsExperience))

major_stats<-test_features%>%group_by(major)%>%summarise(avg.salary_major=mean(yearsExperience),median.salary_major=median(yearsExperience),min.salary_major=min(yearsExperience),max.salary_major=max(yearsExperience))


mean_jobtype<-merge(jobType_stats,test_features[,c(1,3)])
mean2_jobtype<-merge(industry_stats,test_features[,c(1,6)])
mean3_jobtype<-merge(degree_stats,test_features[,c(1,4)])
mean4_jobtype<-merge(major_stats,test_features[,c(1,5)])


train_all_new<-merge(test_features,mean_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean2_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean3_jobtype,by="jobId")
train_all_new<-merge(train_all_new,mean4_jobtype,by="jobId")
final_data_<-train_all_new[,c(-1:-6)]
names(final_data_)[3]<-"jobType"
names(final_data_)[8]<-"industry"
names(final_data_)[13]<-"degree"
names(final_data_)[18]<-"major"
dummy_vars <- dummyVars(~ jobType +  degree + major+industry, data = final_data_)
one_hot_encode_category_vars <- as.data.frame(predict(dummy_vars, newdata = final_data_))
test_data_scored<-cbind(one_hot_encode_category_vars,final_data_[,c(-3,-8,-13,-18)])
names(test_data_scored) <- gsub("[.]", "", names(test_data_scored))


final_data_$predicted_salary<- predict(xgb_final, newdata = as.matrix(test_data_scored))
write.csv(final_data_,"test_scored.csv")
```