

```
####Setting the directory and loading the dataset into R, verifying that dataset is loaded correctly
library(readr)
X338_cert_proj_datasets_v3_0 <- read_csv("C:/Users/bvkka/Desktop/edureka/338_cert_proj_datasets_v3.0.csv")
View(X338_cert_proj_datasets_v3_0)
HR_Management<-X338_cert_proj_datasets_v3_0

str(HR_Management)

###Making some changes to the columns
library(plyr)
HR_Management$salary<- revalue(HR_Management$salary,c("low"=0))
HR_Management$salary<- revalue(HR_Management$salary,c("medium"=1))
HR_Management$salary<- revalue(HR_Management$salary,c("high"=2))

HR_Management$department<- revalue(HR_Management$department,c("hr"=0))
HR_Management$department<- revalue(HR_Management$department,c("IT"=1))
HR_Management$department<- revalue(HR_Management$department,c("management"=2))
HR_Management$department<- revalue(HR_Management$department,c("marketing"=3))
HR_Management$department<- revalue(HR_Management$department,c("product_mng"=4))
HR_Management$department<- revalue(HR_Management$department,c("RandD"=5))
HR_Management$department<- revalue(HR_Management$department,c("sales"=6))
HR_Management$department<- revalue(HR_Management$department,c("support"=7))
HR_Management$department<- revalue(HR_Management$department,c("technical"=8))
HR_Management$department<- revalue(HR_Management$department,c("accounting"=9))

str(HR_Management)

HR_Management$left<-as.factor(as.integer(HR_Management$left))
HR_Management$salary<-as.factor(as.integer(HR_Management$salary))
HR_Management$department<-as.factor(as.integer(HR_Management$department))

str(HR_Management)

###splitting the data into train and test sets
library(caret)
set.seed(12345)
```

```

di <- sample(2, nrow(HR_Management), prob = c(0.7,0.3), replace = TRUE)

train <- HR_Management[di==1,]
test <- HR_Management[di==2,]

###using ggplots for visualizations

library(ggplot2)
ggplot(train,aes(left,fill=left))+geom_bar()
prop.table(table(train$left)) #Percentage of left
prop.table(table(train$satisfaction_level))

#Let us look at each variable and see its influence on the churn of the organization

library(ggplot2)
library(grid)
library(gridExtra)
promotion_last_5yearsPlot <- ggplot(train,aes(promotion_last_5years,fill=left))+geom_density()+facet_grid(~left)
time_spend_companyPlot <- ggplot(train,aes(time_spend_company,fill=left))+geom_bar()
salaryPlot <- ggplot(train,aes(salary,fill=left))+geom_point(size=4,alpha = 0.05)
depPlot <- ggplot(train,aes(department,fill = left))+geom_bar()
grid.arrange(promotion_last_5yearsPlot,time_spend_companyPlot,salaryPlot,depPlot,ncol=2,top = "Fig 1")

satisfaction_levelPlot <- ggplot(train,aes(satisfaction_level,fill=left))+geom_bar()
last_evaluationPlot <- ggplot(train,aes(last_evaluation,fill=left))+geom_bar()
number_projectPlot <- ggplot(train,aes(number_project,fill=left))+geom_bar()
average_monthly_hoursPlot <- ggplot(train,aes(average_monthly_hours,fill=left))+geom_bar()
Work_accidentPlot <- ggplot(train,aes(Work_accident,fill=left))+geom_bar()
grid.arrange(satisfaction_levelPlot,last_evaluationPlot,number_projectPlot,average_monthly_hoursPlot,Work_accidentPlot,ncol=2,top = "Fig 2")

###Binning of Variables
##creating average monthly bins
max(train$average_monthly_hours)
min(train$average_monthly_hours)

```

```

train$average_monthly_hoursGroup<-
with(train,ifelse(average_monthly_hours>300,7,ifelse(average_monthly_hours>250,6,ifelse(average_monthly_hours>200,5,if
else(average_monthly_hours>150,4,ifelse(average_monthly_hours>100,3,ifelse(average_monthly_hours>50,2,1)))))))
test$average_monthly_hoursGroup<-
with(test,ifelse(average_monthly_hours>300,7,ifelse(average_monthly_hours>250,6,ifelse(average_monthly_hours>200,5,if
else(average_monthly_hours>150,4,ifelse(average_monthly_hours>100,3,ifelse(average_monthly_hours>50,2,1)))))))

##creating satisfying level bins
max(train$satisfaction_level)
min(train$satisfaction_level)
train$satisfaction_levelGroup<-
with(train,ifelse(satisfaction_level>0.9,9,ifelse(satisfaction_level>0.8,8,ifelse(satisfaction_level>0.7,7,ifelse(
satisfaction_level>0.6,6,ifelse(satisfaction_level>0.5,5,ifelse(satisfaction_level>0.4,4,ifelse(satisfaction_level
>0.3,3,ifelse(satisfaction_level>0.2,2,ifelse(satisfaction_level>0,1))))))))))
test$satisfaction_levelGroup<-
with(test,ifelse(satisfaction_level>0.9,9,ifelse(satisfaction_level>0.8,8,ifelse(satisfaction_level>0.7,7,ifelse(s
atisfaction_level>0.6,6,ifelse(satisfaction_level>0.5,5,ifelse(satisfaction_level>0.4,4,ifelse(satisfaction_level>
0.3,3,ifelse(satisfaction_level>0.2,2,ifelse(satisfaction_level>0,1))))))))))

##creating timespendatcompany level bins
max(train$time_spend_company)
min(train$time_spend_company)
train$time_spend_companyGroup<-
with(train,ifelse(time_spend_company>9,9,ifelse(time_spend_company>8,8,ifelse(time_spend_company>7,7,ifelse(time_s
pend_company>6,6,ifelse(time_spend_company>5,5,ifelse(time_spend_company>4,4,ifelse(time_spend_company>3,3,ifelse(
time_spend_company>2,2,ifelse(time_spend_company>0,1))))))))))
test$time_spend_companyGroup<-
with(test,ifelse(time_spend_company>9,9,ifelse(time_spend_company>8,8,ifelse(time_spend_company>7,7,ifelse(time_sp
end_company>6,6,ifelse(time_spend_company>5,5,ifelse(time_spend_company>4,4,ifelse(time_spend_company>3,3,ifelse(t
ime_spend_company>2,2,ifelse(time_spend_company>0,1))))))))))

###Correlation of Variables
library(corrplot)
library(psych)

#first make all the variables to numeric or integer to correlate
train$left<-as.numeric(as.factor(train$left))
train$department<-as.numeric(as.factor(train$department))
train$salary<-as.numeric(as.factor(train$salary))

```

```
train$promotion_last_5years<-as.numeric(as.integer(train$promotion_last_5years))
train$Work_accident<-as.numeric(as.integer(train$Work_accident))
train$time_spend_company<-as.numeric(as.integer(train$time_spend_company))
train$average_monthly_hours<-as.numeric(as.integer(train$average_monthly_hours))
train$number_project<-as.numeric(as.integer(train$number_project))
```

```
test$department<-as.numeric(as.factor(test$department))
test$salary<-as.numeric(as.factor(test$salary))
```

```
cor(train) #correlation values between variables
```

```
corrplot(cor(train),method = "circle")
```

```
##applying Logistic model to see which variables are more significant in churning out employers
```

```
model1<-glm(formula = left ~ ., binomial(link="logit"),data =train)
```

```
summary(model1)
```

```
library(MASS)
```

```
exp(cbind(OR=coef(model1),confint(model1)))
```

```
###Building models using Decision Tree, Random Forest ,NB and SVM techniques
```

```
library(caret)
```

```
library(rpart)
```

```
library(ROCR)
```

```
#####Decision Tree#####
```

```
DTree_model <-
```

```
rpart(left~satisfaction_levelGroup+last_evaluation+number_project+average_monthly_hours+time_spend_company+Work_accident, data = train)
```

```

par(mar = rep(2, 4))

plot(DTree_model, margin = 0.1)
text(DTree_model, use.n = TRUE, pretty = TRUE, cex = 0.6)
table(train$left)

pred_DTree_model <- predict (DTree_model, newdata = test)
pred_DTree_model
cm_DTree<-table(pred_DTree_model,test$left)
cm_DTree
accuracy_DTree<-(cm_DTree[1]+cm_DTree[4])/(cm_DTree[1]+cm_DTree[2]+cm_DTree[3]+cm_DTree[4])
accuracy_DTree

confusionMatrix(table(pred_tree, test$left))

```

####Random Forest####

```

library(randomForest)
RF_model <- randomForest(left~., data = train)
RF_model

pred_RF_model <- predict (RF_model, newdata = test, type = "class")

pred_RF_model
cm_RF<-table(pred_RF_model,test$left)
cm_RF
accuracy_RF<-(cm_RF[1]+cm_RF[4])/(cm_RF[1]+cm_RF[2]+cm_RF[3]+cm_RF[4])
accuracy_RF

```

#####Naive Bayes#####

```

library(e1071)

```

```

NB_model <-
naiveBayes(left~salary+time_spend_companyGroup+satisfaction_levelGroup+department+average_monthly_hoursGroup, data
= train,laplace = laplace)

NB_model

pred_NB_model <- predict (NB_model, newdata = test)

pred_NB_model
cm_NB<-table(pred_NB_model,test$left)
cm
accuracy_NB<-(cm[1]+cm[4])/(cm[1]+cm[2]+cm[3]+cm[4])
accuracy_NB

#####SVM#####
library(e1071)

SVM_model <- svm(left~.,data = train,type="C-classification", kernel = "radial", cost = 0.1,
gamma=c(.5,1,2))

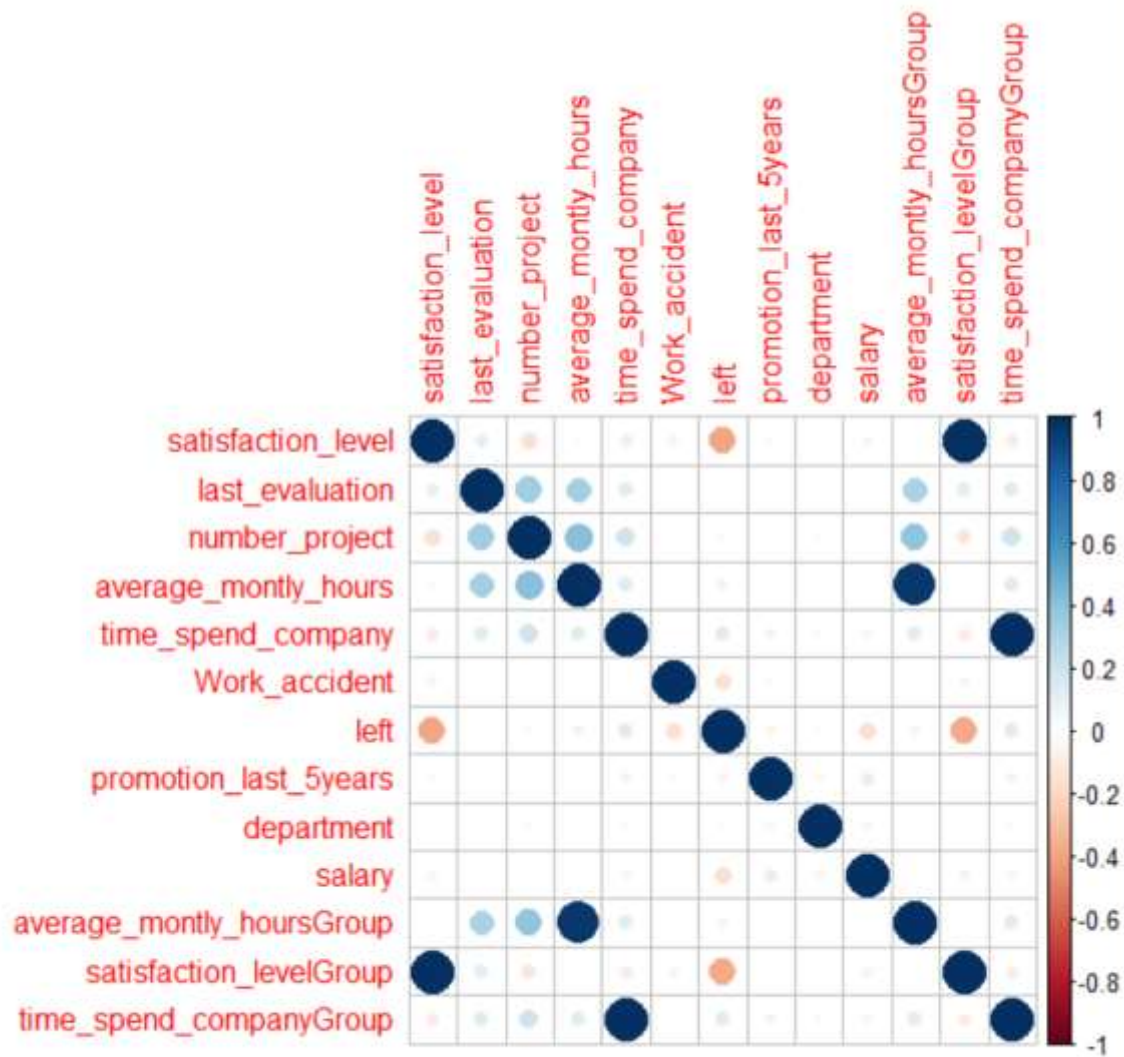
pred_svm <- predict(SVM_model,test, type = "class")
cm_svm<-table(pred_svm,test$left)
cm
accuracy_svm<-(cm[1]+cm[4])/(cm[1]+cm[2]+cm[3]+cm[4])
accuracy_svm

```



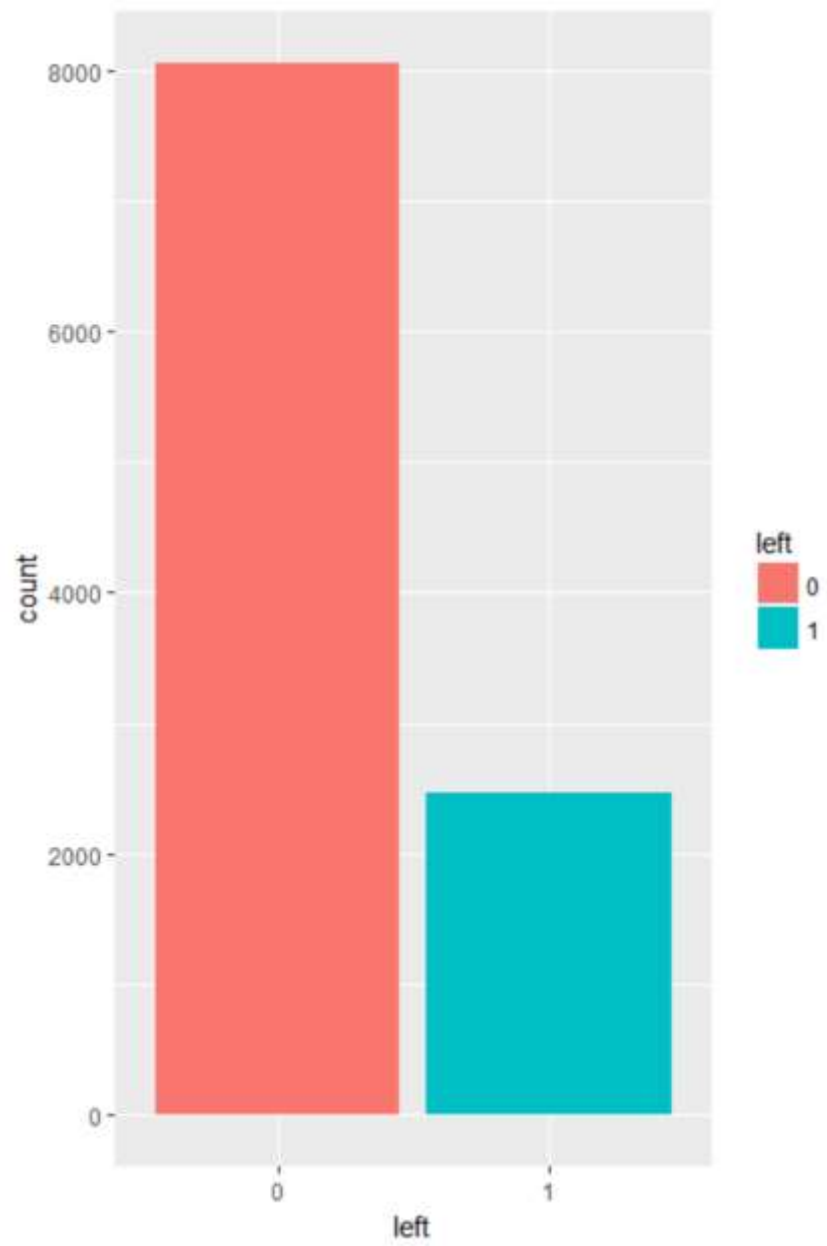
```
summary(train)
satisfaction_level last_evaluation number_project average_monthly_hours
Min. :0.0900 Min. :0.360 Min. :2.000 Min. : 96.0
1st Qu.:0.4400 1st Qu.:0.560 1st Qu.:3.000 1st Qu.:156.0
Median :0.6400 Median :0.720 Median :4.000 Median :200.0
Mean :0.6118 Mean :0.715 Mean :3.804 Mean :201.1
3rd Qu.:0.8100 3rd Qu.:0.870 3rd Qu.:5.000 3rd Qu.:245.0
Max. :1.0000 Max. :1.000 Max. :7.000 Max. :310.0
time_spend_company Work_accident left promotion_last_5years
Min. : 2.0 Min. :0.000 0:8056 Min. :0.00000
1st Qu.: 3.0 1st Qu.:0.000 1:2468 1st Qu.:0.00000
Median : 3.0 Median :0.000 Median :0.00000
Mean : 3.5 Mean :0.151 Mean :0.02147
3rd Qu.: 4.0 3rd Qu.:0.000 3rd Qu.:0.00000
Max. :10.0 Max. :1.000 Max. :1.00000
department salary average_monthly_hoursGroup
Length:10524 Length:10524 Min. :2.000
Class :character Class :character 1st Qu.:4.000
Mode :character Mode :character Median :4.000
Mean :4.515
3rd Qu.:5.000
Max. :7.000
satisfaction_levelGroup time_spend_companyGroup
Min. :1.000 Min. :1.0
1st Qu.:4.000 1st Qu.:2.0
Median :6.000 Median :2.0
Mean :5.591 Mean :2.5
3rd Qu.:8.000 3rd Qu.:3.0
Max. :9.000 Max. :9.0
```





```
> cor(train)
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years
satisfaction_level	1.00000000	0.100001484	-0.152795521	-0.026804048	-0.098831159	5.877844e-02	-0.394190400	0.024038008
last_evaluation	0.10000148	1.000000000	0.349986791	0.330304441	0.135179964	-8.813945e-03	0.004748024	-0.008737904
number_project	-0.15279552	0.349986791	1.000000000	0.411474375	0.196549960	-9.431695e-03	0.030257097	-0.004647329
average_monthly_hours	-0.02680405	0.330304441	0.411474375	1.000000000	0.131026059	-1.388856e-02	0.071158403	-0.003889905
time_spend_company	-0.09883116	0.135179964	0.196549960	0.131026059	1.000000000	-1.456606e-03	0.137620284	0.068879031
Work_accident	0.05877844	-0.008813945	-0.009431695	-0.013888556	-0.001456606	1.000000e+00	-0.155742958	0.036390198
left	-0.39419048	0.004748024	0.030257097	0.071158403	0.137620284	-1.557430e-01	1.000000000	-0.060336240
promotion_last_5years	0.02403801	-0.008737904	-0.004647329	-0.003889905	0.068879031	3.639020e-02	-0.060336240	1.000000000
department	-0.01364294	0.009874083	0.029448950	0.001366705	-0.034399957	6.568120e-05	0.028665569	-0.039917909
salary	0.05370382	-0.004641392	-0.007444476	-0.005370706	0.041718790	1.150883e-02	-0.166254775	0.106466788
average_monthly_hoursGroup	-0.01718891	0.317411487	0.390931909	0.961084820	0.127942423	-1.019747e-02	0.058594583	-0.005117105
satisfaction_levelGroup	0.99244116	0.115030423	-0.130105045	-0.006815489	-0.095253682	5.787055e-02	-0.378168742	0.022814759
time_spend_companyGroup	-0.09883116	0.135179964	0.196549960	0.131026059	1.000000000	-1.456606e-03	0.137620284	0.068879031
department	department	salary	average_monthly_hoursGroup	satisfaction_levelGroup	time_spend_companyGroup			
satisfaction_level	-1.364294e-02	0.053703817	-0.017188915	0.992441161	-0.098831159			
last_evaluation	9.874083e-03	-0.004641392	0.317411487	0.115030423	0.135179964			
number_project	2.944895e-02	-0.007444476	0.390931909	-0.130105045	0.196549960			
average_monthly_hours	1.366705e-03	-0.005370706	0.961084820	-0.006815489	0.131026059			
time_spend_company	-3.439996e-02	0.041718790	0.127942423	-0.095253682	1.000000000			
Work_accident	6.568128e-05	0.011508831	-0.010197467	0.057870550	-0.001456606			
left	2.866557e-02	-0.166254775	0.058594583	-0.378168742	0.137620284			
promotion_last_5years	-3.991791e-02	0.106466788	-0.005117105	0.022814759	0.068879031			
department	1.000000e+00	-0.054699314	0.004476596	-0.013390906	-0.034399957			
salary	-5.469931e-02	1.000000000	0.001460412	0.051095363	0.041718790			
average_monthly_hoursGroup	4.476596e-03	0.001460412	1.000000000	0.001050063	0.127942423			
satisfaction_levelGroup	-1.339091e-02	0.051095363	0.001050063	1.000000000	-0.095253682			
time_spend_companyGroup	-3.439996e-02	0.041718790	0.127942423	-0.095253682	1.000000000			



	0	1
	0.7654884	0.2345116

>

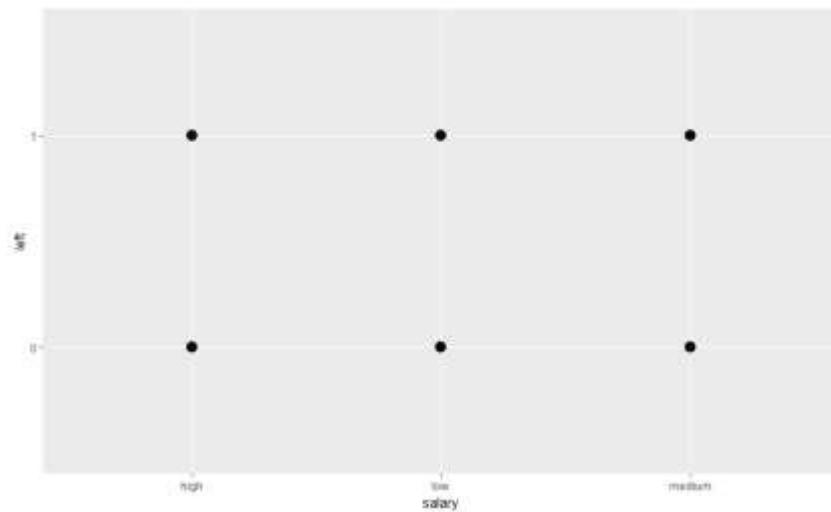
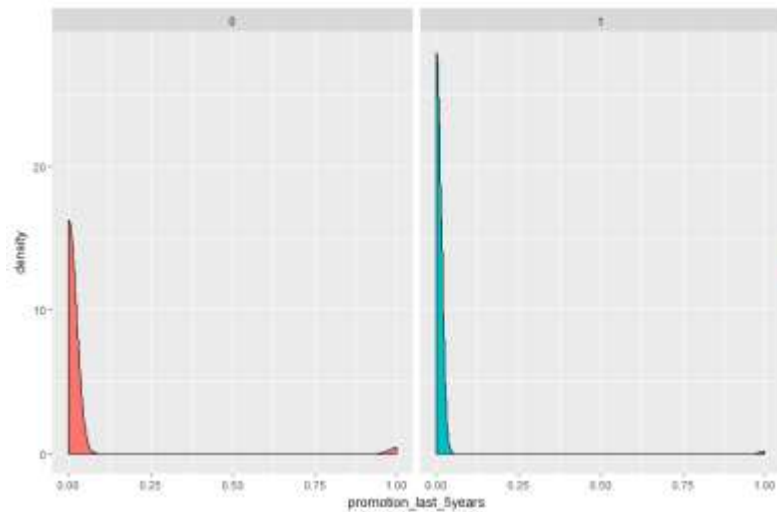
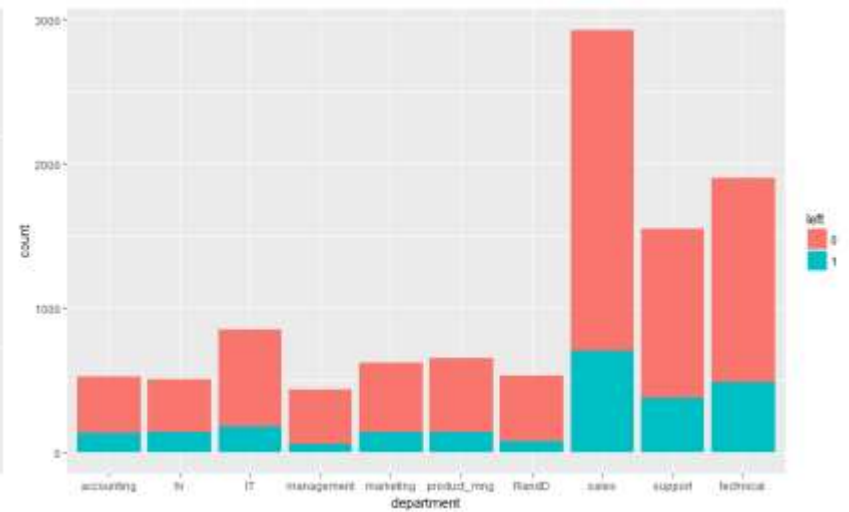
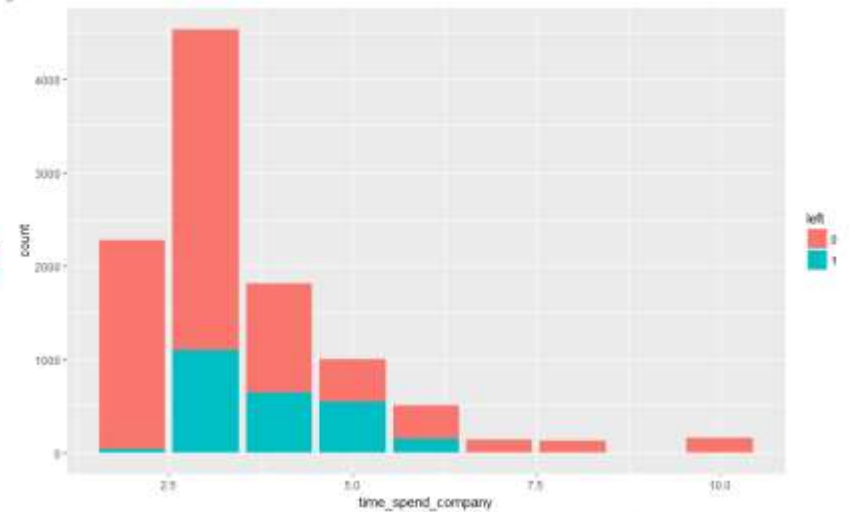


Fig 1



Promotion: Larger proportion of people who have been promoted recently have quit the organization.

Time\_spend\_company: Larger proportion of new comers are quitting the organization which sidelines the recruitment efforts of the organization.

Salary: We are not able to see any distinguishable feature here

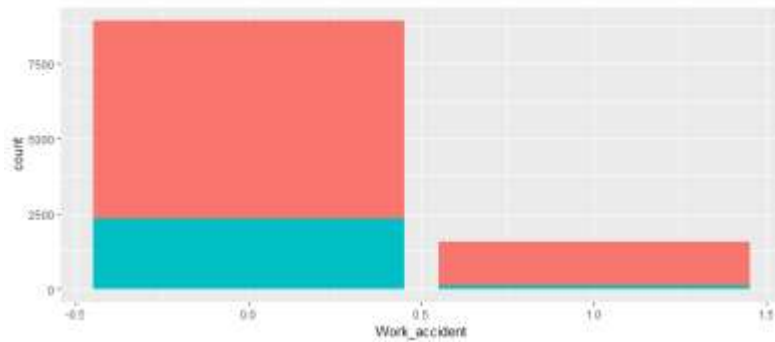
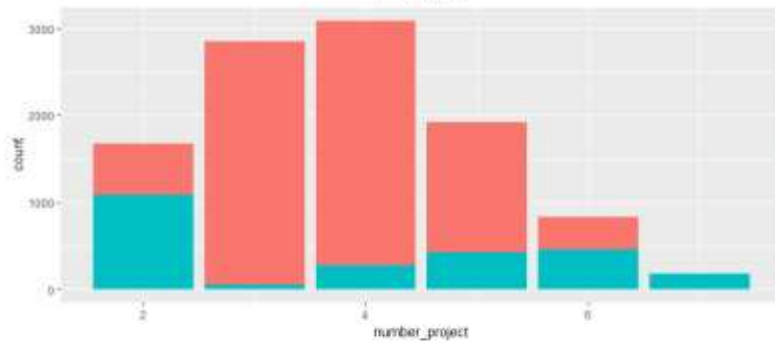
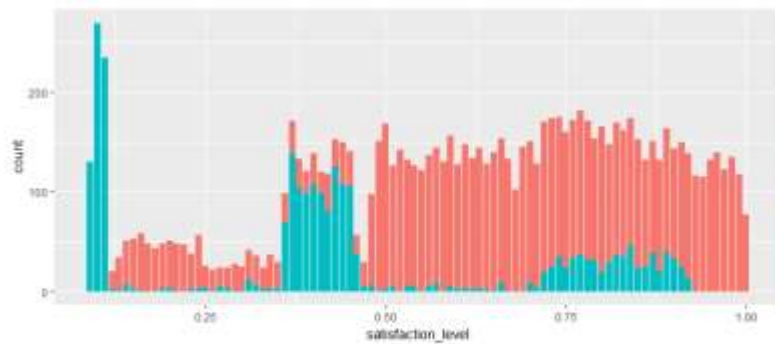
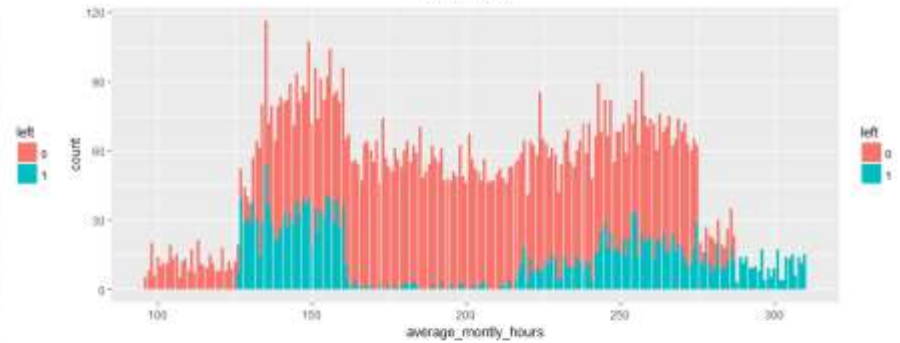
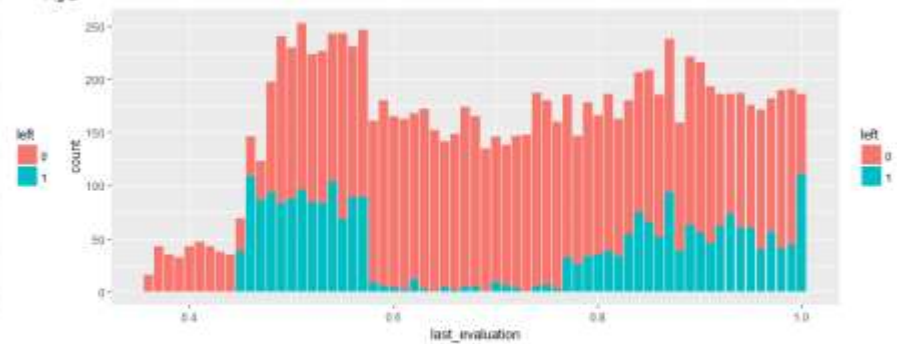


Fig 2



```
Call:
glm(formula = left ~ ., family = binomial(link = "logit"), data = train)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1994	-0.6468	-0.3857	-0.1177	3.1549

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-0.641264	0.250220	-2.563	0.010383	*
satisfaction_level	-14.175698	0.913331	-15.521	< 2e-16	***
last_evaluation	0.617718	0.181328	3.407	0.000658	***
number_project	-0.347432	0.026245	-13.238	< 2e-16	***
average_monthly_hours	0.006815	0.001945	3.503	0.000460	***
time_spend_company	0.260763	0.018706	13.940	< 2e-16	***
Work_accident	-1.513420	0.105344	-14.366	< 2e-16	***
promotion_last_5years	-1.360504	0.306129	-4.444	8.82e-06	***
departmenthr	0.090607	0.160443	0.565	0.572256	
departmentIT	-0.255475	0.149594	-1.708	0.087676	.
departmentmanagement	-0.457203	0.195371	-2.340	0.019274	*
departmentmarketing	-0.122724	0.159199	-0.771	0.440775	
departmentproduct_mng	-0.183987	0.157303	-1.170	0.242148	
departmentRandD	-0.682667	0.178227	-3.830	0.000128	***
departmentsales	-0.101533	0.124859	-0.813	0.416112	
departmentsupport	0.002423	0.133301	0.018	0.985499	
departmenttechnical	0.003665	0.129918	0.028	0.977498	
salarylow	2.057056	0.162425	12.665	< 2e-16	***
salarymedium	1.479659	0.163326	9.060	< 2e-16	***
average_monthly_hoursGroup	-0.162548	0.089333	-1.820	0.068825	.
satisfaction_levelGroup	1.023163	0.092337	11.081	< 2e-16	***
time_spend_companyGroup	NA	NA	NA	NA	

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)



```
> exp(cbind(OR=coef(model1),confint(model1)))
```

```
Waiting for profiling to be done...
```

	OR	2.5 %	97.5 %
(Intercept)	5.266264e-01	3.207391e-01	8.559507e-01
satisfaction_level	6.975457e-07	1.157045e-07	4.153592e-06
last_evaluation	1.854690e+00	1.300424e+00	2.647453e+00
number_project	7.065001e-01	6.709135e-01	7.436245e-01
average_monthly_hours	1.006839e+00	1.003010e+00	1.010690e+00
time_spend_company	1.297920e+00	1.251179e+00	1.346403e+00
Work_accident	2.201558e-01	1.783337e-01	2.695925e-01
promotion_last_5years	2.565313e-01	1.351568e-01	4.522753e-01
departmenthr	1.094839e+00	7.994243e-01	1.499818e+00
departmentIT	7.745486e-01	5.779043e-01	1.039058e+00
departmentmanagement	6.330521e-01	4.299490e-01	9.254226e-01
departmentmarketing	8.845077e-01	6.472932e-01	1.208514e+00
departmentproduct_mng	8.319464e-01	6.111184e-01	1.132523e+00
departmentRandD	5.052677e-01	3.553450e-01	7.149887e-01
departmentsales	9.034511e-01	7.085788e-01	1.156260e+00
departmentsupport	1.002426e+00	7.730331e-01	1.303862e+00
departmenttechnical	1.003671e+00	7.792583e-01	1.297047e+00
salarylow	7.822909e+00	5.745844e+00	1.087467e+01
salarymedium	4.391450e+00	3.218966e+00	6.113909e+00
average_monthly_hoursGroup	8.499753e-01	7.133831e-01	1.012569e+00
satisfaction_levelGroup	2.781979e+00	2.322393e+00	3.335442e+00
time_spend_companyGroup	NA	NA	NA

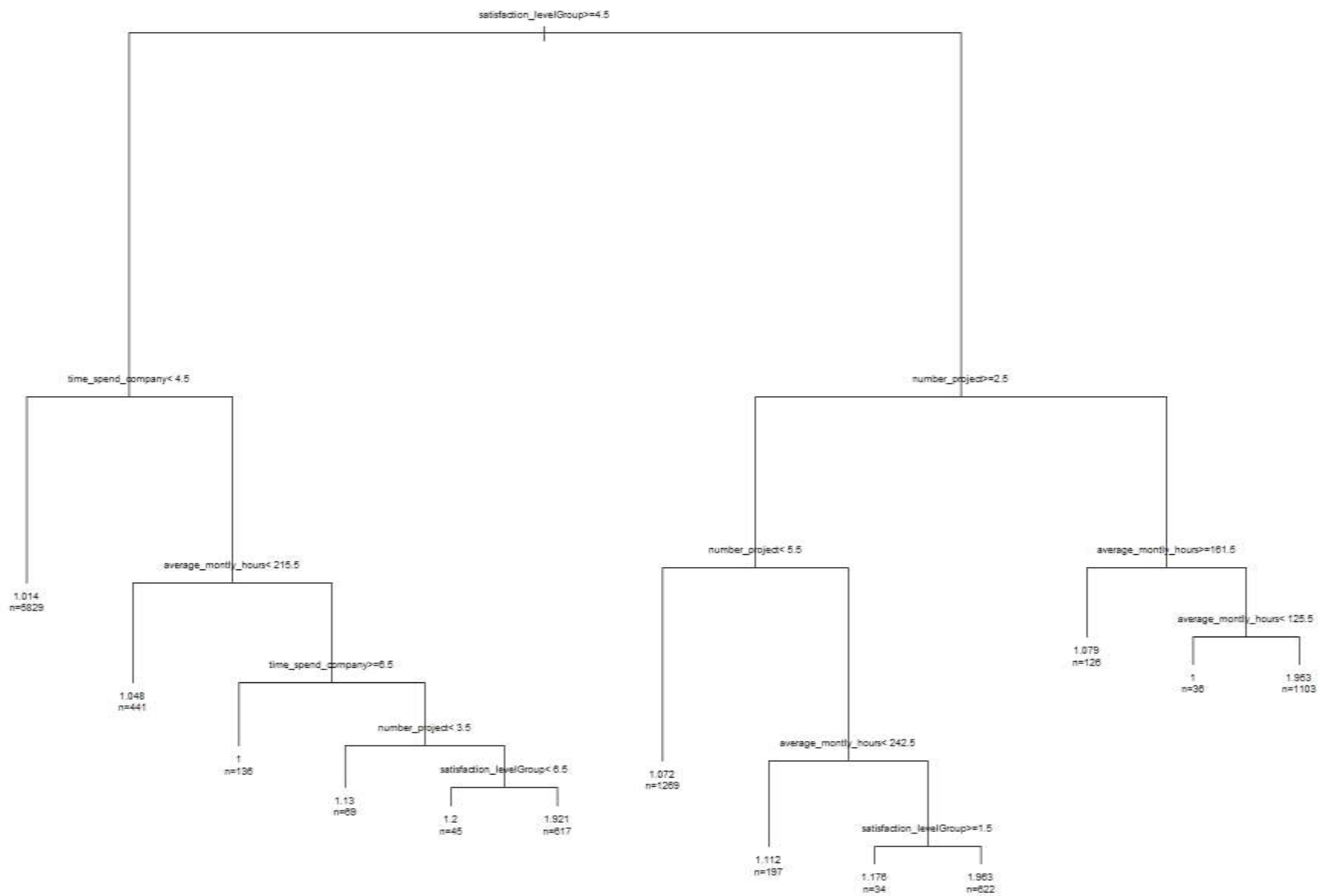
```
>
```

## Decision Tree

```
> DTree_model <- rpart(left~., data = train)
> DTree_model
n= 10524

node), split, n, deviance, yval
* denotes terminal node

1) root 10524 1889.225000 1.234512
 2) satisfaction_levelGroup>=4.5 7137 622.484500 1.096539
   4) time_spend_company< 4.5 5829 80.846460 1.014068 *
   5) time_spend_company>=4.5 1308 325.311200 1.464067
    10) average_monthly_hours< 215.5 441 20.000000 1.047619 *
    11) average_monthly_hours>=215.5 867 189.926200 1.675894
      22) time_spend_company>=6.5 136 0.000000 1.000000 *
      23) time_spend_company< 6.5 731 116.238000 1.801642
        46) number_project< 3.5 69 7.826087 1.130435 *
        47) number_project>=3.5 662 74.086100 1.871601
          94) satisfaction_levelGroup< 6.5 45 7.200000 1.200000 *
          95) satisfaction_levelGroup>=6.5 617 45.108590 1.920583 *
 3) satisfaction_levelGroup< 4.5 3387 844.591700 1.525244
   6) number_project>=2.5 2122 475.057500 1.338360
    12) number_project< 5.5 1269 84.474390 1.071710 *
    13) number_project>=5.5 853 166.121900 1.735053
      26) average_monthly_hours< 242.5 197 19.543150 1.111675 *
      27) average_monthly_hours>=242.5 656 47.035060 1.922256
        54) satisfaction_levelGroup>=1.5 34 4.941176 1.176471 *
        55) satisfaction_levelGroup< 1.5 622 22.149520 1.963023 *
  7) number_project< 2.5 1265 171.102000 1.838735
    14) average_monthly_hours>=161.5 126 9.206349 1.079365 *
    15) average_monthly_hours< 161.5 1139 81.201050 1.922739
      30) average_monthly_hours< 125.5 36 0.000000 1.000000 *
      31) average_monthly_hours>=125.5 1103 49.548500 1.952856 *
```



```

> cm_DTree<-table(pred_DTree_model,test$left)
> cm_DTree

pred_DTree_model      0      1
1                12      0
1.01406759306914 2438    30
1.04646464646465  188     7
1.0546875         41     4
1.07171000788022 487    39
1.07692307692308  60     6
1.09090909090909  17     2
1.11167512690355  68    12
1.15151515151515   9     7
1.91826923076923  20   289
1.96065873741995  20   467
1.96308186195827  12   240
> accuracy_DTree<-(cm_DTree[1]+cm_DTree[4])/(cm_DTree[1]+cm_DTree[2]+cm_DTree[3]+cm_DTree[4])
> accuracy_DTree

```

*Decision Tree Accuracy 0.780834914611006*

## Random Forest

```
> RF_model
```

```
Call:
```

```
randomForest(formula = left ~ ., data = train)  
Type of random forest: regression  
Number of trees: 500
```

```
No. of variables tried at each split: 4
```

```
Mean of squared residuals: 0.01225822  
% Var explained: 93.17
```

```
> accuracy_RF<-(cm_RF[1]+cm_RF[4])/(cm_RF[1]+cm_RF[2]+cm_RF[3]+cm_RF[4])
```

```
> accuracy_RF
```

```
[1] 0.8978102
```

```
> |
```

# Naïve Bayes

```
> library(e1071)
> NB_model <- naiveBayes(left~, data = train)
>
> NB_model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      1      2
0.7654884 0.2345116

Conditional probabilities:
satisfaction_level
Y      [,1]      [,2]
1 0.6660348 0.2178236
2 0.4345089 0.2638863

last_evaluation
Y      [,1]      [,2]
1 0.7145022 0.1622166
2 0.7164182 0.1969855

number_project
Y      [,1]      [,2]
1 3.783515 0.9790132
2 1.871560 1.8298391

average_monthly_hours
Y      [,1]      [,2]
1 199.1574 45.76882
2 207.5596 61.58019

time_spent_company
Y      [,1]      [,2]
1 1.388814 1.5700884
2 3.864263 0.5734856

Work_accident
Y      [,1]      [,2]
1 0.18185284 0.3857465
2 0.05024311 0.3186985

promotion_last_5years
Y      [,1]      [,2]
1 0.026315789 0.16888263
2 0.005672609 0.07511882

department
Y      [,1]      [,2]
1 6.485612 2.491879
2 6.574959 2.535451

salary
Y      [,1]      [,2]
```

Accuracy : 0.81

95% CI : (0.7728, 0.8435)

No Information Rate : 0.53

P-Value [Acc > NIR] : <2e-16

Kappa : 0.6174

McNemar's Test P-Value : 0.2183

Sensitivity : 0.8453

Specificity : 0.7702

Pos Pred Value : 0.8058

Neg Pred Value : 0.8153

Prevalence : 0.5300

Detection Rate : 0.4480

Detection Prevalence : 0.5560

Balanced Accuracy : 0.8077

'Positive' Class : Neg

## Support Vector Machine

```
> summary(SVM_model)
```

Call:

```
svm(formula = left ~ ., data = train, type = "C-classification",  
     kernel = "radial", cost = 0.1, gamma = c(0.5, 1, 2))
```

Parameters:

```
  SVM-Type:  C-classification  
SVM-Kernel:  radial  
    cost:    0.1  
   gamma:    0.5 1 2
```

Number of Support Vectors: 2975

```
( 1041 1934 )
```

Number of Classes: 2

Levels:

```
1 2
```



```
> cm_svm<-table(pred_svm,test$left)
> cm

pred_svm      0      1
      1 3343  162
      2   29  941
> accuracy_svm<-(cm[1]+cm[4])/(cm[1]+cm[2]+cm[3]+cm[4])
> accuracy_svm
[1] 0.9573184
>
```

# Summary

From the above example, we can see that Decision Tree and Random Forest can be used for customer churn analysis for this dataset equally fine.

Throughout the analysis, I have learned several important things:

- Features such as satisfaction\_level, satisfaction\_levelGroup, last\_evaluation, number\_project, average\_monthly\_hours, time\_spend\_company, Work\_Accident appear to play a role in customer leaving.
- Out of all the models used, for the employees leaving/churning – the accuracy of SVM model is highest
- SVM>RF>NB>DT