Applied Data Science Spring 2017 Project5 Team4 Main Report

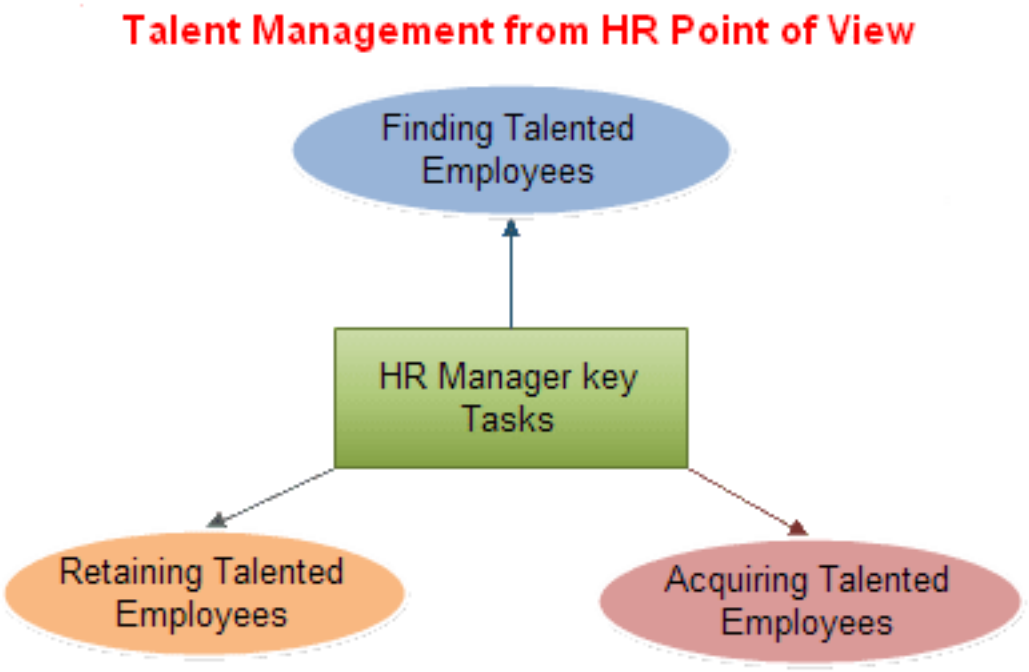
Yuan Mei, Yingxin Zhang, Kexin Nie, Xuanzi Xu, Senyao Han, He Zhu

April 25, 2017

Figure 1:

Introduction

The purpose of our project is to make a prediction engine of employees' future decision(Whether they will leave the company) for the company. It's crucial to the company because: first, human resource department need to use the number of staff quiting to determine the new year recruitment plan; second, they also need to find out the factors behind the staff losing to develop better company mechanic to retain the old talents.



Our project contains two parts: first is our shiny app(a handy tool for the company);second is our main documents showing the calculation and comparision between Cox Model(Survival Analysis), Decision Tree, Naive Bayes, Random Forest, and Logistic Regression.

The data our models based on is the human resource data from Kaggle. Our data set contains information for 14999 employees, which include their income, satifaction of the current company and whether have promotion, etc. The data set also provides their current status(left or not).

Variables	Value
satisfaction_level	[0,1]
last_evaluation	[0,1]
number_project	Nonnegative Integers
average_montly_hours	Nonnegative Integers
time_spend_company	Nonnegative Integers
Work_accident	0 or 1
left	0 or 1
promotion_last_5years	0 or 1
sales	Department
salary	low, medium, high

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: ggpubr

##
## Attaching package: 'survminer'

## The following object is masked from 'package:ggpubr':
##
##     theme_classic2

## The following object is masked from 'package:ggplot2':
##
##     %+%

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths

##
## Attaching package: 'ggvis'

## The following object is masked from 'package:ggplot2':
##
##     resolution
```

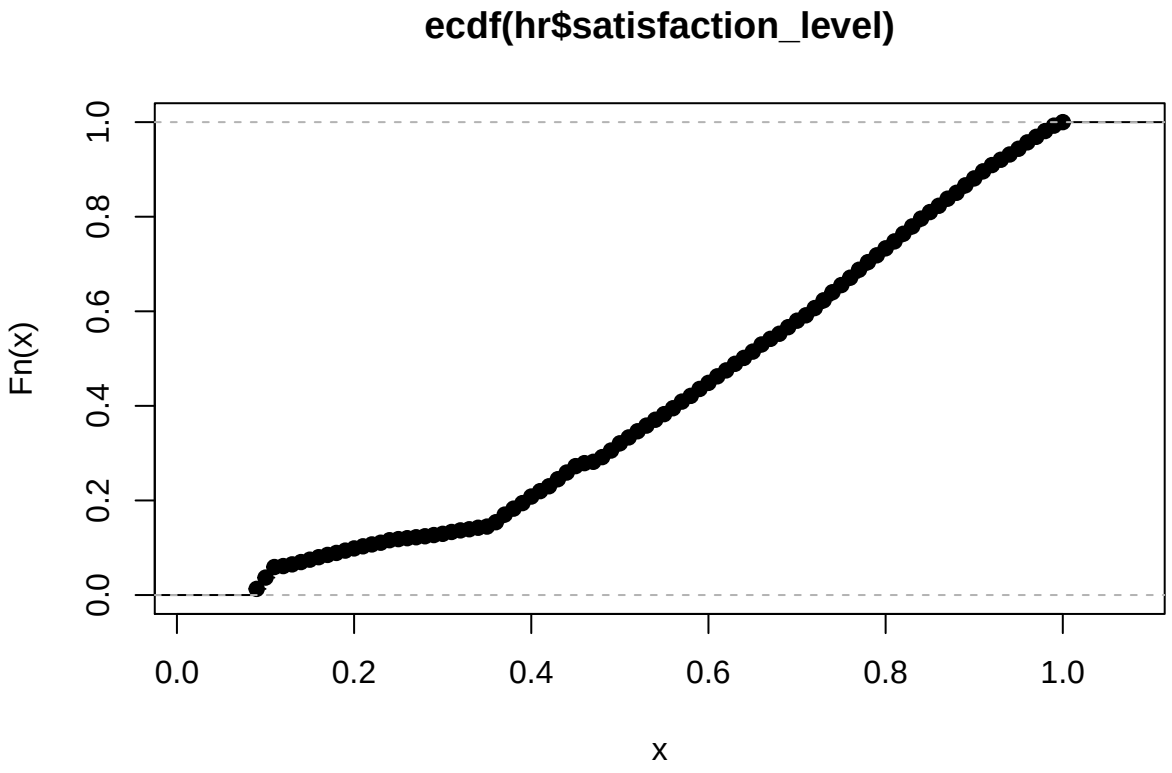
Step1 Processing Data

Import data

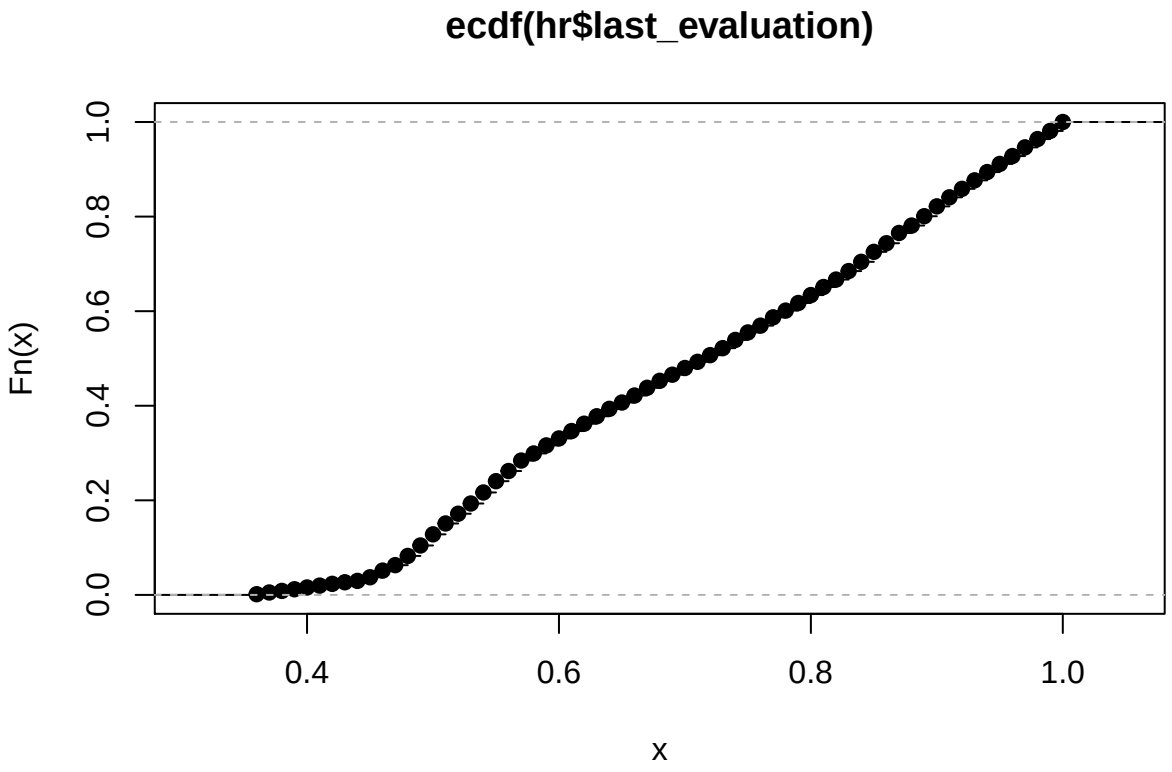
Exploratory data analysis

Since we need categorical data when using Survival analysis model, we split the continous vairables(satisfaction_level,last_evalua into) into different categories according to the distribution.As we can see in the plots, these three indicaters are nearly uniformly distributed.As for “satisfaction_level”,the staff tends to stay when satisfaction level<0.5;the staff tends to leave when satisfaction level>0.5; So we use quantile(0.5) to split the data.As for “last evaluation”, the staff’s behaviors can be grouped into three groups.“last evaluation” <0.6, 0.6<“last evaluation” <0.8,0.8 < last evaluation“.As for”average_montly_hours“, the staff’s behaviors can be grouped into three groups.”average_montly_hours“<160, 160<”last evaluation" < 240, 240 < last evaluation“.

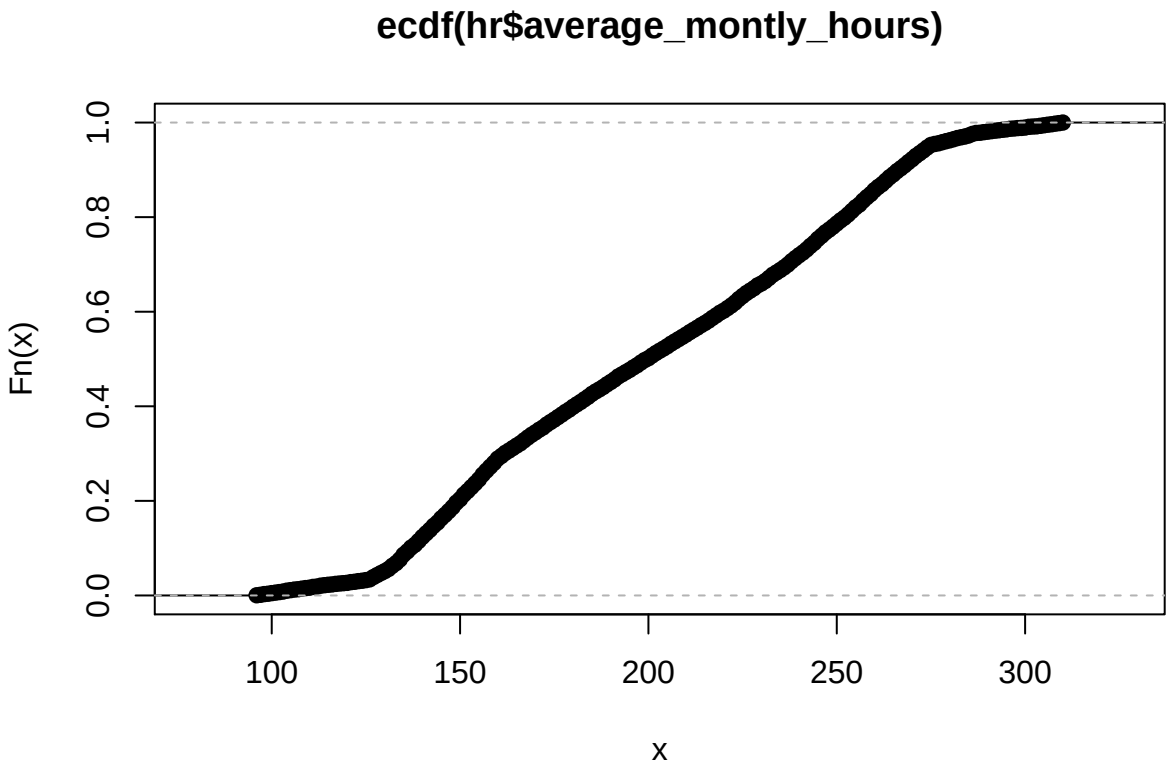
```
plot(ecdf(hr$satisfaction_level))
```




```
plot(ecdf(hr$last_evaluation))
```

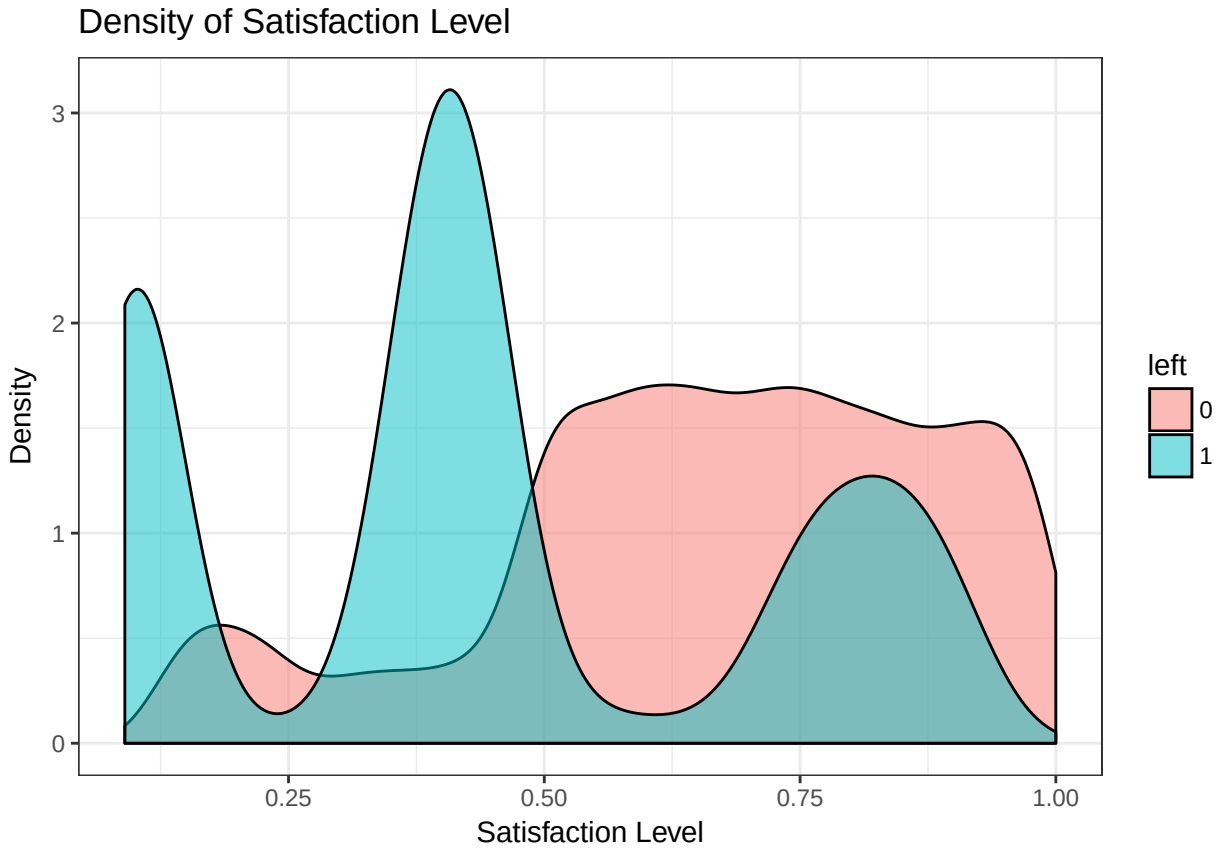



```
plot(ecdf(hr$average_montly_hours))
```

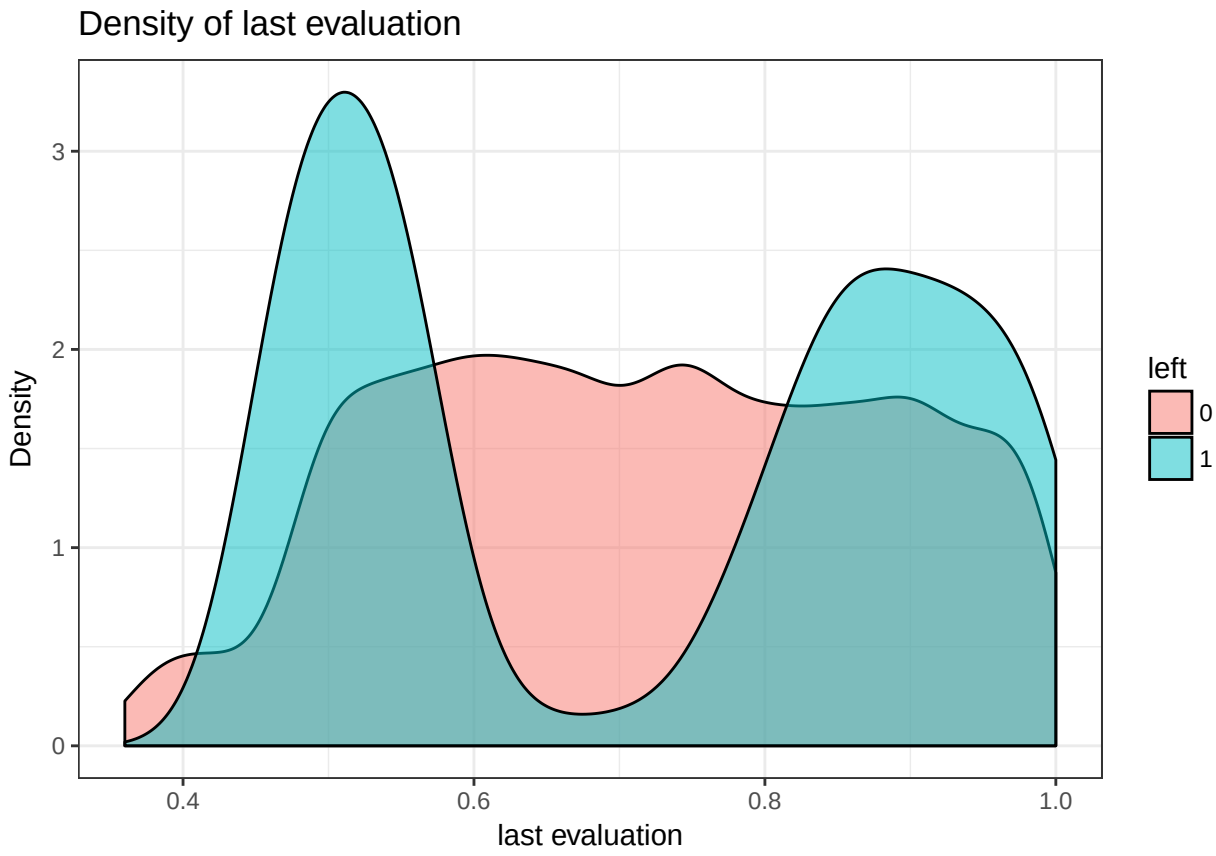


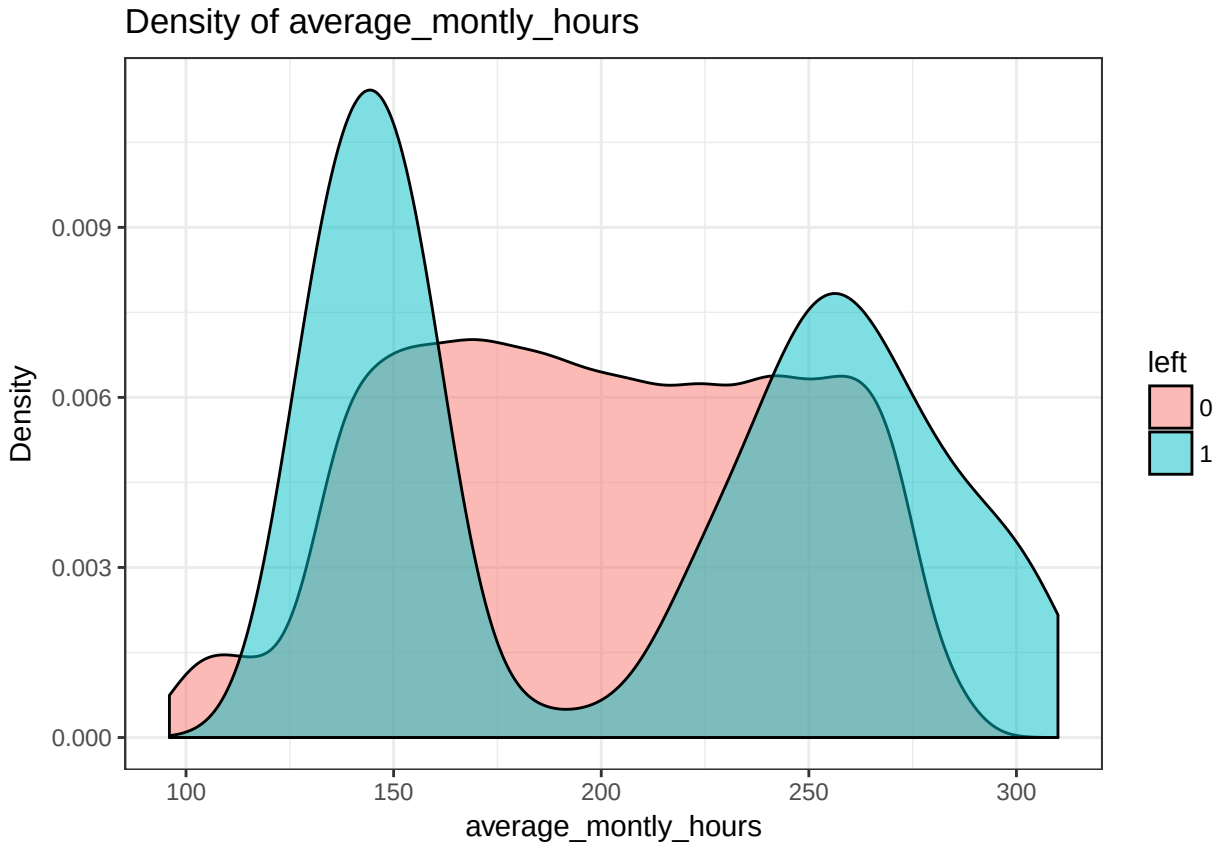

```
hr$left <- factor(hr$left)
ggplot(hr, aes(satisfaction_level)) +
```

```
  geom_density(aes(group = left, fill = left), alpha = 0.5) +
  theme_bw() + xlab("Satisfaction Level") + ylab("Density") +
  ggtitle("Density of Satisfaction Level")
```




```
ggplot(hr, aes(last_evaluation)) +
  geom_density(aes(group = left, fill = left), alpha = 0.5) +
  theme_bw() + xlab("last evaluation") + ylab("Density") +
  ggtitle("Density of last evaluation")
```


Density of last evaluation
Density
3
2
1
0
0.4
0.6
0.8
1.0
last evaluation
left
0
1


Density of average_montly_hours
Density
0.009
0.006
0.003
0.000
100
150
200
250
300
average_montly_hours
left
0
1

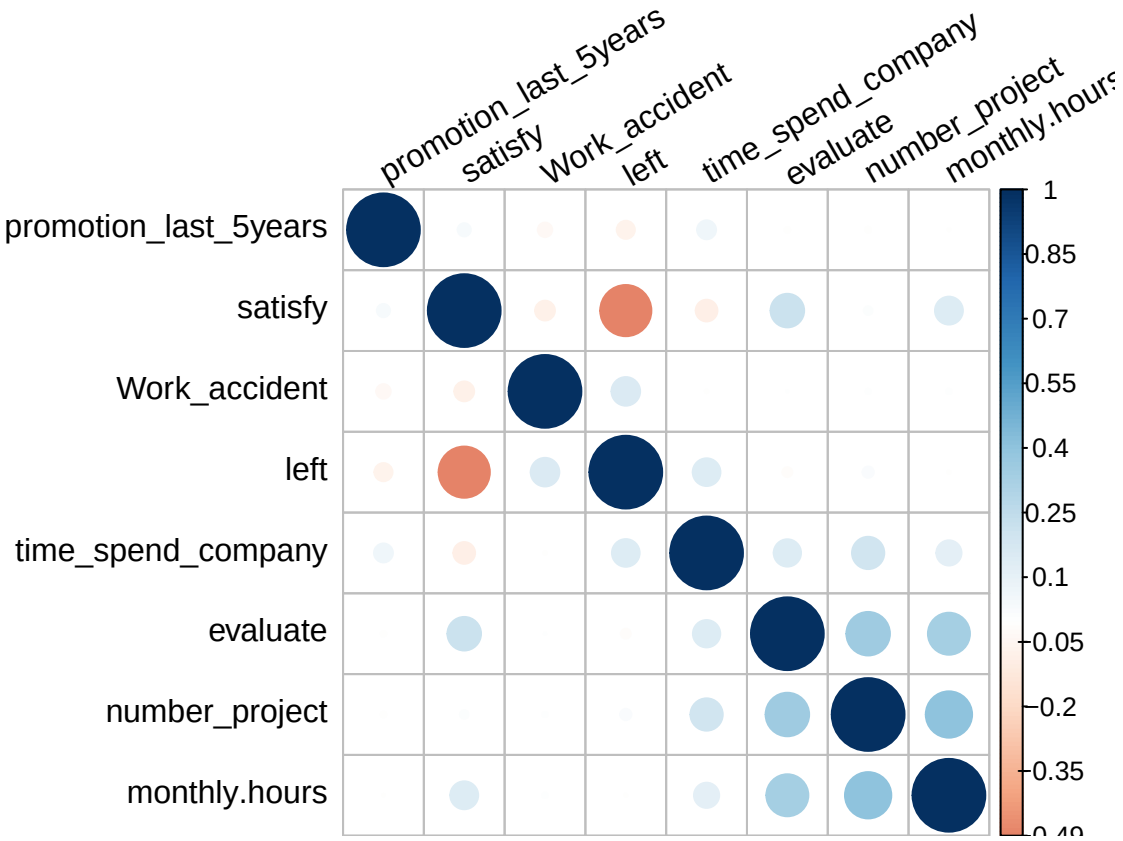
Generate Dataframe

```
#0 is not satisfied 1: very very happy
satisfy<-rep(0,nrow(hr))
satisfy[hr$satisfaction_level>= 0.5]<- 1
hr$satisfy<-satisfy

# 0 is low evaluation; 1 is medium evaluation; 2 is high evaluation.
evaluate<-rep(0,nrow(hr))
evaluate[hr$last_evaluation>= 0.6 & hr$last_evaluation<= 0.8]<- 1
evaluate[hr$last_evaluation > 0.8] <-2
hr$evaluate<-evaluate

# 0 is spending low average monthly hours; 1 is spending medium average monthly hours; 2 is spending hi
monthly.hours<-rep(0,nrow(hr))
monthly.hours[hr$average_montly_hours>= 160 & hr$average_montly_hours<= 240]<- 1
monthly.hours[hr$average_montly_hours > 240] <-2
hr$monthly.hours<-monthly.hours

# Work accident
hr$Work_accident<-ifelse(as.logical(hr$Work_accident), 0, 1)
```

Correlation plot

As we can see from the correlation plot, the variables “Satify”,“Promotion_last_5years”,“Work_accident” is highly correlated staff status “left”. So we choose these indicators as our feartures.

```
M<-hr[,c(-9,-10)]
M$left<-as.numeric(M$left)
M<-M[,c(-1,-2,-4)]
M<-cor(M)
corrplot( M ,is.corr = FALSE, type = "full", order = "hclust",
         tl.col = "black", tl.srt = 30)
```



Step 2 Survival Analysis

Due to the information provided in our data set, we decide to use a Cox Proportional Hazard Model to predict the future performance. Because the left column represents the current status of an employee, we also know the time they left and the employement time if they still work in the company. We believe the cox model will work will on this kind of data.

A brief Introduction of Cox Model

The purpose of the model is to evaluate simultaneously the effect of several factors on survival. In other words, it allows us to examine how specified factors influence the rate of a particular event happening (e.g., infection, death) at a particular point in time. This rate is commonly referred as the hazard rate. Predictor variables (or factors) are usually termed covariates in the survival-analysis literature.

The Cox model is expressed by the hazard function denoted by h(t). Briefly, the hazard function can be interpreted as the risk of dying at time t. It can be estimated as follow:

h(t)=h0(t)?exp(b1x1+b2x2+. . . +bpxp)

where,

t represents the survival time

h(t) is the hazard function determined by a set of p covariates (x1,x2,. . . ,xpx1,x2,. . . ,xp) the coefficients (b1,b2,. . . ,bpb1,b2,. . . ,bp) measure the impact (i.e., the effect size) of covariates. the term h0(t) is called the baseline hazard. It corresponds to the value of the hazard if all the xixi are equal to zero (the quantity exp(0) equals 1). The 't' in h(t) reminds us that the hazard may vary over time. The Cox model can be written as a multiple linear regression of the logarithm of the hazard on the variables xixi, with the baseline hazard being an 'intercept' term that varies with time.

The quantities exp(bi)exp(bi) are called hazard ratios (HR). A value of bibi greater than zero, or equivalently a hazard ratio greater than one, indicates that as the value of the ithith covariate increases, the event hazard increases and thus the length of survival decreases.

Put another way, a hazard ratio above 1 indicates a covariate that is positively associated with the event probability, and thus negatively associated with the length of survival.

In summary,

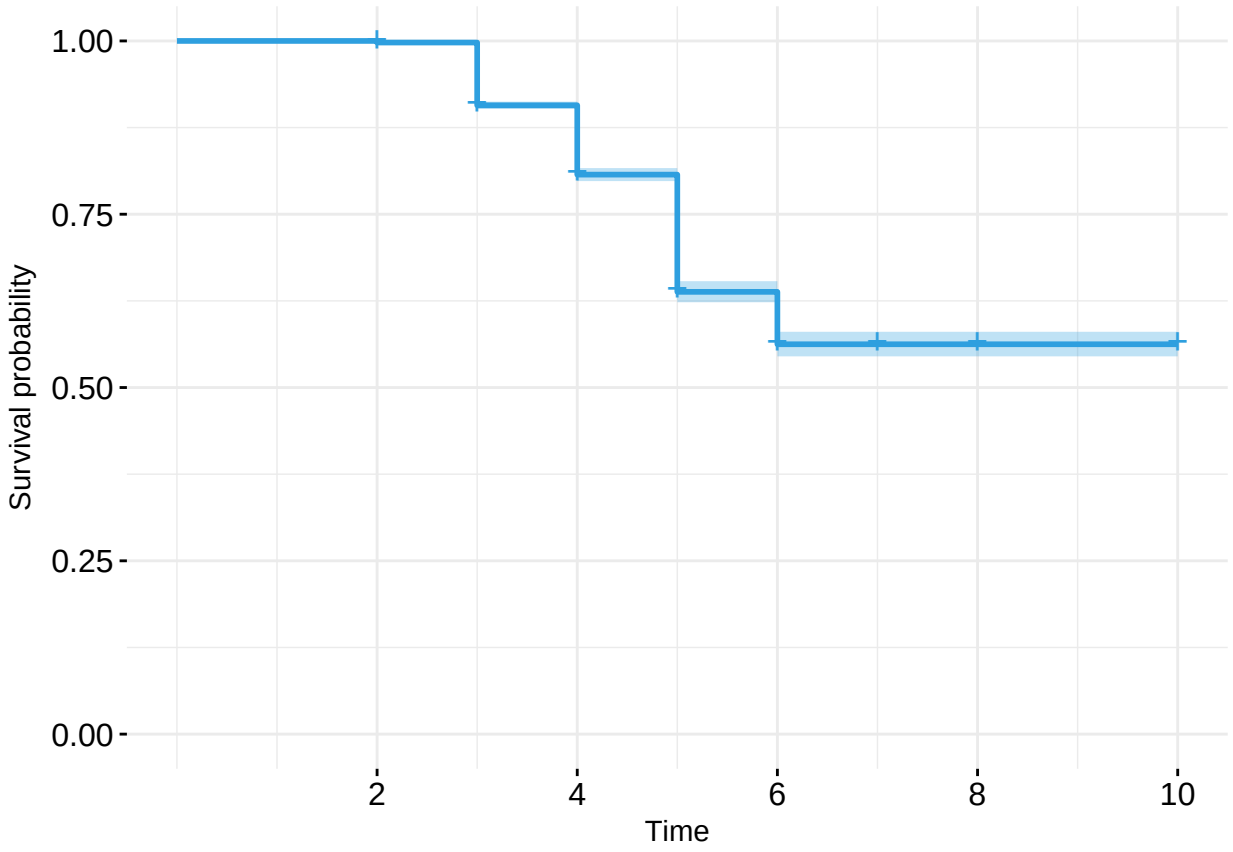
- HR = 1: No effect
- HR < 1: Reduction in the hazard
- HR > 1: Increase in Hazard

Our main results are in Survival Rates at t, which is exp(-Integral of HR from 0 to t)

The first plot is for our full model, the total survival rates over years for all the data. We look at the summary and find the model z value is significant.

```
#we first need to group some non-level variables
#cox model: our full model
hr$left<-as.numeric(hr$left)
hr.cox <- coxph(Surv(time_spend_company, left) ~
                  satisfy+promotion_last_5years+Work_accident, data = hr)

#baseline values
ggsurvplot(survfit(hr.cox), color = "#2E9FDF",
           ggtheme = theme_minimal(),main="Survival Probability Plot for the full Cox Model")
```

```
## Warning in .get_data(fit, data = data): The `data` argument is not
## provided. Data will be extracted from model fit.
```



Evaluation of Cox Model

We took a look at the summary of our full model. R-square is really low compared to a regular regression model. However it is pretty usual for a real-life cox model. We also looked at the p-values for our chosen variables, and all of them have significant p-values. Also there are results for three different tests, they are also significant. (Even though we observe a 0 for p-values, it is really just close to zero). In conclusion, we believe this model is significant.

```
summary(hr.cox)
```

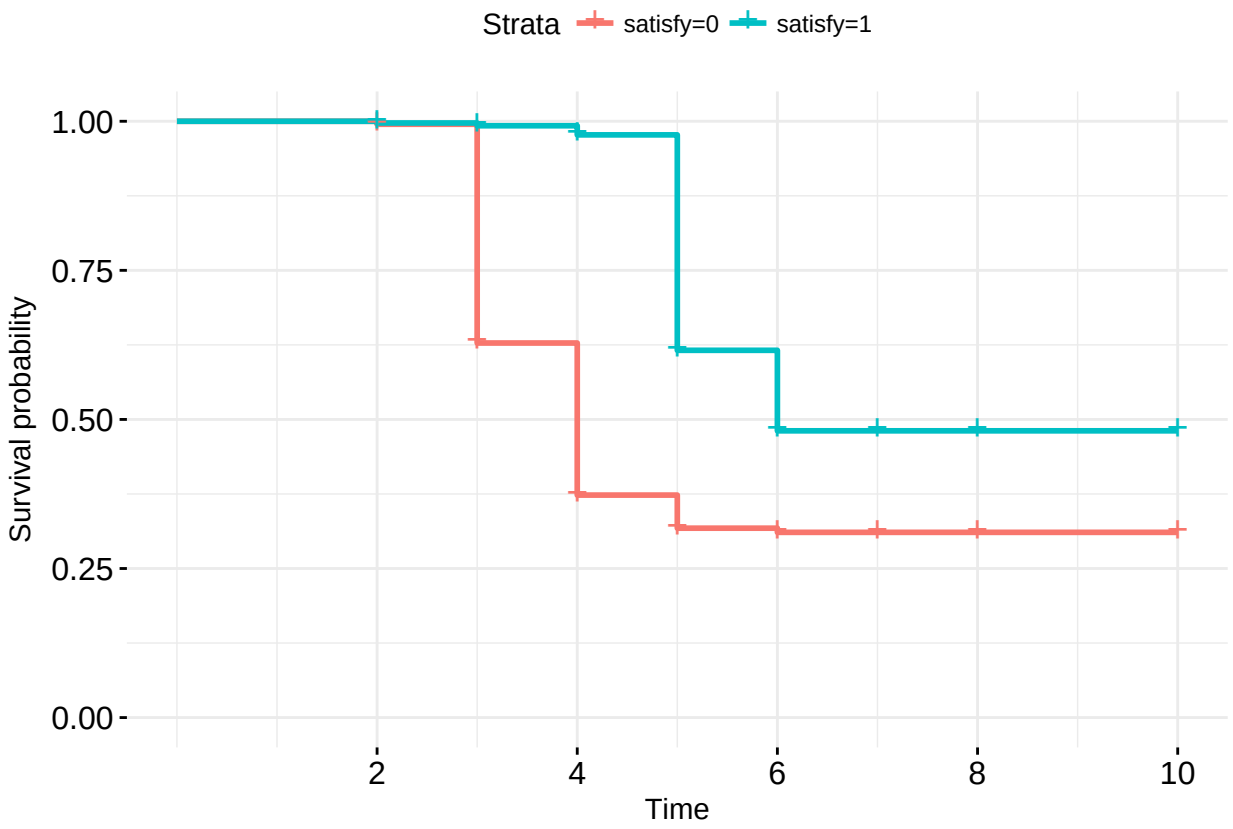
```
## Call:
## coxph(formula = Surv(time_spend_company, left) ~ satisfy + promotion_last_5years +
##     Work_accident, data = hr)
##
##   n= 14999, number of events= 3571
##
##                            coef exp(coef) se(coef)       z Pr(>|z|)
## satisfy               -1.63371   0.19520  0.03708 -44.054  < 2e-16 ***
## promotion_last_5years -1.49810   0.22355  0.23022  -6.507 7.65e-11 ***
## Work_accident          1.20602   3.34016  0.07887  15.291  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## satisfy                  0.1952     5.1229    0.1815    0.2099
## promotion_last_5years    0.2236     4.4732    0.1424    0.3510
## Work_accident            3.3402     0.2994    2.8618    3.8986
```

```
##
## Concordance= 0.823  (se = 0.006 )
## Rsquare= 0.165   (max possible= 0.983 )
## Likelihood ratio test= 2711  on 3 df,   p=0
## Wald test            = 2249  on 3 df,   p=0
## Score (logrank) test = 2793  on 3 df,   p=0
```

Application I:Then we take a look at the effect of individual variables

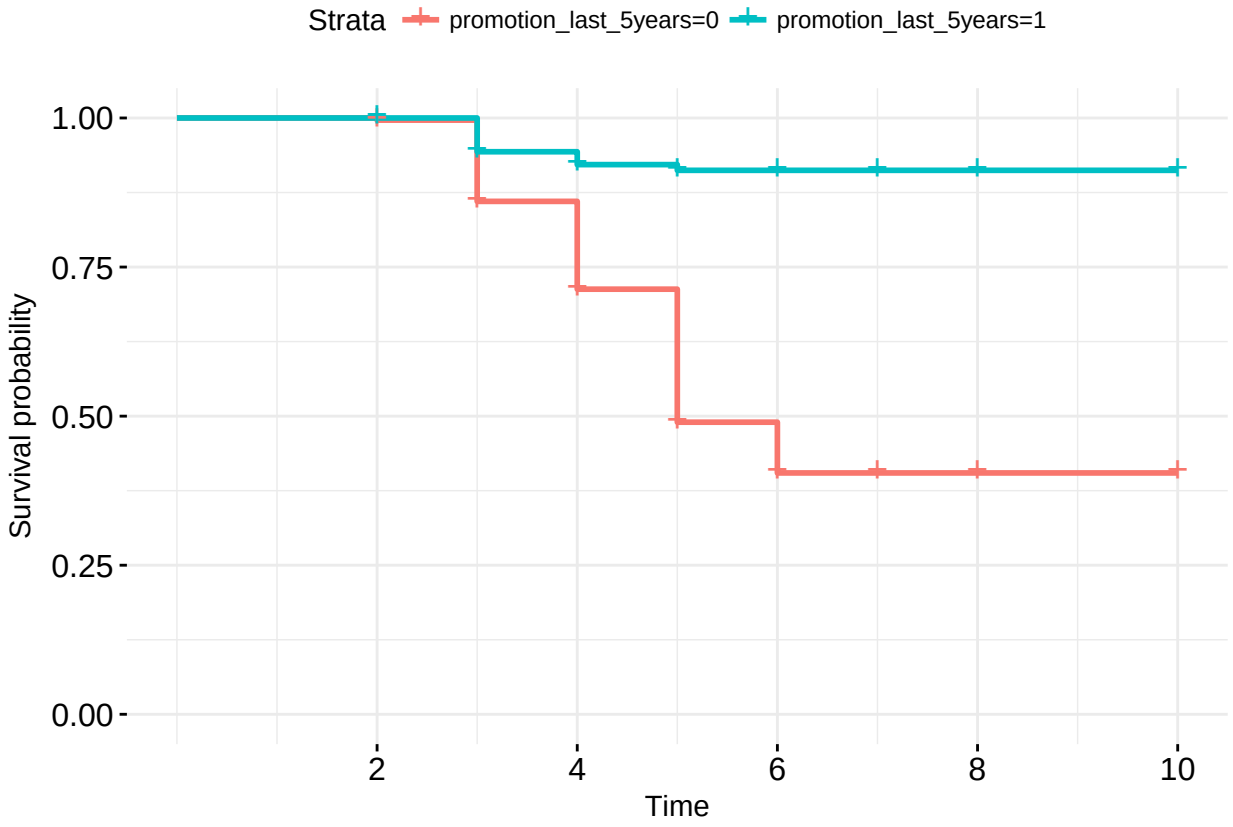
```
#Visualization for different variables
#example: satisfy+promotion_last_5years+Work_accident
fit<-survfit(Surv(time_spend_company, left) ~ satisfy, data = hr)
ggsurvplot(fit,ggtheme = theme_minimal())
```

```
## Warning in .get_data(fit, data = data): The `data` argument is not
## provided. Data will be extracted from model fit.
```



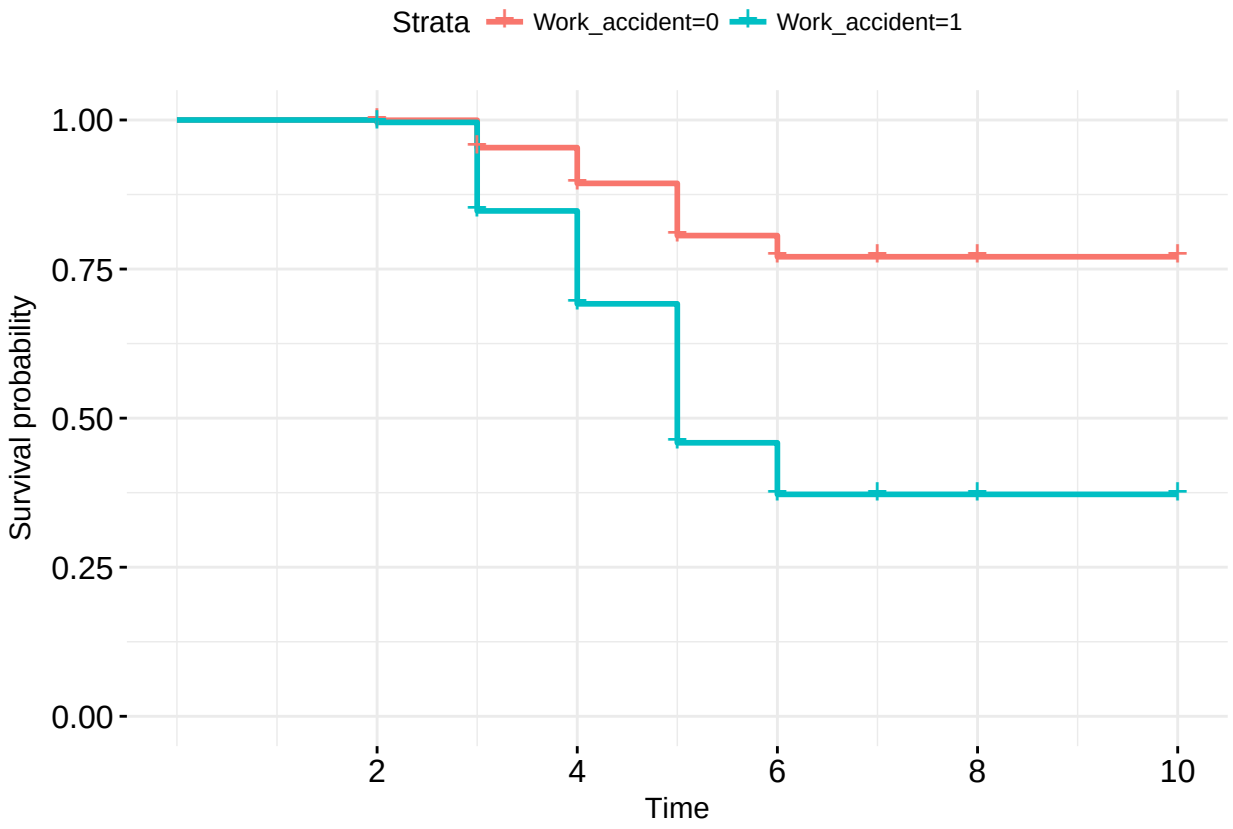

```
fit1<-survfit(Surv(time_spend_company, left) ~ promotion_last_5years, data = hr)
ggsurvplot(fit1,ggtheme = theme_minimal())
```

```
## Warning in .get_data(fit, data = data): The `data` argument is not
## provided. Data will be extracted from model fit.
```



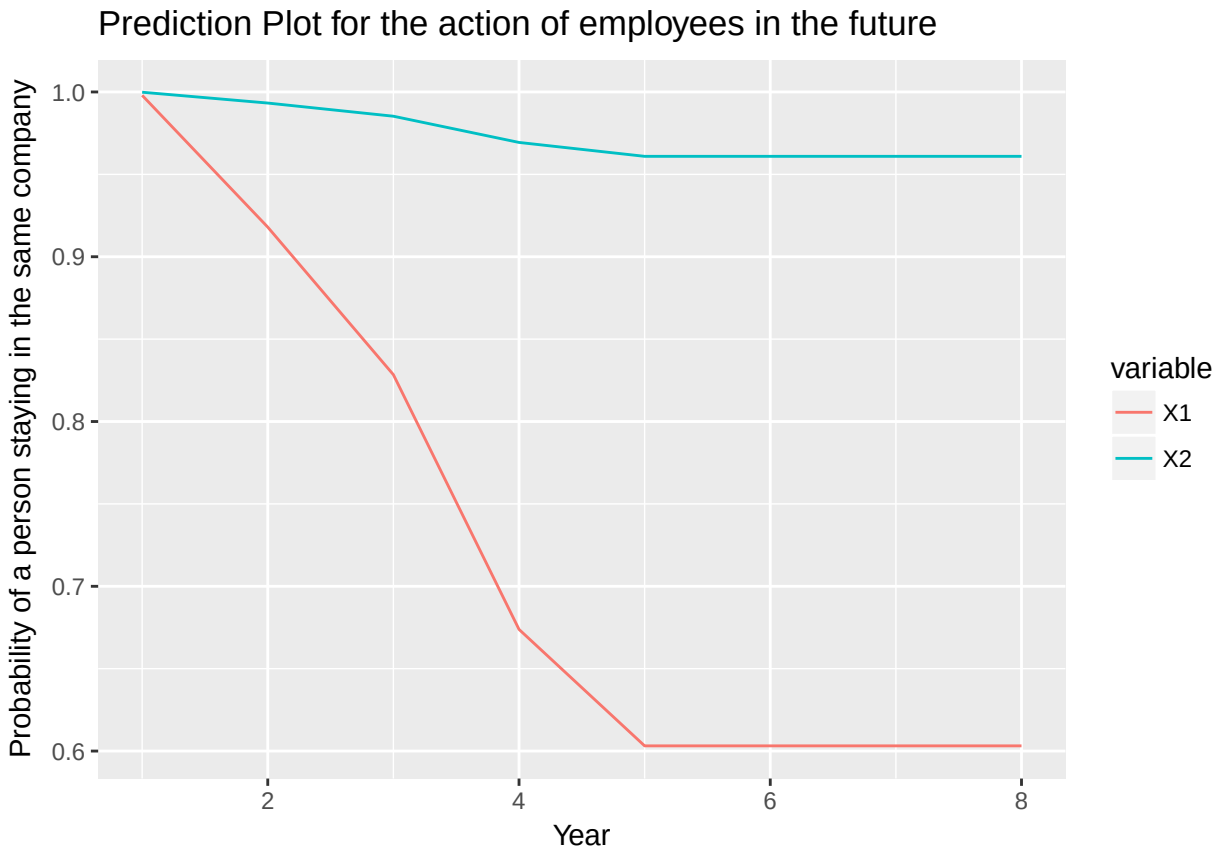

```
fit2<-survfit(Surv(time_spend_company, left) ~ Work_accident, data = hr)
ggsurvplot(fit2,ggtheme = theme_minimal())
```

```
## Warning in .get_data(fit, data = data): The `data` argument is not
## provided. Data will be extracted from model fit.
```



Application II:We can also use the model to predict the future action of employees. The following is the prediction for two made-up employees

```
# predict new data
new <- with(hr,
               data.frame(satisfy=c(0.9,0.8), Work_accident=c(1,0), promotion_last_5years=c(0,1))
               )
#predicted values
fit1<-survfit(hr.cox, newdata = new)
predict<-data.frame(fit1$surv)
predict$time<-1:8

#convert the dataframe to long format
predict_long<-melt(predict, id = "time")
ggplot(data=predict_long, aes(x=time, y=value, colour=variable))+
  geom_line()+
  ggtitle("Prediction Plot for the action of employees in the future") +
  labs(x="Year",y="Probability of a person staying in the same company")
```



Step3 More Prediction Models

Cross-Validation

we split the data randomly into training set using 5-fold-cross-validation

```
# Set the target variable as a factor
hr$left <- as.factor(hr$left)
## install.packages("caret")
library("caret")
# cross-validation
train_control<- trainControl(method="cv", number=5, repeats=3)
head(train_control)
```

```
## $method
## [1] "cv"
##
## $number
## [1] 5
##
## $repeats
## [1] 3
##
## $search
```

```
## [1] "grid"
##
## $p
## [1] 0.75
##
## $initialWindow
## NULL
```

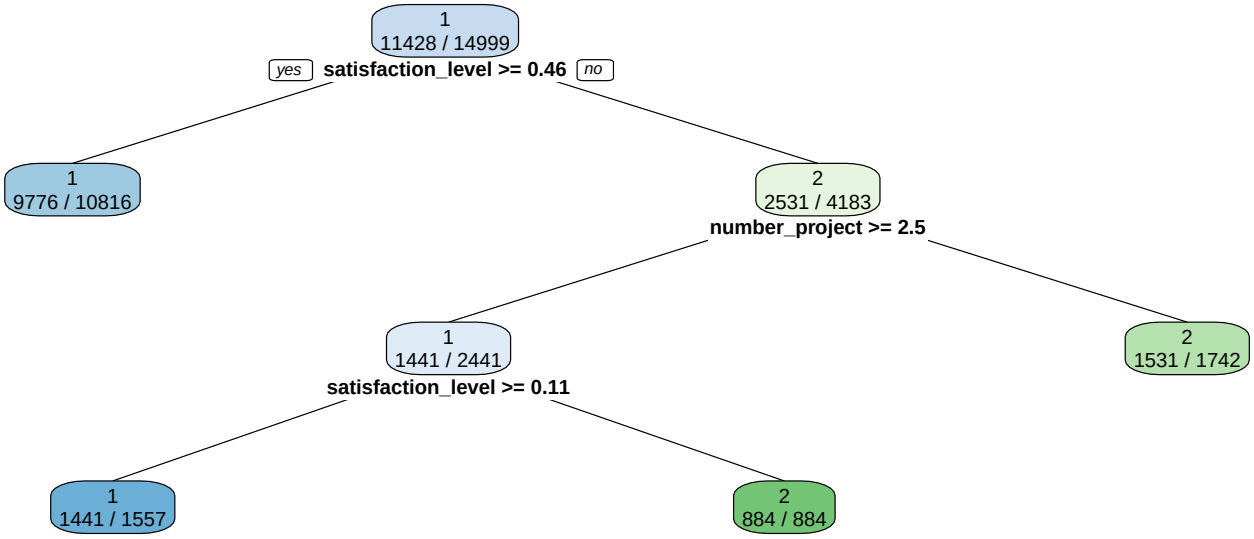
Decision tree

General idea of decision tree is,using a tree-like graph or model of decisions and their possible consequences(The status of the employee).

```
# train the model
rpartmodel<- train(left~., data=hr, trControl=train_control, method="rpart")
# make predictions
predictions<- predict(rpartmodel,hr)
hr_model_tree<- cbind(hr,predictions)
# summarize results
confusionMatrix<- confusionMatrix(hr_model_tree$predictions,hr_model_tree$left)
confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 11217  1156
##          2   211  2415
##
##                Accuracy : 0.9089
##                  95% CI : (0.9041, 0.9134)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7236
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9815
##             Specificity : 0.6763
##          Pos Pred Value : 0.9066
##          Neg Pred Value : 0.9196
##              Prevalence : 0.7619
##          Detection Rate : 0.7478
##    Detection Prevalence : 0.8249
##       Balanced Accuracy : 0.8289
##
##        'Positive' Class : 1
##
```

```
rpart.plot(rpartmodel$finalModel, type = 2, fallen.leaves = F, cex = 0.8, extra = 2)
```



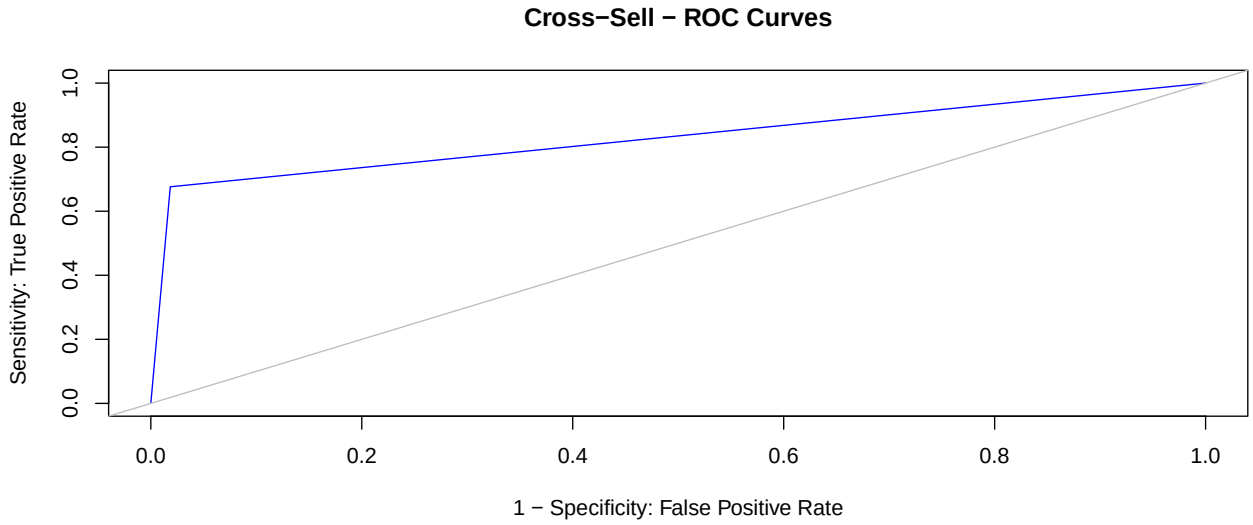

```
#fancyRpartPlot(rpartmodel$finalModel)

library("ROCR")
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
hr_model_tree$predictions <- as.numeric(paste(hr_model_tree$predictions))
#
perf.obj <- prediction(predictions=hr_model_tree$predictions, labels=hr_model_tree$left)
# # Get data for ROC curve
roc.obj1 <- performance(perf.obj, measure="tpr", x.measure="fpr")
plot(roc.obj1,
     main="Cross-Sell - ROC Curves",
     xlab="1 - Specificity: False Positive Rate",
     ylab="Sensitivity: True Positive Rate",
    col="blue")
abline(0,1,col="grey")
```



Random Forest

The idea outperform decision tree by constructing multiple decision trees, and classify the new object down each of the trees in the forest. The forest chooses the classifacation having the most votes from the trees.

```
# train the model
randomforestModel<- train(left~., data=hr, trControl=train_control, ntree=30,method="rf")
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

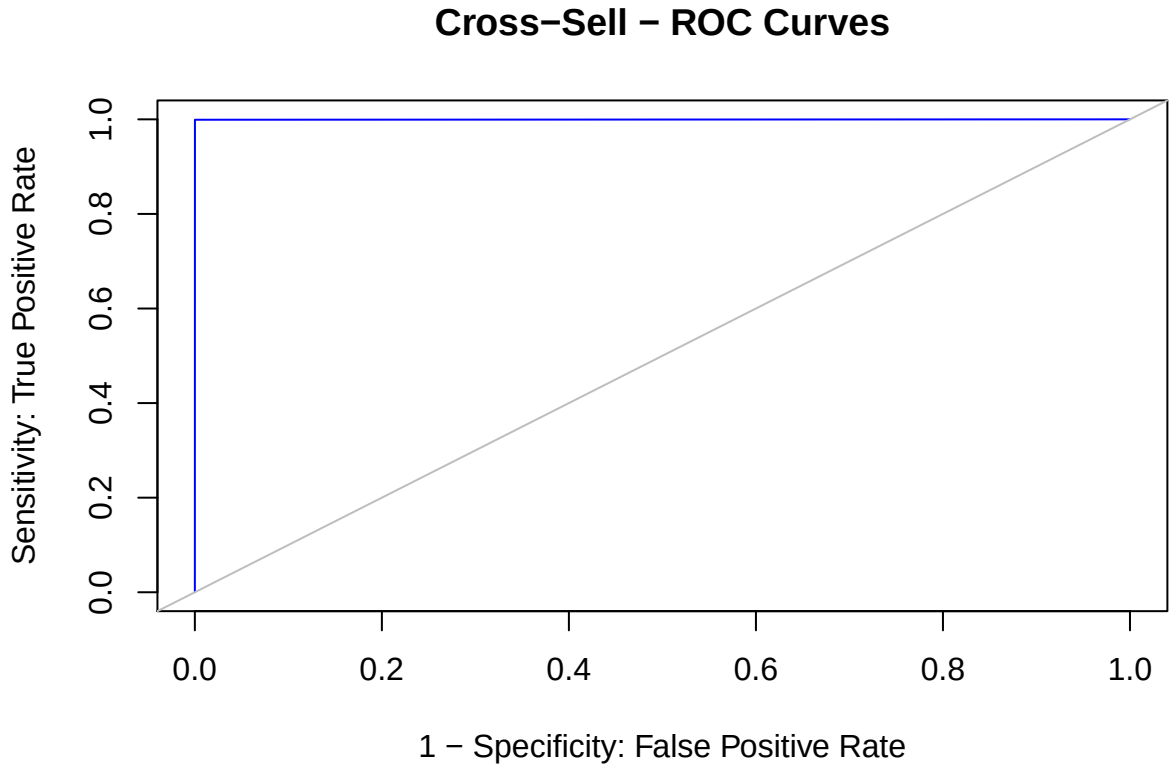
```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# make predictions
predictions<- predict(randomforestModel,hr)
hr_model_randomforest<- cbind(hr,predictions)
# summarize results
confusionMatrix<- confusionMatrix(hr_model_randomforest$predictions,hr_model_randomforest$left)
confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 11427     3
##          2     1  3568
##
##                Accuracy : 0.9997
##                  95% CI : (0.9993, 0.9999)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9993
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9999
##             Specificity : 0.9992
##          Pos Pred Value : 0.9997
##          Neg Pred Value : 0.9997
##              Prevalence : 0.7619
##          Detection Rate : 0.7619
##    Detection Prevalence : 0.7621
##       Balanced Accuracy : 0.9995
##
##        'Positive' Class : 1
##
```

```
library("ROCR")
hr_model_randomforest$predictions <- as.numeric(paste(hr_model_randomforest$predictions))
#
perf.obj <- prediction(predictions=hr_model_randomforest$predictions, labels=hr_model_randomforest$left)
# # Get data for ROC curve
roc.obj2 <- performance(perf.obj, measure="tpr", x.measure="fpr")
plot(roc.obj2,
     main="Cross-Sell - ROC Curves",
     xlab="1 - Specificity: False Positive Rate",
     ylab="Sensitivity: True Positive Rate",
    col="blue")
abline(0,1,col="grey")
```



Naives Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. We use bayesian probability to calculate the probability of occurence of each features and decide the classfication results based on the highest probability.

```
# train the model
e1071model2 <- train(left~., data=hr, trControl=train_control, method="nb")
```

```
## Loading required package: klaR
```

```
## Loading required package: MASS
```

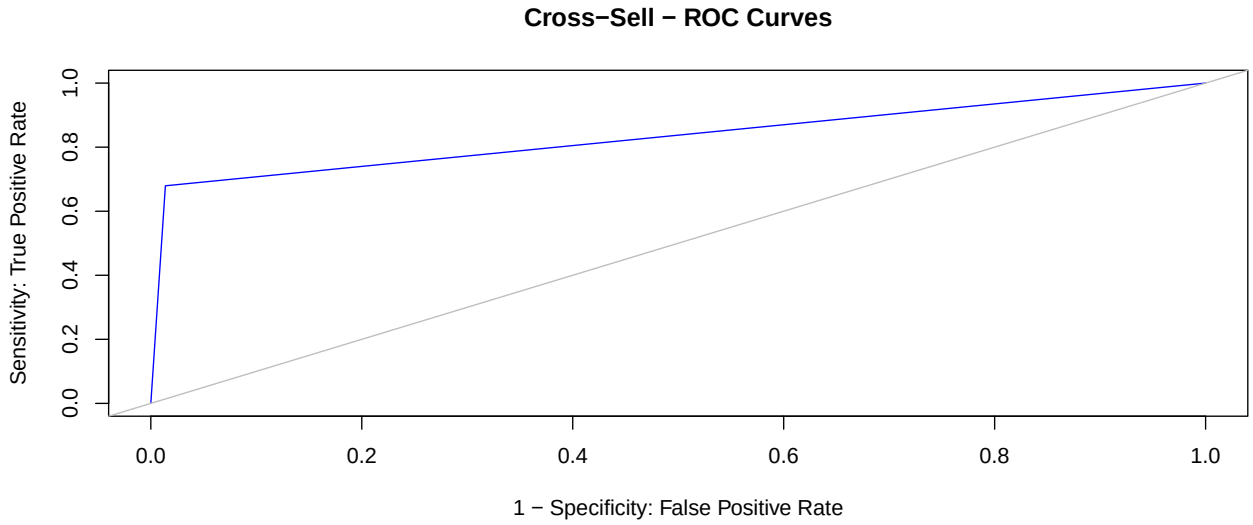
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# make predictions
predictions<- predict(e1071model2,hr)
e1071modelbinded <- cbind(hr,predictions)
# summarize results
confusionMatrix<- confusionMatrix(e1071modelbinded$predictions,e1071modelbinded$left)
confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 11270  1145
##          2   158  2426
##
##                Accuracy : 0.9131
##                  95% CI : (0.9085, 0.9176)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7354
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9862
##             Specificity : 0.6794
##          Pos Pred Value : 0.9078
##          Neg Pred Value : 0.9389
##              Prevalence : 0.7619
##          Detection Rate : 0.7514
##    Detection Prevalence : 0.8277
##       Balanced Accuracy : 0.8328
##
##        'Positive' Class : 1
##
```

```
library("ROCR")
e1071modelbinded$predictions <- as.numeric(paste(e1071modelbinded$predictions))
#
perf.obj <- prediction(predictions=e1071modelbinded$predictions, labels=e1071modelbinded$left)
# # Get data for ROC curve
roc.obj3 <- performance(perf.obj, measure="tpr", x.measure="fpr")
plot(roc.obj3,
     main="Cross-Sell - ROC Curves",
     xlab="1 - Specificity: False Positive Rate",
     ylab="Sensitivity: True Positive Rate",
    col="blue")
abline(0,1,col="grey")
```



Logistic regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

```
# train the model
gmlmodel <- train(left~., data=hr, trControl=train_control, method="LogitBoost")
```

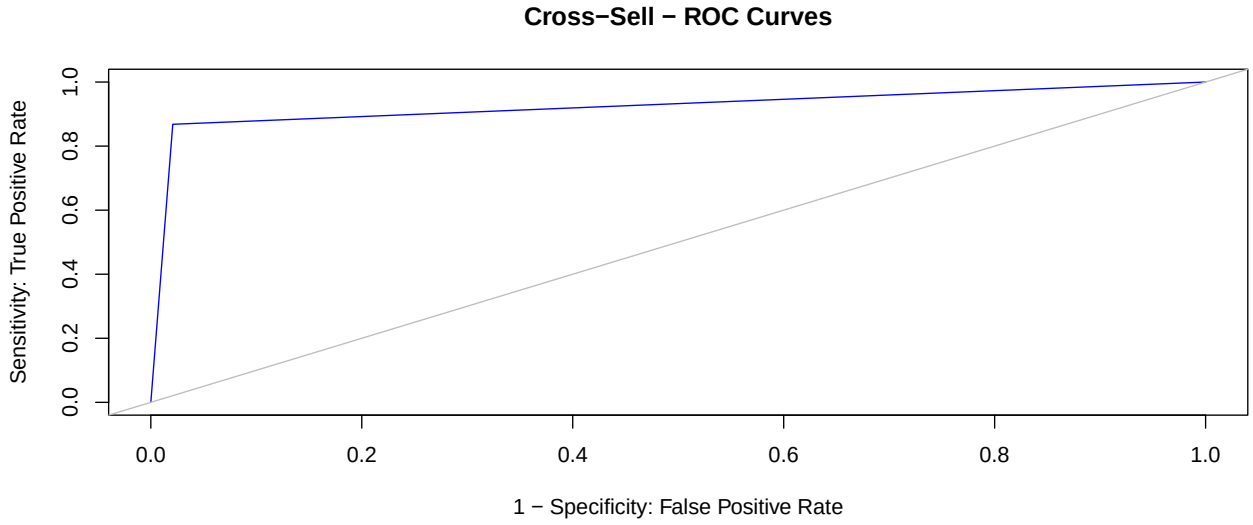
```
## Loading required package: caTools
```

```
# make predictions
predictions<- predict(gmlmodel,hr)
gmlmodelbinded <- cbind(hr,predictions)
# summarize results
confusionMatrix<- confusionMatrix(gmlmodelbinded$predictions,gmlmodelbinded$left)
confusionMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 11190   471
##          2   238  3100
##
##                Accuracy : 0.9527
##                  95% CI : (0.9492, 0.9561)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8667
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9792
##             Specificity : 0.8681
```

```
##          Pos Pred Value : 0.9596
##          Neg Pred Value : 0.9287
##              Prevalence : 0.7619
##          Detection Rate : 0.7460
##    Detection Prevalence : 0.7775
##       Balanced Accuracy : 0.9236
##
##        'Positive' Class : 1
##
```

```r
library("ROCR")
gmlmodelbinded$predictions <- as.numeric(paste(gmlmodelbinded$predictions))

perf.obj <- prediction(predictions=gmlmodelbinded$predictions, labels=gmlmodelbinded$left)
# Get data for ROC curve
roc.obj4 <- performance(perf.obj, measure="tpr", x.measure="fpr")
plot(roc.obj4,
     main="Cross-Sell - ROC Curves",
     xlab="1 - Specificity: False Positive Rate",
     ylab="Sensitivity: True Positive Rate",
     col="blue")
abline(0,1,col="grey")
```

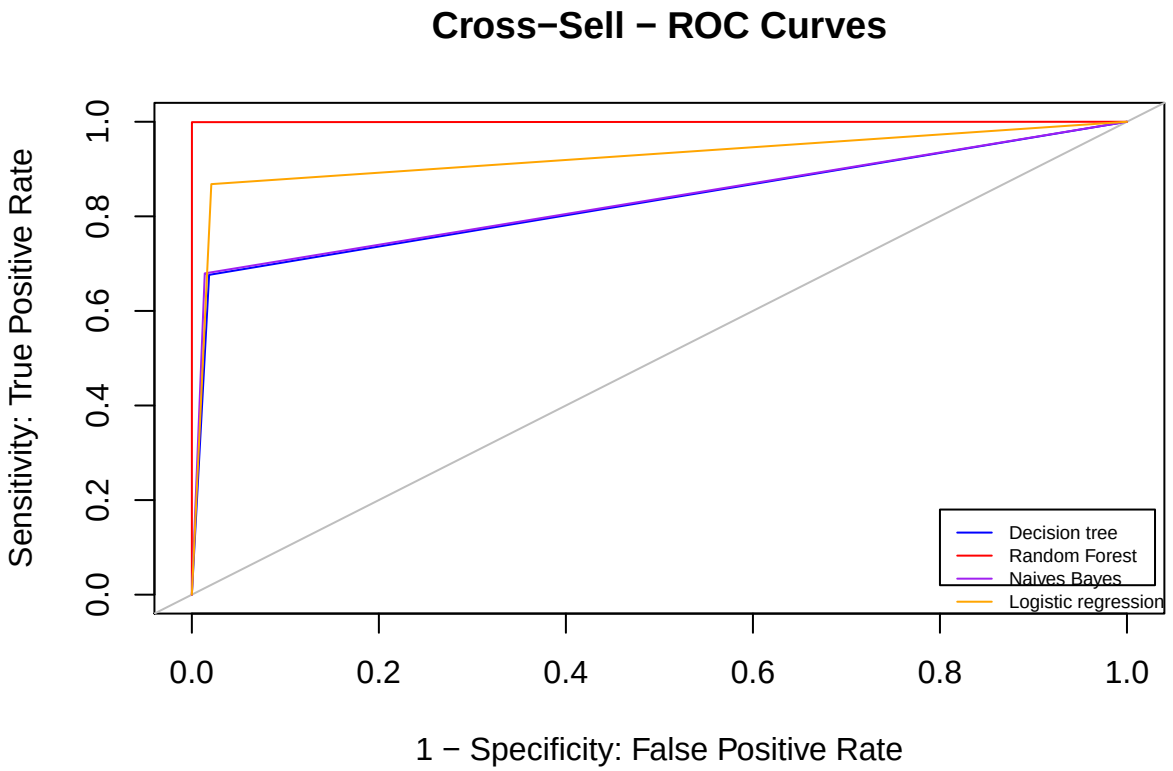


Step4 Comparision

Conclusion: Random Forest outperform other models.

```r
library("ROCR")
plot(roc.obj1,
     main="Cross-Sell - ROC Curves",
     xlab="1 - Specificity: False Positive Rate",
     ylab="Sensitivity: True Positive Rate",
    col="blue")
abline(0,1,col="grey")
```

```
plot(roc.obj2,col="red",add=T)
plot(roc.obj3,col="purple",add=T)
plot(roc.obj4,col="orange",add=T)
legend(x=c(0.80,1.03),y=c(0.02,0.18),legend=c("Decision tree","Random Forest","Naives Bayes","Logistic
```



Shiny App Link

[I'm shiny app link, click me] https://ads-yz3032.shinyapps.io/who_will_leave/