Logistic Regression Model:

Function: model <- glm(selected ~ stats + perc + as.factor(rank) , family = binomial)

```
full.model = glm(resp ~., data = train2.data, family = "binomial")

summary(full.model)

model.1 = glm(admit ~ gre + gpa + rank, data = binary1, family = "binomial")

summary(model.1)
```

Linear Regression Model:

```
result<-lm(point_score~shooting_per,data=shooting)
cars.lm=lm(mpg~hp,data=mtcars)
mod2 = lm(income ~ prestige.c + women.c, data = newdata)

summary(mod2)
```

Decision Tree:

```
library(rpart)

library(partykit)

tree.1 = rpart(donated_blood~., data=transfusion, control = rpart.control(minsplit = 30, minbucket = 20, maxdepth = 4))

plot(as.party(tree.1))

tree.1 = rpart(High.quality~., data=wine.base, control = rpart.control(minsplit=60, minbucket = 30, maxdepth = 4))
```

Random Forest:

```
library(randomForest)
library(caret)
rf.model.1 = randomForest(y ~ ., data=train.data, ntree = 200,importance = TRUE)
print(rf.model.1)
rf.model.2 = randomForest(y ~ duration + month + age + day + poutcome + job, data=train.data,ntree = 200,importance = TRUE)
print(rf.model.2)
```

SVM:

```
library(e1071)
svm.model.1 = svm(Species ~., data = iris.train, kernel = "linear", cost = 0.1, scale = FALSE)
print(svm.model.1)
plot(svm.model.1, iris.train[,columns])
```

Confusion Matrix:

```
library(caret)
confusionMatrix(data = svm.pred, reference = iris.test$Species)
```

KNN:

```
library(class)
test.pred = knn(train = cancer.train, test = cancer.test, cl = train.labels, k = 10)
test.pred
print(test.pred)
summary(test.pred)
```

Time Series:

```
## Simple Moving Average (SMA)

library(TTR)
sma3 = SMA(birthtimeseries.monthly, n = 3)
sma3
plot.ts(sma3)

## weighted Moving Average (WMA)

library(TTR)
wma3 = WMA(birthtimeseries.monthly, n = 3, wts = 1:3)
wma3
plot.ts(wma3)

## forecasting for next 12 month

library(forecast)
birthseries.forecast12 = forecast.HoltWinters(birthseries.forecast, h =12)  ## Forecast for next 12 months
birthseries.forecast12   ## forecasted value for next 12 month
```

ARIMA:

```
fit = arima(log(AirPassengers), c(1,1,1), seasonal = list(order = c(1,1,1), period = 12)) ## period = 12 means 12 months
```

Text Mining

```
library(tm)
library(tau)
library(plyr)
library(SnowballC)
library(wordcloud)
library(ggplot2)
library(data.table)
library(topicmodels)

#create a corpus

mycorpus<-Corpus(VectorSource(tweets$message))
mycorpus
```

Naïve Bayes:

```
library(e1071)
sms_classifier <- naiveBayes(sms_train, sms_train_labels)
sms_classifier
```

Market Basket Analysis:

```
library(arules)
library(arulesViz)
library(datasets)

rules = apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))
summary(rules)
```

Clustering:

```
library(datasets)
set.seed(100)
kmeans.1 = kmeans(attitude, 4, nstart = 100) ## nstart=100 means 100 number of iteration
kmeans.1
```