Report Summary –

Outcome of the Data Mining Techniques on XYZ Insurance Company Data

A data set (claims1.csv – Refer Annexure) is provided by XYZ Insurance company on the Auto Insurance Claims Reported details alongwith the Fraud details i.e. whether any claim fraud has been observed and detected against each of the Auto Insurance Claim Reported Transaction. The data set consists of claim reported transactions (First Notice of Loss) has 33 variables having the details like Month, Week of the claim, Policy Details, Personal details of the Policy Holder, Policy Type, Person at Fault during Accident, Vehicle details with Price, etc. including the variable whether Fraud Found in case of individual transaction.

Given the data set, XYZ Insurance company wants to apply various data mining techniques to generate a best fit model which will help to predict the potential fraud in future in any insurance claim reported at the stage of First Notice of Loss and to understand the variables which has a high dependency on the outcome of the claim fraud. Such an analysis will help XYZ Insurance company to check and investigate potential fraud at the very initial stage of claim reporting and refer the claim to Special Fraud Investigation Office (SFIO) accordingly for further investigation.

The data mining techniques like Logistic Regression, Decision Tree & Random Forests have been applied on the given data set to achieve the objective as mentioned.

The report specifies the individual steps taken while doing Logistic Regression, Decision Tree & Random Forests and the inferences drawn in each step.

Method 1: Logistic Regression

Step 1: Transformation of the categorical predictor variables like Vehicle Price, Marital Status, Policy Type, Vehicle Category, Fault, Accident Area using dummy variables

Reason: The significant categorical predictor variables have been transformed to 0 & 1 using dummy variables

Reference R Code:

```
setwd("D:R")

aa<-read.csv("claims1.csv",header = TRUE)

aa


aa$VehiclePrice1<-as.numeric(aa$VehiclePrice=="more than 69000")

aa

aa$VehiclePrice2<-as.numeric(aa$VehiclePrice=="20000 to 29000")

aa
```

```
aa$VehiclePrice3<-as.numeric(aa$VehiclePrice=="30000 to 39000")
aa

aa$VehiclePrice4<-as.numeric(aa$VehiclePrice=="40000 to 59000")
aa

aa$VehiclePrice5<-as.numeric(aa$VehiclePrice=="60000 to 69000")
aa

aa$MaritalStatus1<-as.numeric(aa$MaritalStatus=="Married")
aa

aa$MaritalStatus2<-as.numeric(aa$MaritalStatus=="Single")
aa

aa$MaritalStatus3<-as.numeric(aa$MaritalStatus=="Widow")
aa

aa$PolicyType1<-as.numeric(aa$PolicyType=="Sedan - Collision")
aa

aa$PolicyType2<-as.numeric(aa$PolicyType=="Sedan - Liability")
aa

aa$VehicleCategory1<-as.numeric(aa$VehicleCategory=="Sedan")
aa

aa$Fault1<-as.numeric(aa$Fault=="Policy Holder")
aa
```

```
aa$AccidentArea1<-as.numeric(aa$AccidentArea=="Urban")

aa

summary(aa)

boxplot(aa)
```

Inference/Output –The values of the categorical variables are transformed using dummy variables 0 & 1

Step 2: Removal of the original categorical variables which have been transformed using dummy variables and some other assumed insignificant variable

Reason : The original categorical variable which have been transformed using dummy variables have been removed to do away with high multicollinearity with their transformed variables. In addition, some insignificant variables are also removed.

Reference R Code:

```
ak<-subset(aa,select = c(VehiclePrice, MaritalStatus, PolicyType, VehicleCategory, Fault,
AccidentArea, MonthClaimed,Month,WeekOfMonth, DayOfWeek, DayOfWeekClaimed,
AddressChange_Claim))

ak
```

Inference/Output – The subset after removal of the variables is created

Step 3: Checking multicollinearity amongst predictor variables

Reason: Idea is to check the multicollinearity amongst the predictor variables to check if any predictor variable/s are moderately/highly correlated with each other resulting in unstable parameter estimates making it very difficult to assess the effect of predictor variables on dependent variables

Reference R Code:

```
var<-glm(FraudFound_P~.,data=ak,family="binomial")

summary(var)

library(car)

vf<-vif(var)

vf
```

Inference/Ouput: Since the GVIF of the predictor variables BasePolicy and transformed variable PolicyType2 are significantly higher than 10, inference is

drawn that both the predictor variables have high multicollinearity and should be dropped from the model

Step 4: Running logistic regression with the predictor variables & performing log likelihood test and pseudo R Square test

Reason: The idea is to run logistic regression and perform log likelihood test and pseudo R Square test to find the goodness of fit for the model

Reference R Code:

```
gp<-glm(FraudFound_P~.-BasePolicy-PolicyType2,data=ak,family="binomial")

summary(gp)

##Loglikelihood test and pseudo R Square test

library(lmtest)

test<-lrtest.default(gp)

(test)

library(pscl)

pR2(gp)
```

Inference/Output – The outcome of the loglikehood test is T=6987.6-5952.2=1035.4 with 73 degrees of freedom and a corresponding p-value that is significantly low. Thus in a hypothesis test, a large value of T with significantly low p-value indicates that the fitted model is significantly better than the null model that uses only the intercept term

The outcome of the Pseudo R Square which is 0.1481 indicates a good fit of the model over the null model

Step 5: Running logistic regression model on training data set and testing data set

Reason: To check the accuracy of the model

Reference R Code:

```
library(caTools)

set.seed(200)

spl<-sample.split(ak,SplitRatio=.7)

spl
```

```
aatr<-subset(ak,spl==TRUE)
aatst<-subset(ak,spl==FALSE)
head(aatst)
nrow(aatst)
fit<-glm(FraudFound_P~.-BasePolicy-PolicyType2,data=aatr,family="binomial")
fit
summary(gp)
nk<-predict(fit,aatst,type='response')
nk
pk<-ifelse(nk>=.5,1,0)
pk
sk<-data.frame(aatst,nk,pk)
write.csv(sk,"aatstn.csv")

##Creation of Confusion Matrix & Checking Accuracy of the Model
tb<-table(pk,aatst$FraudFound_P)
tb
diag(tb)
mk<-sum(diag(tb))/sum(tb)
mk
1-mk
```

Inference/Output: Accuracy of the Model is 0.940481 or 94.05%

Step 7: Plotting the ROC Curve and deduction of the area under the curve

Reference R Code:

```
library(ROCR)
rf<-prediction(nk,aatst$FraudFound_P)
rg<-performance(rf,"tpr","fpr")
library(ggplot2)
```

```
plot(rg)

abline(a=0,b=1)

tt<-performance(rf,"auc")

tt
```

Inference/Output: Area under the ROC Curve is 78.22%. We can deduce from the OUTPUT of the above steps that the logistic regression model thus created is a good fit model.

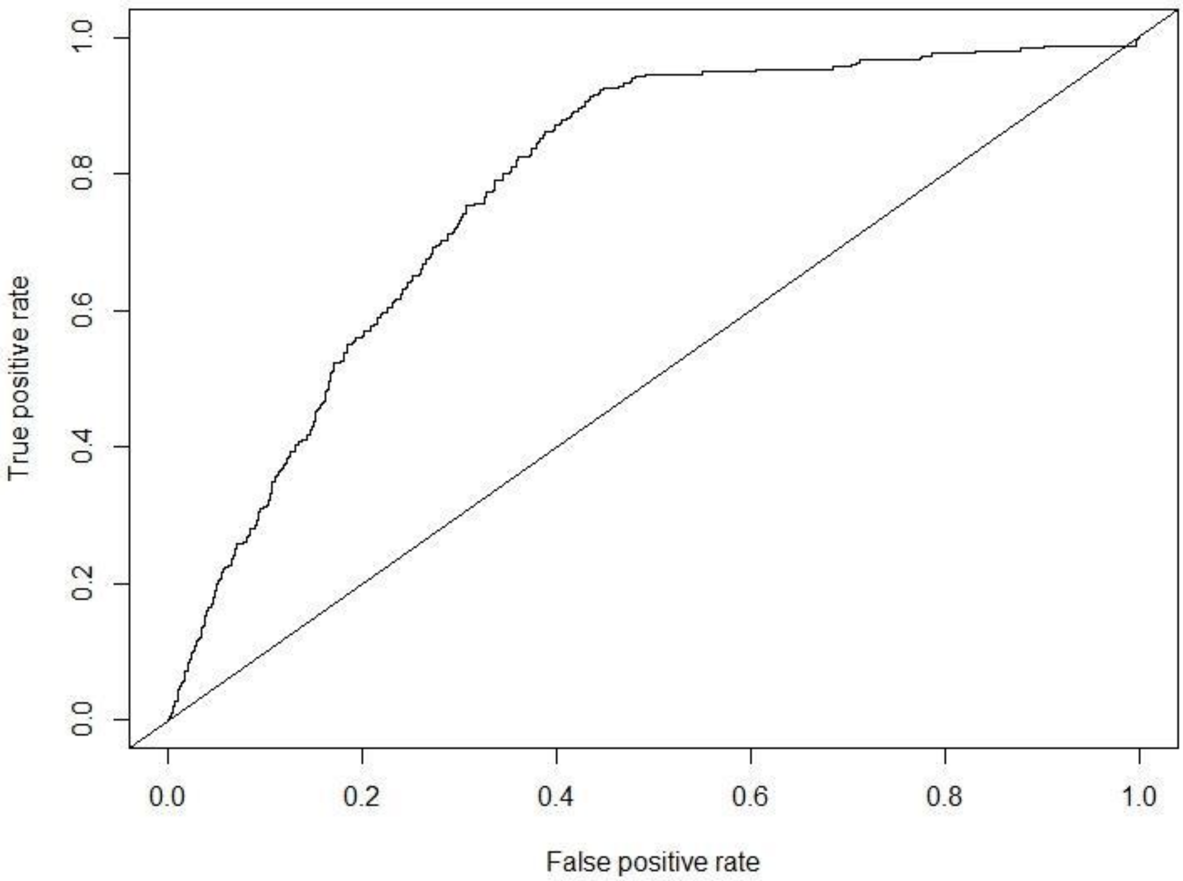

Fig.1 ROC Curve

Step 8: Plotting the Lift Chart to measure the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

Reference R Code:

```
library(lift)

lf<- performance(rf,"lift","rpp")

lf

plot(lf)
```

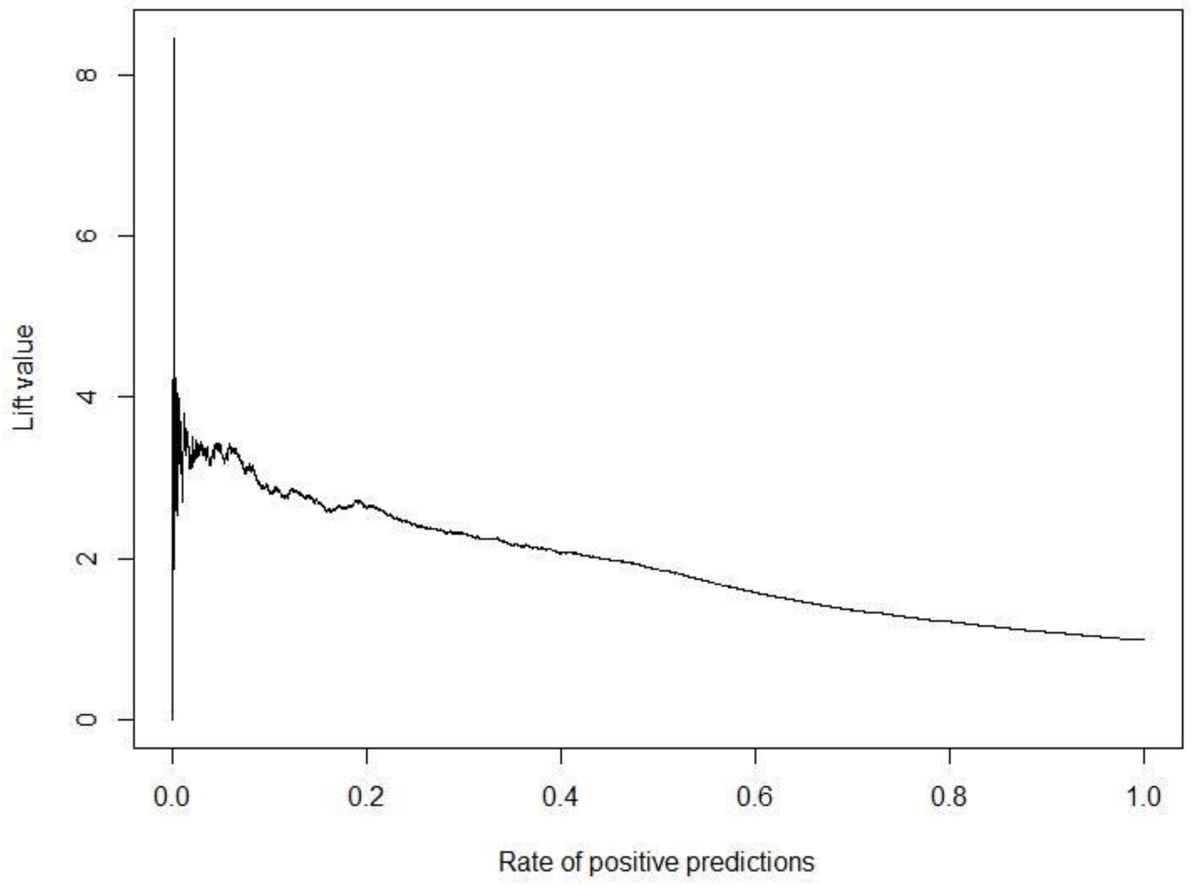

Fig 2. Lift Chart

Step 9: Plotting the Precision vs. Recall graph

Reference R Code:

pr<- performance(rf, "prec", "rec")

pr

plot(pr)

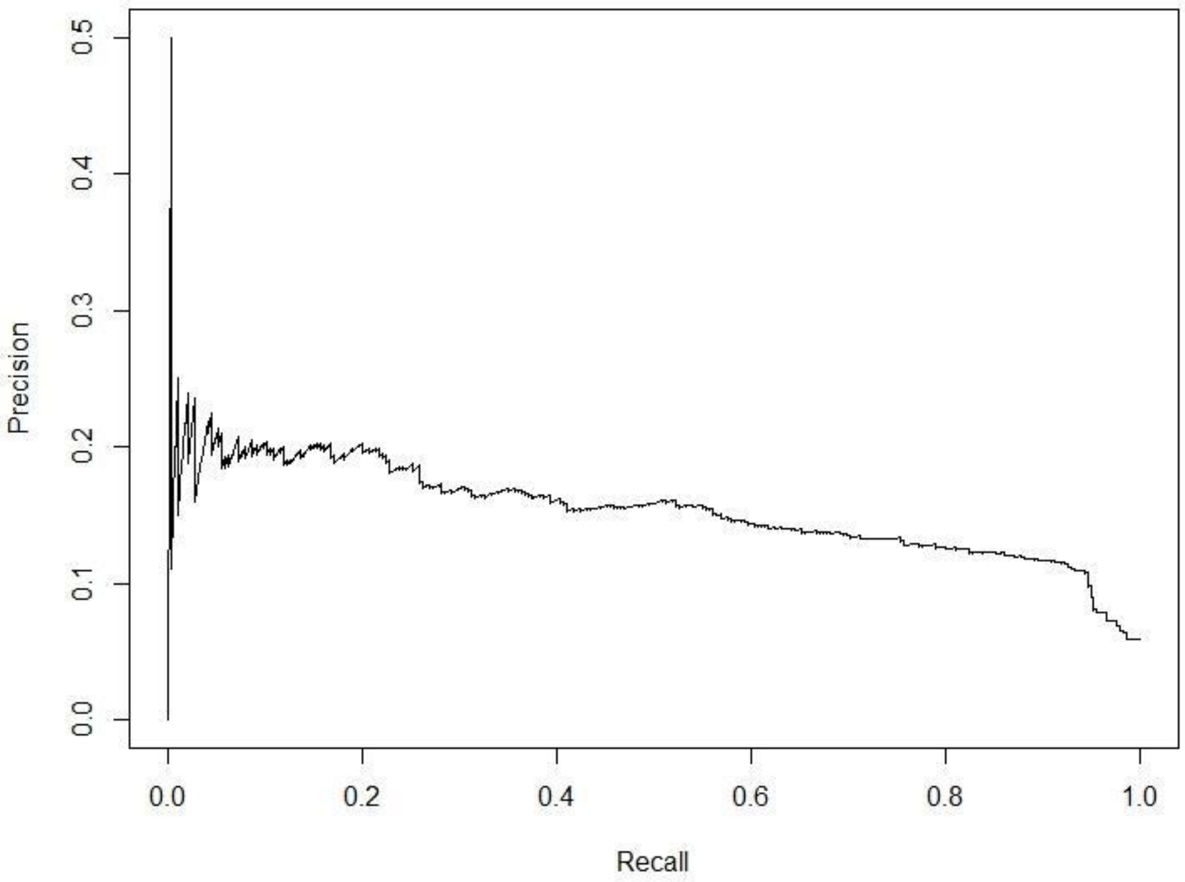

Fig 3. Precision vs. Recall Graph

Method 2 : Decision Tree

Step 1: Running Decision Tree model with the all predictor variables

Reference R Code:

```
library(rpart)
setwd("D:R")
aa<-read.csv("claims1.csv",header=TRUE)
head(aa)
library(caTools)
set.seed(200)
dplit<-sample.split(aa,SplitRatio=.7)
aatr<-subset(aa,dplit==TRUE)
aatst<-subset(aa,dplit==FALSE)
length(aatst$FraudFound_P)
ab<-rpart(FraudFound_P~.,data=aatr,method="class")
plot(ab,margin=0.3)
text(ab,pretty=0)
aab<-predict(ab,aatst,type="class")
aab
##Creation of Confusion Matrix & Checking Accuracy of the Model
tb<-table(aatst$FraudFound_P,aab)
tb
sum(diag(tb))/sum(tb)
```

Inference/Output: The Decision Tree is generated and the Accuracy of the Model is 0.940522 or 94.05%

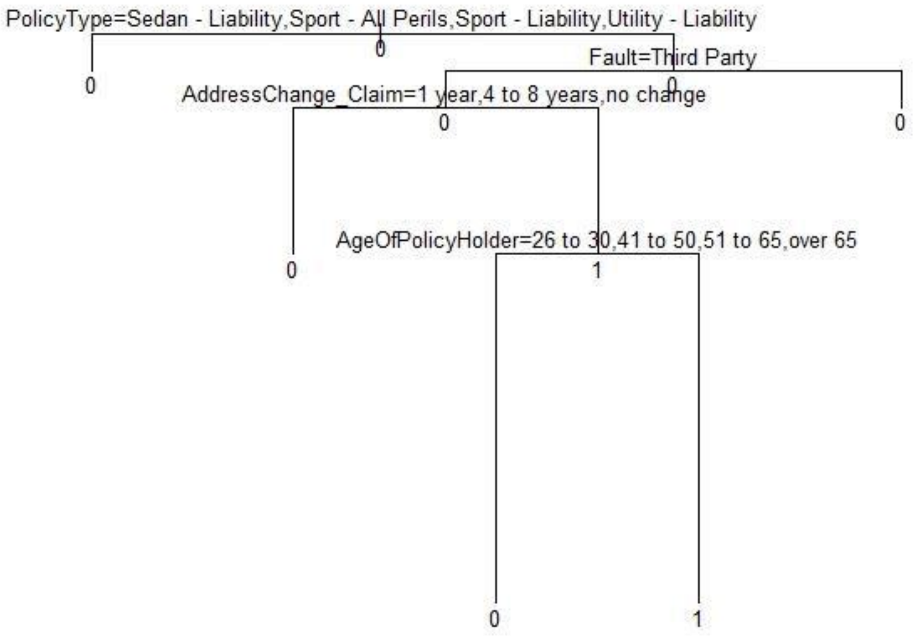
PolicyType=Sedan - Liability,Sport - All Perils,Sport - Liability,Utility - Liability
0
0
Fault=Third Party
0
AddressChange_Claim=1 year,4 to 8 years,no change
0
0
0
AgeOfPolicyHolder=26 to 30,41 to 50,51 to 65,over 65
1
0
1

Fig 4. Decision Tree

Inference/Output: The Decision tree shows Policy Type at the root and further down the line creation of decision branches dependent on the variables like Fault, AddressChange_Claim & AgeOfPolicyHolder with decision 0 or 1 at the leaf node (class label).

Step 2: Plotting the ROC Curve and deduction of the area under the curve

Reference R Code:

```
library(ROCR)
aac<-predict(ab,aatst,type="prob")
head(aac)
aad<-aac[,2]
head(aad)
aae<-prediction(aad,aatst$FraudFound_P)
rocc<-performance(aae,"tpr","fpr")
library(ggplot2)
plot(rocc)
abline(a=0,b=1)
performance(aae,"auc")
```

Inference/Output: Area under the ROC Curve is 78.52%. We can deduce from the OUTPUT of the above steps that the Decision Tree model thus created is a good fit model.

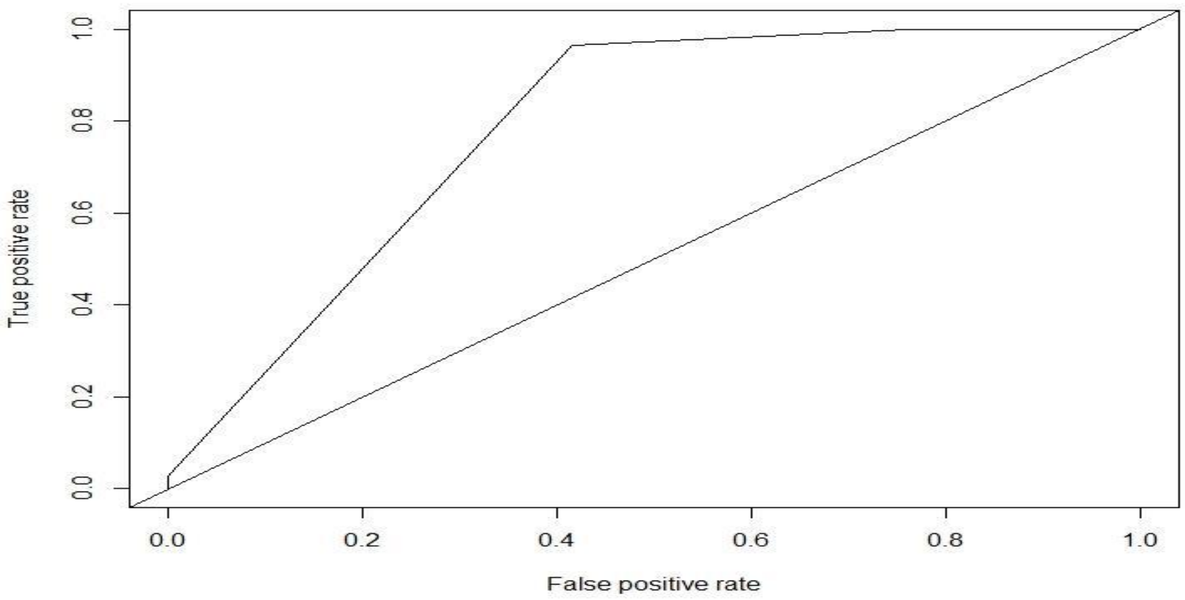

Fig 5. ROC Curve

Step 3: Plotting the Lift Chart to measure the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

Reference R Code:

```
library(lift)

lf<- performance(aae,"lift","rpp")

lf

plot(lf)
```

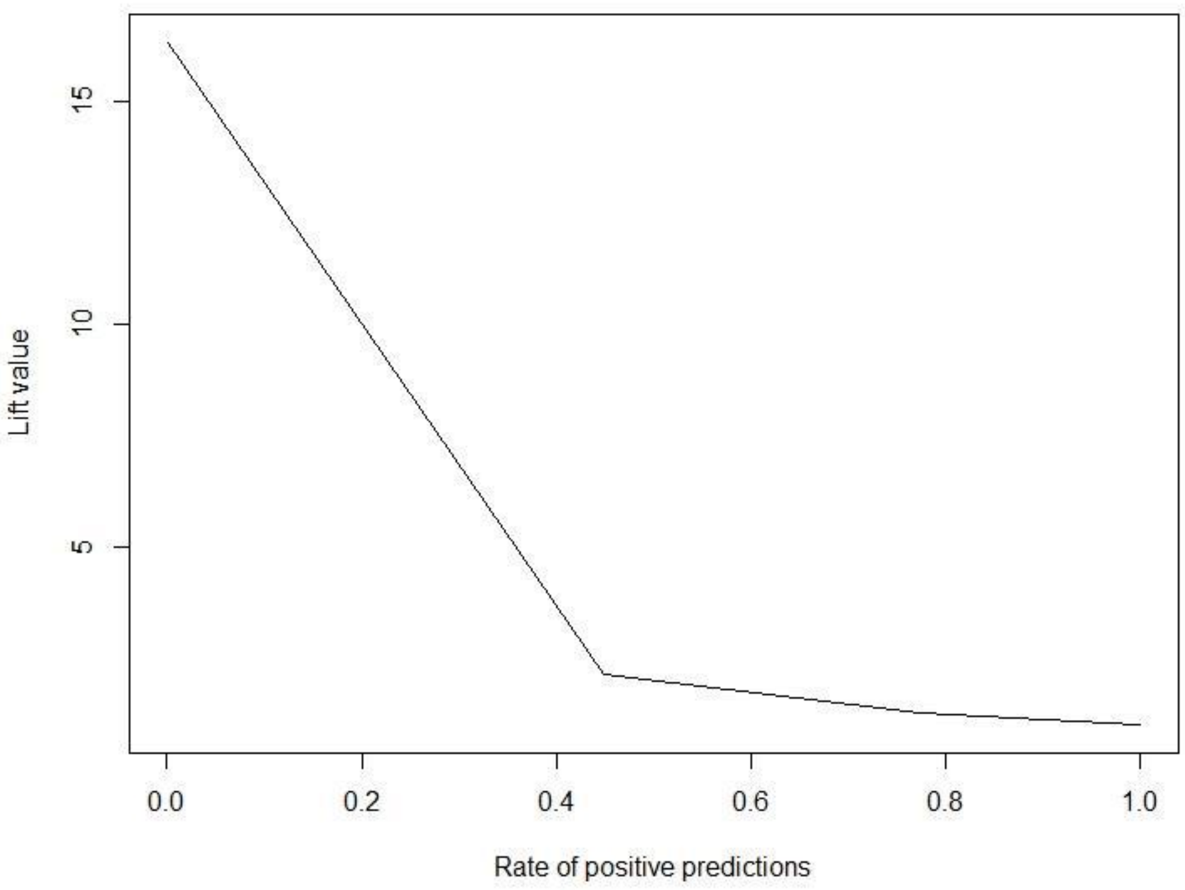

Fig 6. Lift Chart

Step 4: Plotting the Precision vs. Recall graph

Reference R Code:

```
pr<- performance(aae, "prec", "rec")
pr
plot(pr)
```

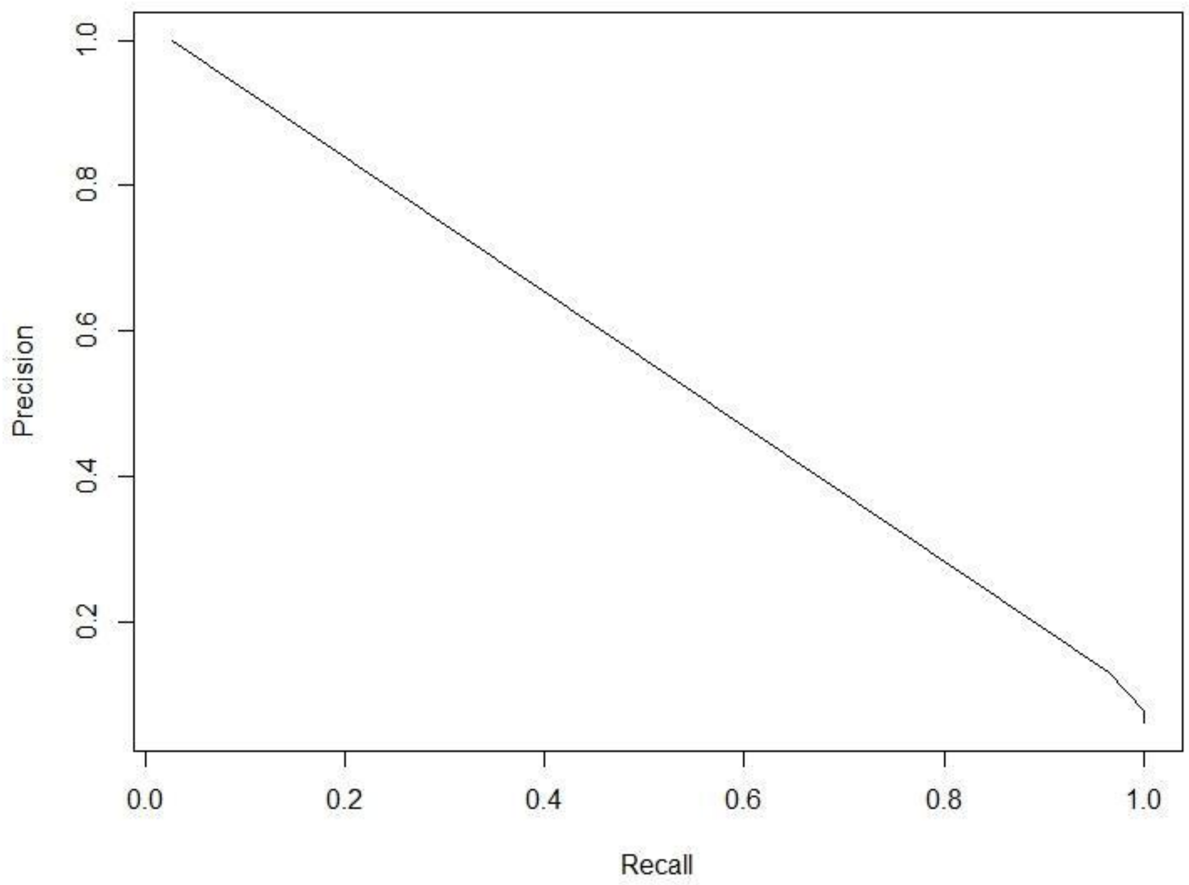

Fig 7. Precision vs. Recall Graph

Step 4: Prediction of Fraud on the new data set (claims_new1.csv) applying the Decision Tree Model

Reference R Code:

```
setwd("D:R")
abnew<-read.csv("claims_new1.csv",header=TRUE)
abnew
abnewpr<-predict(ab,abnew,type="class")
abnewpr
abnewpro<-predict(ab,abnew,type="prob")
abnewpro
sk<-data.frame(abnew,abnewpr)
write.csv(sk,"claims_new1_predict.csv")
```

The result of prediction is attached in the excel in the Annexure (Refer column “abnewpr”in the excel “claims_new1_predict.csv”)

Method 3 : Random Forests

Step 1: Running Decision Tree model with the all predictor variables

Reference R Code:

```
library(randomForest)

library(caTools)

setwd("D:R")

aa<-read.csv("claims1.csv",header = TRUE)

aa

set.seed(200)

split<-sample.split(aa,SplitRatio=.7)

aatr<-subset(aa,split==TRUE)

aatst<-subset(aa,split==FALSE)

head(aatr)

rf<-randomForest(FraudFound_P~.,data=aatr,method="class")

aapr1<-predict(rf,aatst,type="class")

aapr2<-ifelse(aapr1>=.5,1,0)

head(aapr2)

tb1<-table(aatst$FraudFound_P,aapr2)

tb1

mk<-sum(diag(tb1))/sum(tb1)

mk

1-mk
```

Inference/Output: The Accuracy of the Model is 0.9415918 or 94.16%

Step 2: Plotting the ROC Curve and deduction of the area under the curve

Reference R Code:

```
library(ROCR)
```

```
aapr3<-prediction(aapr1,aatst$FraudFound_P)

aapr4<-performance(aapr3,"tpr","fpr")

library(ggplot2)

plot(aapr4)

abline(a=0,b=1)

aapr5<-performance(aapr3,"auc")

aapr5
```

Inference/Output: Area under the ROC Curve is 89.28%. We can deduce from the OUTPUT of the above steps that the Random Forests model thus created is a good fit model.

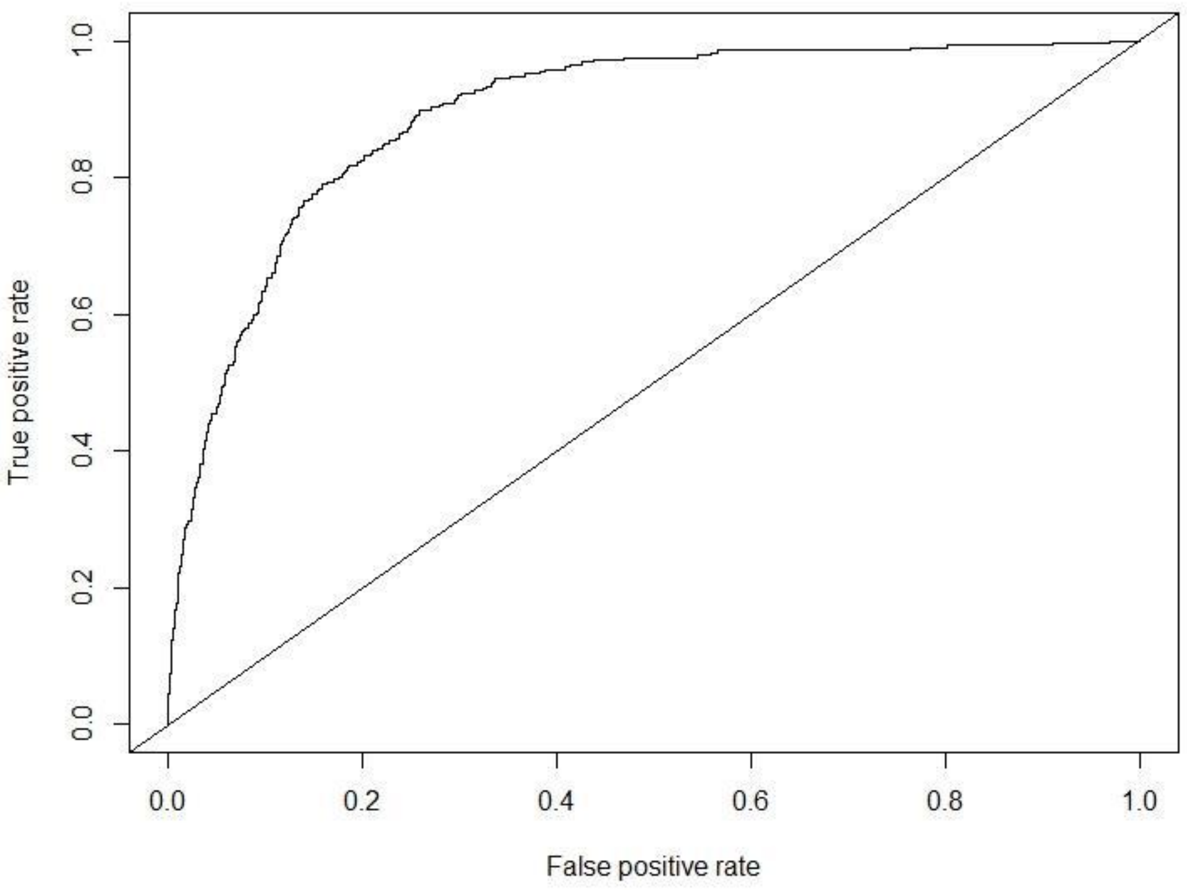

Fig 8. ROC Curve

Step 3: Plotting the Lift Chart to measure the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

Reference R Code:

```
library(lift)

lf<- performance(aapr3,"lift","rpp")

lf

plot(lf)
```

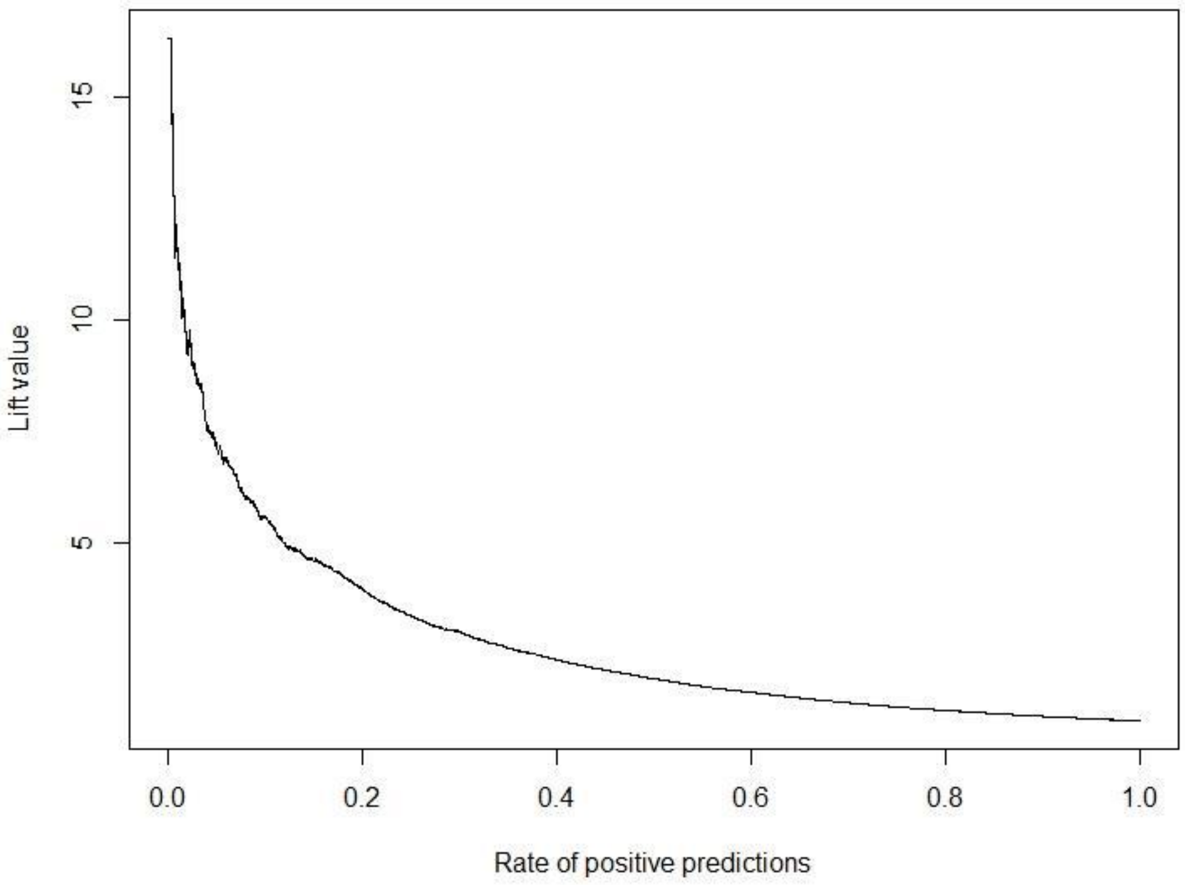

Fig 9. Lift Chart

Step 4: Plotting the Precision vs. Recall graph

Reference R Code:

```
pr<- performance(aapr3, "prec", "rec")
pr
plot(pr)
```

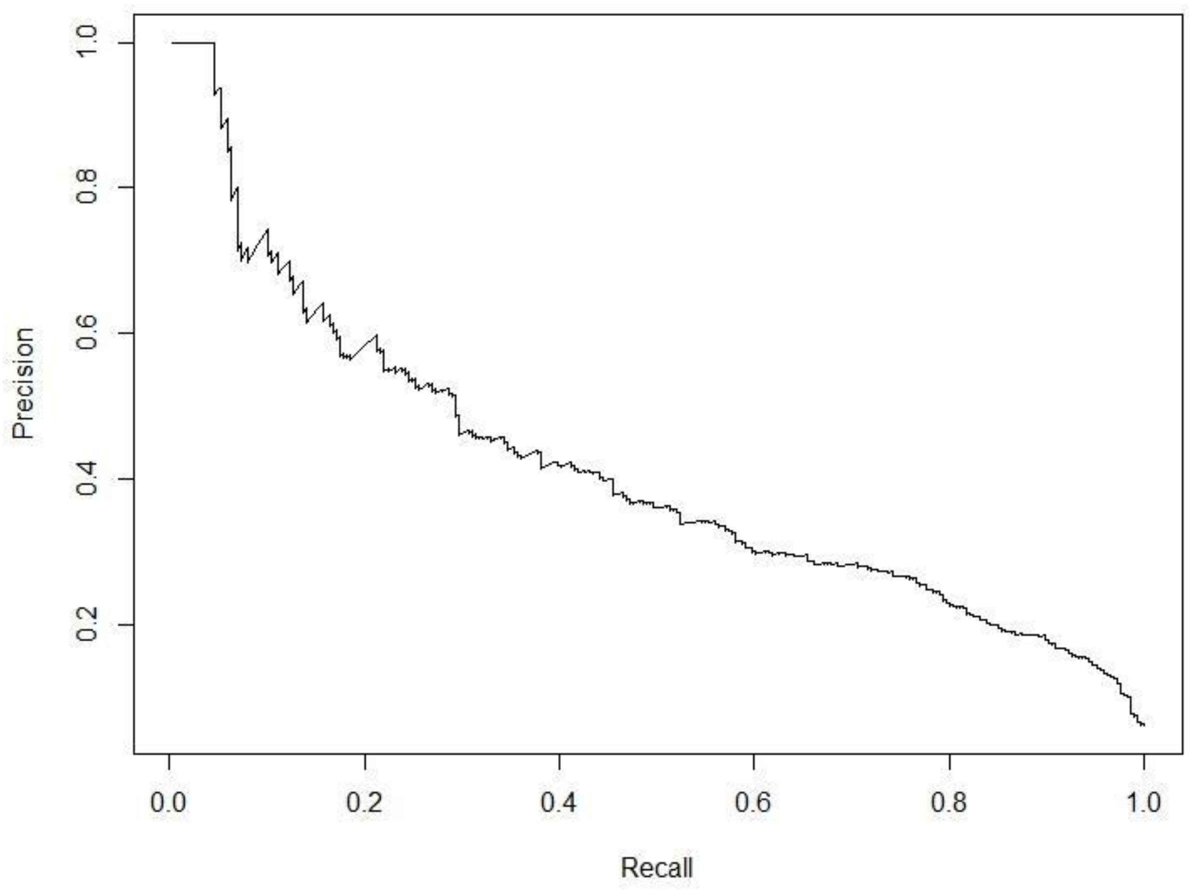

Fig 10. Precision vs. Recall Graph

Overall Inference: The Random Forests Model is the best predictive model out of Logistic Regression, Decision Tree & Random Forests.

ANNEXURE

1. **Insurance Data Set of XYZ Insurance Co**

claims1.csv

2. **Result of Prediction on new data set by Decision Tree Model**

claims_new1.csv

claims_new1_predict.csv

3. **R codes for Logistic Regression, Decision Tree, Random Forests**

R_Code_Logistic.txt

R_Code_Decision_Tree.txt

R_Code_Random_Forests.txt