library(caret)

#set working directory and reading data

setwd("F:/Ivy Proschool/Stats&R/Logistic Regression/Customer churn for telecom")

d=read.csv("dataset.csv")

head(d)

#data checking

dim(d)

str(d)

summary(d)

#detecting and removing missing values

sapply(d, function(x)sum(is.na(x)))

d[is.na(d$TotalCharges),]

nrow(d)

d<- na.omit(d)

nrow(d)

#converting necessary variables into factors

d$SeniorCitizen<- as.factor(d$SeniorCitizen)

d$Churn<- as.factor(d$Churn)

#Preparing training and test data

set.seed(113)

partitionData<- function(dataset=NULL,prob,target)

{

inTrain<- createDataPartition(y=dataset[,target],p=prob,list=FALSE)

inTrain

}

#import caret library to access createDataPartition() function.

library(caret)

train.sample<- partitionData(dataset = d,prob = 0.7,target = "Churn")

train.data<- d[train.sample,]

test.data<- d[-train.sample,]

nrow(train.data)

nrow(test.data)

#run logistic regression on full data

model<- glm(Churn~.,data = train.data,family = "binomial")

summary(model)

#considering all the variables

model1<- glm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+MultipleLines+InternetService+OnlineSecurity+

OnlineBackup+DeviceProtection+ TechSupport + StreamingTV + StreamingMovies + Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model1)

#only keeping multiplelines=yes

model2<- glm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+I(MultipleLines=="YES")+InternetService+OnlineSecurity+

OnlineBackup+DeviceProtection+ TechSupport + StreamingTV + StreamingMovies + Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model2)

#removing variable InternetService, StreamingMovie.

model3<- glm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+I(MultipleLines=="YES")+OnlineSecurity+

OnlineBackup+DeviceProtection+ TechSupport + StreamingTV + Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model3)

#removing variable Partner,StreamingTV

model3<- glm(Churn~gender+SeniorCitizen+Dependents+tenure+PhoneService+I(MultipleLines=="YES")+OnlineSecurity+

OnlineBackup+DeviceProtection+ TechSupport + Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model3)

#removing variable techsupport,Deviceprotection,OnlineBackup,MultipleLine

model4<- glm(Churn~gender+SeniorCitizen+Dependents+tenure+PhoneService+OnlineSecurity+

Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model4)

#removing variable gender

model5<- glm(Churn~SeniorCitizen+Dependents+tenure+PhoneService+OnlineSecurity+

Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model5)

#removing variable onlinesecurity and dependent

model6<- glm(Churn~SeniorCitizen+tenure+PhoneService+

Contract + PaperlessBilling +

PaymentMethod + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model6)

#Only keep payment method=ElectronicCheck

model6<- glm(Churn~SeniorCitizen+tenure+PhoneService+

Contract + PaperlessBilling +

I(PaymentMethod=="Electronic check") + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model6)

#Final modellticollinearity before checking mu

model6<- glm(Churn~SeniorCitizen+tenure+PhoneService+

Contract + PaperlessBilling +

I(PaymentMethod=="Electronic check") + MonthlyCharges + TotalCharges, data = train.data, family = "binomial")

summary(model6)

#Checking for multicollinearity

library(car)

vif(model6)

#Final modellticollinearity after checking mu- removing Total Charges due to collinearity

model6<- glm(Churn~SeniorCitizen+PhoneService+tenure+

Contract + PaperlessBilling +

I(PaymentMethod=="Electronic check") + MonthlyCharges , data = train.data, family = "binomial")

summary(model6)

vif(model6)

#---------------------------------------wald test------------------

library(aod)

wald.test(b=coef(model6),Sigma=vcov(model6),Terms=1:8)

#--------------------------------Lagrange Multiplier/score test

#null hypthesis is null model not equal to model6

modelchi<- model6$null.deviance- model6$deviance #diff bet null deviance and deviance

chidf<- model6$df.null-model6$df.residual

modelchi

chidf

chisq.prob<- 1-pchisq(modelchi,chidf)

format(round(chisq.prob,2),nsmall = 5)

#here P-value is less than 0.005 we can reject the null hypothesis

#-------------------------------Calculatinf R-Square

#Predicting power of model with R-square.Calculating R2

library(BaylorEdPsych)

PseudoR2(model6)

#-----------------------Calculating R-square Manually---------------------------

#Calculating R-Square with Hosmer-Lemeshow

R2.hl<- modelchi/model6$null.deviance

R2.hl

#Cox and Snell R-square.

R2.cs<- 1-exp((model6$deviance-model6$null.deviance)/nrow(train.data))

R2.cs

#Max-rescale R-Square(NagelKarke)

R2.max\_rescale<- R2.cs/(1-exp((model6$null.deviance/(nrow(train.data)))))

R2.max\_rescale

#--------------------Lackfit Deviance Test----------------------

residuals(model6)

residuals(model6,"pearson")

sum(residuals(model6,type="pearson")^2)

deviance(model6)

1-pchisq(deviance(model6),df.residual(model6))

#here P-value should be greater than 0.005 so that we can accept the null hypothesis is that

#Observed probability is equal to expected probability

#-------------------------- hosmer Lemeshow test

library(ResourceSelection)

hl<- hoslem.test(as.integer(train.data$Churn),fitted(model6),g=10)

hl

#------------------------interpretation

#coefficient(odd)

model6$coefficients

#coefficients(odds ratio)

exp(model6$coefficients)

#variable importance of model

varImp(model6)

#predicted probabilities

prediction<- predict(model6,newdata=train.data,type="response")

prediction

write.csv(prediction,"Prediction.csv")

#ROC Curve

library(pROC)

rocCurve<- roc(response=train.data$Churn,predictor = prediction,levels = rev(levels(train.data$Churn)))

rocCurve

plot(rocCurve)

#fit statistic

predclass<- ifelse(prediction>coords(rocCurve,"best",transpose = TRUE)[1],1,0)

confusion<- table(predicted=predclass,Actual=train.data$Churn)

Accuracy\_rate<- sum(diag(confusion)/sum(confusion))

Gini<- 2\*auc(rocCurve)-1

#Preparing AUC matrix

AUCMatrix<- data.frame(c(coords(rocCurve,"best",transpose = TRUE),AUC=auc(rocCurve),AccuracyRate=Accuracy\_rate,Gini=Gini))

AUCMatrix<- data.frame(rownames(AUCMatrix),AUCMatrix)

row.names(AUCMatrix)<- NULL

names(AUCMatrix)<- c("Metrix","Values")

AUCMatrix

#KS Statistic Calculation.

train.data$m1.yhat<- predict(model6,train.data,type="response")

library(ROCR)

m1.scores<- prediction(train.data$m1.yhat,train.data$Churn)

plot(performance(m1.scores,"tpr","fpr"),col="red")

abline(0,1,lty=8,col="gray")

m1.perf<- performance(m1.scores,"tpr","fpr")

ks1.logit <- max(attr(m1.perf, "y.values")[[1]] - (attr(m1.perf, "x.values")[[1]]))

ks1.logit

#-------Running Logistics Regression on Test Data------

model\_test<- glm(Churn~SeniorCitizen+PhoneService+tenure+

Contract + PaperlessBilling +

I(PaymentMethod=="Electronic check") + MonthlyCharges , data = test.data, family = "binomial")

summary(model\_test)

vif(model\_test)

#---------------------------------------wald test

wald.test(b=coef(model\_test),Sigma=vcov(model\_test),Terms=1:8)

#--------------------------------Lagrange Multiplier/score test

#null hypthesis is null model not equal to model6

test\_modelchi<- model\_test$null.deviance- model\_test$deviance #diff bet null deviance and deviance

test\_chidf<- model\_test$df.null-model\_test$df.residual

test\_modelchi

test\_chidf

test\_chisq.prob<- 1-pchisq(test\_modelchi,test\_chidf)

format(round(test\_chisq.prob,2),nsmall = 5)

#here P-value is less than 0.005 we can reject the null hypothesis

#-------------------------------Calculatinf R-Square

#Predicting power of model with R-square.Calculating R2

PseudoR2(model\_test)

#-----------------------Calculating R-square Manually---------------------------

#Calculating R-Square with Hosmer-Lemeshow

R2.hl\_test<- test\_modelchi/model\_test$null.deviance

R2.hl\_test

#Cox and Snell R-square.

R2.cs\_test<- 1-exp((model\_test$deviance-model\_test$null.deviance)/nrow(test.data))

R2.cs\_test

#Max-rescale R-Square(NagelKarke)

R2.max\_rescale\_test<- R2.cs/(1-exp((model\_test$null.deviance/(nrow(test.data)))))

R2.max\_rescale\_test

#--------------------Lackfit Deviance Test----------------------

residuals(model\_test)

residuals(model\_test,"pearson")

sum(residuals(model\_test,type="pearson")^2)

deviance(model\_test)

1-pchisq(deviance(model\_test),df.residual(model\_test))

#here P-value should be greater than 0.005 so that we can accept the null hypothesis is that

#Observed probability is equal to expected probability

#-------------------------- hosmer Lemeshow test

hl\_test<- hoslem.test(as.integer(test.data$Churn),fitted(model\_test),g=10)

hl\_test

#------------------------interpretation

#coefficient(odd)

model\_test$coefficients

#coefficients(odds ratio)

exp(model\_test$coefficients)

#variable importance of model

varImp(model\_test)

#predicted probabilities

prediction\_test<- predict(model\_test,newdata=test.data,type="response")

prediction\_test

write.csv(prediction\_test,"Prediction\_test.csv")

#ROC Curve

test\_rocCurve<- roc(response=test.data$Churn,predictor = prediction\_test,levels = rev(levels(test.data$Churn)))

test\_rocCurve

plot(test\_rocCurve)

#fit statistic

test\_predclass<- ifelse(prediction\_test>coords(test\_rocCurve,"best",transpose = TRUE)[1],1,0)

test\_confusion<- table(predicted=test\_predclass,Actual=test.data$Churn)

test\_Accuracy\_rate<- sum(diag(test\_confusion)/sum(test\_confusion))

test\_Gini<- 2\*auc(test\_rocCurve)-1

#Preparing AUC matrix

test\_AUCMatrix<- data.frame(c(coords(test\_rocCurve,"best",transpose = TRUE),AUC=auc(test\_rocCurve),AccuracyRate=test\_Accuracy\_rate,test\_Gini=Gini))

test\_AUCMatrix<- data.frame(rownames(test\_AUCMatrix),test\_AUCMatrix)

row.names(test\_AUCMatrix)<- NULL

names(test\_AUCMatrix)<- c("Metrix","Values")

test\_AUCMatrix

#KS Statistic Calculation.

test.data$m1.yhat<- predict(model\_test,test.data,type="response")

m1.scores<- prediction(test.data$m1.yhat,test.data$Churn)

plot(performance(m1.scores,"tpr","fpr"),col="red")

abline(0,1,lty=8,col="gray")

m1.perf<- performance(m1.scores,"tpr","fpr")

ks1.logit <- max(attr(m1.perf, "y.values")[[1]] - (attr(m1.perf, "x.values")[[1]]))

ks1.logit

#-----------------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*------------------------------------------------