#include libraries

library(boot)

library(car)

library(QuantPsyc)

library(lmtest)

library(sandwich)

library(vars)

library(nortest)

library(MASS)

library(ggplot2)

#setworking directory

setwd("F:/Ivy Proschool/Stats&R/Linear regression/Ivyproject/R\_Dataset/LM\_Project1")

#read the data dile

d1<- read.csv("Data-Copy.csv",header = TRUE)

#removing Customer ID and Location as it doesn't make any sense in predicting the value.

d2=subset(d1,select=-c(Customer,Effective.To.Date))

data<- d2

summary(data)

#detecting and removing outlier

num=data[sapply(data,is.numeric)]

summary(num)

str(num)

ggplot(data=num,aes(x="",y=Customer.Lifetime.Value,col="Red"))+geom\_boxplot()

bench<- 8962+1.5\*IQR(num$Customer.Lifetime.Value)

num=num[num$Customer.Lifetime.Value<bench,]

ggplot(data=num,aes(x="",y=Customer.Lifetime.Value,col="Red"))+geom\_boxplot()

summary(num)

bench1<-8163+1.5\*IQR(num$Customer.Lifetime.Value)

num=num[num$Customer.Lifetime.Value<bench1,]

ggplot(data=num,aes(x="",y=Customer.Lifetime.Value,col="Red"))+geom\_boxplot()

summary(num)

bench3<- 8011+1.5\*IQR(num$Customer.Lifetime.Value)

data=data[data$Customer.Lifetime.Value<bench3,]

ggplot(data = data,aes(x="",y=Customer.Lifetime.Value,col="Green"))+geom\_boxplot()

ggplot(data=num,aes(x="",y=Income,col="Red"))+geom\_boxplot() #No outliers

#Outliers in Monthly premium auto

ggplot(data=num,aes(x="",y=Monthly.Premium.Auto,col="Red"))+geom\_boxplot()

summary(num)

bench<- 109+1.5\*IQR(num$Monthly.Premium.Auto)

num=num[num$Monthly.Premium.Auto<bench,]

ggplot(data=num,aes(x="",y=Monthly.Premium.Auto,col="Red"))+geom\_boxplot()

summary(num)

bench1<-106+1.5\*IQR(num$Monthly.Premium.Auto)

num=num[num$Monthly.Premium.Auto<bench1,]

ggplot(data = num,aes(x="",y=Monthly.Premium.Auto,col="Red"))+geom\_boxplot()

data=data[data$Monthly.Premium.Auto<bench1,]

ggplot(data = data,aes(x="",y=Monthly.Premium.Auto,col="Green"))+geom\_boxplot()

ggplot(data = num,aes(x="",y=Months.Since.Last.Claim,col="Red"))+geom\_boxplot()#No outliers

ggplot(data = num,aes(x="",y=Months.Since.Policy.Inception,col="Red"))+geom\_boxplot()#No outliers

ggplot(data = num,aes(x="",y=Number.of.Open.Complaints,col="Red"))+geom\_boxplot()#No outliers

#Outliers in Number of policies

ggplot(data = num,aes(x="",y=Number.of.Policies,col="Red"))+geom\_boxplot()

summary(num)

bench2<- 4+1.5\*IQR(num$Number.of.Policies)

num=num[num$Number.of.Policies<bench2,]

ggplot(data = num,aes(x="",y=Number.of.Policies,col="Red"))+geom\_boxplot()

summary(num)

bench3<- 3+1.5\*IQR(num$Number.of.Policies)

num=num[num$Number.of.Policies<bench3,]

ggplot(data = num,aes(x="",y=Number.of.Policies,col="Red"))+geom\_boxplot()

data=data[data$Number.of.Policies<bench3,]

ggplot(data = data,aes(x="",y=Number.of.Policies,col="Green"))+geom\_boxplot()

#outliers in Total Cliam Amount variable

ggplot(data = num,aes(x="",y=Total.Claim.Amount,col="Red"))+geom\_boxplot()

summary(num)

bench4<- 528+1.5\*IQR(num$Total.Claim.Amount)

num=num[num$Total.Claim.Amount<bench4,]

ggplot(data = num,aes(x="",y=Total.Claim.Amount,col="Red"))+geom\_boxplot()

summary(num)

bench5<- 513.60+1.5\*IQR(num$Total.Claim.Amount)

num=num[num$Total.Claim.Amount<bench5,]

ggplot(data = num,aes(x="",y=Total.Claim.Amount,col="Red"))+geom\_boxplot()

data=data[data$Total.Claim.Amount<bench5,]

ggplot(data = data,aes(x="",y=Total.Claim.Amount,col="Red"))+geom\_boxplot()

data$Marital.Status.Dummy=ifelse(data$Marital.Status=="Divorced",0,ifelse(data$Marital.Status=="Married",1,2))

data$Location.Code.Dummy=ifelse(data$Location.Code=="Rural",0,ifelse(data$Location.Code=="Suburban",1,2))

data$Coverage.Dummy=ifelse(data$Coverage=="Basic",0,ifelse(data$Coverage=="Extended",1,2))

data$Vehicle.Class.Dummy=ifelse(data$Vehicle.Class=="Four-Door Car",0,ifelse(data$Vehicle.Class=="Luxury Car",1,ifelse(data$Vehicle.Class=="Luxury SUV",2,ifelse(data$Vehicle.Class=="Sports Car",3,ifelse(data$Vehicle.Class=="SUV",4,5)))))

data$Policy.Type.Dummy=ifelse(data$Policy.Type=="Corporate Auto",0,ifelse(data$Policy.Type=="Personal Auto",1,2))

data$Education.Dummy=ifelse(data$Education=="Bachelor",0,ifelse(data$Education=="College",1,ifelse(data$Education=="Doctor",2,ifelse(data$Education=="High School or Below",3,4))))

data$Gender\_Dummy=ifelse(data$Gender=="M",1,0)

data$EmploymentStatus.Dummy=ifelse(data$EmploymentStatus=="Disabled",0,ifelse(data$EmploymentStatus=="Employed",1,ifelse(data$EmploymentStatus=="Medical Leave",2,ifelse(data$EmploymentStatus=="Retired",3,4))))

data$Renew.Offer.Type.Dummy=ifelse(data$Renew.Offer.Type=="Offer1",0,ifelse(data$Renew.Offer.Type=="Offer2",1,ifelse(data$Renew.Offer.Type=="Offer3",2,3)))

data$Sales.Channel.Dummy=ifelse(data$Sales.Channel=="Agent",0,ifelse(data$Sales.Channel=="Branch",1,ifelse(data$Sales.Channel=="Call Center",2,3)))

data$Vehicle.Size.Dummy=ifelse(data$Vehicle.Size=="Large",0,ifelse(data$Vehicle.Size=="Medsize",1,2))

data$State.Dummy=ifelse(data$State=="Arizona",0,ifelse(data$State=="California",1,ifelse(data$State=="Nevada",2,ifelse(data$State=="Oregon",3,4))))

data$Response.Dummy=ifelse(data$Response=="Yes",1,0)

#data=data[,-c(8)]

#data=data[,-c(9)]

#data=data[,-c(3)]

#data=data[,-c(4)]

#data Exploration check data structure and summary

str(data)

summary(data)

#sapply(data,is.numeric)

#data$Income=as.numeric(as.character(data$Income))

#to check if data is populated/imported properly

head(data)

tail(data)

summary(data)

write.csv(data,"data1.csv")

data1<- read.csv("data1.csv")

head(data1)

#splitting data into train and test data

#samp<- sample(2,nrow(data1), replace=TRUE, prob=c(0.8,0.2))

data1\_dummy<- data1

library(caret)

set.seed(123)

partition\_data<-createDataPartition(data1\_dummy$Customer.Lifetime.Value, p=.7,list = FALSE)

train.data<- data1\_dummy[partition\_data,]

test.data<- data1\_dummy[-partition\_data,]

#set.seed(1)

#split\_data<- split(data1,sample(1:nrow(data)>round(nrow(data)\*.7)))

#train.data<- split\_data$"False"

#test.data<- split\_data$"True"

head(test.data)

head(train.data)

#fit<- lm(formula=1/(Customer.Lifetime.Value)~Income+EmploymentStatus.Dummy+Renew.Offer.Type+Monthly.Premium.Auto+

# Number.of.Open.Complaints+Number.of.Policies+Coverage+I(Vehicle.Class=="Sports Car")+I(Vehicle.Class=="SUV"),data = train.data)

#initial model

fit<- lm(formula=1/(Customer.Lifetime.Value)~ State+Response+Coverage+Education+EmploymentStatus+Gender+Income+Location.Code+Marital.Status+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Policy.Type+Policy+Renew.Offer.Type+Sales.Channel+Total.Claim.Amount+Vehicle.Class+Vehicle.Size,data = train.data)

summary(fit)

#version-2 Removing Policy,Vehicle size.

fit<- lm(formula=1/(Customer.Lifetime.Value)~ State+Response+Coverage+Education+EmploymentStatus+Gender+Income+Location.Code+Marital.Status+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Policy.Type+Renew.Offer.Type+Sales.Channel+Total.Claim.Amount+Vehicle.Class,data = train.data)

summary(fit)

#version-3 Removing PolicyType,Vehicle Class,Marital status,Gender

fit<- lm(formula=1/(Customer.Lifetime.Value)~ State+Response+Coverage+Education+EmploymentStatus+Income+Location.Code+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Sales.Channel+Total.Claim.Amount,data = train.data)

summary(fit)

#version-4 Removing State,Response,Education,Location Code

fit<- lm(formula=1/(Customer.Lifetime.Value)~ Coverage+EmploymentStatus+Income+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Sales.Channel+Total.Claim.Amount,data = train.data)

summary(fit)

#version-5 Removing Employment status Employeed keep rest as it is.

fit<- lm(formula=1/(Customer.Lifetime.Value)~ Coverage+I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Sales.Channel+Total.Claim.Amount,data = train.data)

summary(fit)

#version-6 Removing Sales Channel only keep Web Sales channel because it's having los P-Value

fit<- lm(formula=1/(Customer.Lifetime.Value)~ Coverage+I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+I(Sales.Channel=="Web")+Total.Claim.Amount,data = train.data)

summary(fit)

#version-7 Keeping Coverage premium as it's having low P-value

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(Coverage=="Premium")+I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Monthly.Premium.Auto+Months.Since.Last.Claim+Months.Since.Policy.Inception+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+I(Sales.Channel=="Web")+Total.Claim.Amount,data = train.data)

summary(fit)

#version-8 Removing Months.since,last claim , months.since policyinception and caoverage as it's at low p-value still.

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Total.Claim.Amount,data = train.data)

summary(fit)

vif(fit)

#version-9 Adding dummy of gender or Marital status and examining R-squared.

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Marital.Status.Dummy+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Total.Claim.Amount,data = train.data)

summary(fit)

vif(fit)

#version-10 Adding dummy of coverage,education,Location code and examining R-squared

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Marital.Status.Dummy+Coverage.Dummy+Location.Code.Dummy+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Total.Claim.Amount,data = train.data)

summary(fit)

vif(fit)

#version-11 Adding dummy of state,response,Vehicle class,Vehicle size and examining R-squared

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Marital.Status.Dummy+Coverage.Dummy+Location.Code.Dummy+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Total.Claim.Amount+Vehicle.Class.Dummy,data = train.data)

summary(fit)

vif(fit)

#version-12 Adding dummy of Policy type and examining R-squared

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Income+Marital.Status.Dummy+Coverage.Dummy+Location.Code.Dummy+

Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Total.Claim.Amount+Vehicle.Class.Dummy+Policy.Type.Dummy,data = train.data)

summary(fit)

vif(fit)

#version-11 Final model

fit<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Monthly.Premium.Auto,data = train.data)

summary(fit)

vif(fit)

#get the predicted values

fitted(fit)

#MAPE

train.data$pred<- fitted(fit)

write.csv(train.data,"mape.csv")

#Calculating MAPE calculating deviation from actual observation.

attach(train.data)

(sum((abs(Customer.Lifetime.Value-pred))/Customer.Lifetime.Value))/nrow(train.data)

detach(train.data)

#Residual Analysis

res<- train.data

res$stu\_res<- studres(fit)# Studentized residual

res$stud.del.resids<- rstudent(fit) # studentized deleted residual

res$leverage<- hatvalues(fit)# leverage values

res$cooks\_des<- cooks.distance(fit) # cook's distance

res$dffits<-dffits(fit) # DFFITS

res$dfbetas<- dfbeta(fit) # DF betas

res$cov\_ratio <- covratio(fit) #covariance ratio

write.csv(res,"Res.csv")

####Checking of Assumptions.

#Durbinwatson test Checking autocorrelation. (Assumption-1, Satisfied assumption )

durbinWatsonTest(fit)

dwt(fit)

#VIF test (Assumption-2, Satisfied assumption )

vif(fit)

####checking constant error variance. (Assumption-3 Voilated)

#Bruech-pagan test

bptest(fit)

#cook-weisberg test

ncvTest(lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Monthly.Premium.Auto,data = train.data))

####Normality test

#Anderson-darling test for normality (Assumption-4 Voilated)

resids<- fit$residuals

ad.test(resids)

#lilliefors(Kolmogorov-smirnov) normality test

lillie.test(resids)

#pearson chi-square test for normality

pearson.test(resids)

#Shapiro Francia test for normality

sf.test(resids)

#QQ Plot for residual analysis

qqnorm(resids)

detach(train.data)

############################TESTING THE MODEL ON TEST DATA#######################################

fit1<- lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Monthly.Premium.Auto,data = test.data)

summary(fit1)

#checking for multicollinearity

vif(fit1)

#get the predicted values

fitted(fit1)

#MAPE

test.data$pred<- fitted(fit1)

write.csv(test.data,"mape1.csv")

attach(test.data)

(sum((abs(Customer.Lifetime.Value-pred))/Customer.Lifetime.Value))/nrow(test.data)

detach(test.data)

#Residual Analysis

res1<- test.data

res1$stu\_res<- studres(fit1)# Studentized residual

res1$stud.del.resids<- rstudent(fit1) # studentized deleted residual

res1$leverage<- hatvalues(fit1)# leverage values

res1$cooks\_des<- cooks.distance(fit1) # cook's distance

res1$dffits<-dffits(fit1) # DFFITS

res1$dfbetas<- dfbeta(fit1) # DF betas

res1$cov\_ratio <- covratio(fit1) #covariance ratio

write.csv(res1,"Res1.csv")

####Checking of Assumptions.

#Durbinwatson test (Assumption-1 Satisfied)

durbinWatsonTest(fit1)

dwt(fit1)

#VIF test (Assumption-2 Satisfied)

vif(fit1)

####checking constant error variance.

#Bruech-pagan test (Assumption-3 Voilated)

bptest(fit1)

#cook-weisberg test

ncvTest(lm(formula=1/(Customer.Lifetime.Value)~ I(EmploymentStatus=="Medical Leave")+I(EmploymentStatus=="Retired")+I(EmploymentStatus=="Unemployed")+Number.of.Open.Complaints+Number.of.Policies+Renew.Offer.Type+Monthly.Premium.Auto,data = test.data))

####Normality test

#Anderson-darling test for normality (Assumption-4 Voilated)

resids1<- fit1$residuals

ad.test(resids1)

#lilliefors(Kolmogorov-smirnov) normality test

lillie.test(resids1)

#pearson chi-square test for normality

pearson.test(resids1)

#Shapiro Francia test for normality

sf.test(resids1)

#QQ Plot for residual analysis

qqnorm(resids1)