**Ensemble Models- regression and Time-Series**

For regression models:

* For each UID, first MSE is calculated by dividing data such that training set consist of data up to 18Q2 for which actuals are present and testing set contains data of 18Q3, 18Q4 and 19Q1 whose actuals are also present so that we can calculate MSE for each models and find the best model for each UID.
* After finding best model for each UID, predictions for 19Q2 and 19Q3 (for which actuals are not present) is done for each UID by best model.

Then done predictions from time-series model.

After we get predictions from best regression and time-series model, take weighted average of predictions of both models that results in final prediction.

**Code:**

**#ensemble average of regression and time-series model**

**# with auto arima**

library(glmnet)

library(party)

library(randomForest)

library(forecast)

u1<-unique(data2[1])

n1<-nrow(u1)

df1=data.frame(UID=character(0),MSE\_LR=numeric(0),MSE\_Lasso=numeric(0),MSE\_Ridge=numeric(0),MSE\_DT=numeric(0),MSE\_RF=numeric(0),BEST=character(0),Regression\_Pred\_19Q2=numeric(0),Regreesion\_Pred\_19Q3=numeric(0),Time\_Series\_Pred\_19Q2=numeric(0),Time\_Series\_Pred\_19Q3=numeric(0),Final\_Pred\_19Q2=numeric(0),Final\_Pred\_19Q3=numeric(0))

i=1

while (i<=n1){

ntrain1<-subset(data2,UID==u1[1,] & (Period=='14-Q1'| Period=='14-Q2'| Period=='14-Q3'| Period=='14-Q4'| Period=='15-Q1'| Period=='15-Q2'| Period=='15-Q3'| Period=='15-Q4'| Period=='16-Q1'| Period=='16-Q2'| Period=='16-Q3'| Period=='16-Q4'| Period=='17-Q1'| Period=='17-Q2'| Period=='17-Q3'| Period=='17-Q4'| Period=='18-Q1'| Period=='18-Q2'),select=c(11:32))

ntest1<-subset(data2,UID==u1[1,] & (Period=='18-Q3'| Period=='18-Q4'| Period=='19-Q1'),select=c(11:32))

#linear regression model

lr\_model<- lm(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1)

new\_test1<-data.frame(ntest1[c(2:22)])

pred\_lr<-predict(lr\_model,newdata=new\_test1)

mse\_lr<-mean((ntest1$Actuals-pred\_lr)^2)

# Ridge

x\_train<-as.matrix(ntrain1[2:22])

y\_train<-(ntrain1$Actuals)

y\_train<-jitter(y\_train,factor = 1,amount=NULL)

x\_test<-as.matrix(ntest1[c(2:22)])

y\_test<-(ntest1$Actuals)

alpha0.fit<-cv.glmnet(x\_train,y\_train,type.measure="mse",alpha=0,grouped = FALSE)

alpha0.predicted<-predict(alpha0.fit,s=alpha0.fit$lambda.1se,newx=x\_test)

mse\_r<-mean((y\_test-alpha0.predicted)^2)

# lasso

alpha1.fit<-cv.glmnet(x\_train,y\_train,type.measure="mse",alpha=1,grouped = FALSE)

alpha1.predicted<-predict(alpha1.fit,s=alpha1.fit$lambda.1se,newx=x\_test)

mse\_l<-mean((y\_test-alpha1.predicted)^2)

# DT

tree<-ctree(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1,controls=ctree\_control(mincriterion = 0.9))

p<-predict(tree,ntest1[2:22])

mse\_DT=mse(ntest1[[1]],p)

# RF

rf<-randomForest(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1)

p1<-predict(rf,ntest1[2:22])

mse\_RF=mse(ntest1[[1]],p1)

df2<-data.frame(UID=u1[i,],MSE\_LR=mse\_lr,MSE\_Lasso=mse\_l,MSE\_Ridge=mse\_r,MSE\_DT=mse\_DT,MSE\_RF=mse\_RF,BEST='',Regression\_Pred\_19Q2='',Regression\_Pred\_19Q3='',Time\_Series\_Pred\_19Q2='',Time\_Series\_Pred\_19Q3='',Final\_Pred\_19Q2='',Final\_Pred\_19Q3='')

d<-colnames(df2[2:6])[which.min(df2[2:6])]

df2$BEST<-sub('MSE\_','',d)

#for predicting actuals for 19Q2 and 19Q3

ntrainp<-subset(data2,UID==u1[i,] & (Period=='14-Q1'| Period=='14-Q2'| Period=='14-Q3'| Period=='14-Q4'| Period=='15-Q1'| Period=='15-Q2'| Period=='15-Q3'| Period=='15-Q4'| Period=='16-Q1'| Period=='16-Q2'| Period=='16-Q3'| Period=='16-Q4'| Period=='17-Q1'| Period=='17-Q2'| Period=='17-Q3'| Period=='17-Q4'| Period=='18-Q1'| Period=='18-Q2'| Period=='18-Q3'| Period=='18-Q4'| Period=='19-Q1'),select=c(11:32))

ntestp<-subset(data2,UID==u1[i,] & (Period=='19-Q2'| Period=='19-Q3'),select=c(11:32))

x\_trainp<-as.matrix(ntrainp[2:22])

y\_trainp<-(ntrainp$Actuals)

y\_trainp<-jitter(y\_trainp,factor = 1,amount=NULL)

x\_testp<-as.matrix(ntestp[c(2:22)])

y\_testp<-(ntestp$Actuals)

if(df2$BEST=='LR'){

modelp<- lm(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp)

new\_testp<-data.frame(ntestp[c(2:22)])

predp<-predict(modelp,newdata=new\_testp)

df2$Regression\_Pred\_19Q2<-predp[[1]]

df2$Regression\_Pred\_19Q3<-predp[[2]]

} else if(df2$BEST=='Lasso'){

alpha1.fitp<-cv.glmnet(x\_trainp,y\_trainp,type.measure="mse",alpha=1,grouped = FALSE)

alpha1.predictedp<-predict(alpha1.fitp,s=alpha1.fitp$lambda.1se,newx=x\_testp)

df2$Regression\_Pred\_19Q2<-alpha1.predictedp[[1]]

df2$Regression\_Pred\_19Q3<-alpha1.predictedp[[2]]

} else if(df2$BEST=='Ridge'){

alpha0.fitp<-cv.glmnet(x\_trainp,y\_trainp,type.measure="mse",alpha=0,grouped = FALSE)

alpha0.predictedp<-predict(alpha0.fitp,s=alpha0.fitp$lambda.1se,newx=x\_testp)

df2$Regression\_Pred\_19Q2<-alpha0.predictedp[[1]]

df2$Regression\_Pred\_19Q3<-alpha0.predictedp[[2]]

} else if(df2$BEST=='DT'){

tree1<-ctree(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp,controls=ctree\_control(mincriterion = 0.9))

pt<-predict(tree1,ntestp[2:22])

df2$Regression\_Pred\_19Q2<-pt[[1]]

df2$Regression\_Pred\_19Q3<-pt[[2]]

} else {

rf1<-randomForest(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp)

pf<-predict(rf1,ntestp[2:22])

df2$Regression\_Pred\_19Q2<-pf[[1]]

df2$Regression\_Pred\_19Q3<-pf[[2]]

}

# predictions from arima

xi1<-ts(ntrainp$Actuals,frequency=4,start=c(2014,1))

pi1<-auto.arima(xi1)

qi1<-forecast(pi1,h=2)

q2<-data.frame(qi1)

df2$Time\_Series\_Pred\_19Q2<-q2$Point.Forecast[[1]]

df2$Time\_Series\_Pred\_19Q3<-q2$Point.Forecast[[2]]

#ensemble of regression and time-series

df2$Final\_Pred\_19Q2<-0.5\*df2$Regression\_Pred\_19Q2+0.5\*df2$Time\_Series\_Pred\_19Q2

df2$Final\_Pred\_19Q3<-0.5\*df2$Regression\_Pred\_19Q3+0.5\*df2$Time\_Series\_Pred\_19Q3

df1<-rbind(df1,df2)

i=i+1

}

df1

write\_xlsx(df1,"C:\\Users\\saloni\\Downloads\\Final\_Modeling\_output.xlsx")

**Code:**

**# best among 6 models**

**# with auto arima**

library(glmnet)

library(party)

library(randomForest)

u1<-unique(data2[1])

n1<-nrow(u1)

n1<-2

df1=data.frame(UID=character(0),MSE\_LR=numeric(0),MSE\_Lasso=numeric(0),MSE\_Ridge=numeric(0),MSE\_DT=numeric(0),MSE\_RF=numeric(0),MSE\_Arima=numeric(0),BEST=character(0),Pred\_19Q2=numeric(0),Pred\_19Q3=numeric(0))

i=1

while (i<=n1){

ntrain1<-subset(data2,UID==u1[i,] & (Period=='14-Q1'| Period=='14-Q2'| Period=='14-Q3'| Period=='14-Q4'| Period=='15-Q1'| Period=='15-Q2'| Period=='15-Q3'| Period=='15-Q4'| Period=='16-Q1'| Period=='16-Q2'| Period=='16-Q3'| Period=='16-Q4'| Period=='17-Q1'| Period=='17-Q2'| Period=='17-Q3'| Period=='17-Q4'| Period=='18-Q1'| Period=='18-Q2'),select=c(11:32))

ntest1<-subset(data2,UID==u1[i,] & (Period=='18-Q3'| Period=='18-Q4'| Period=='19-Q1'),select=c(11:32))

#linear regression model

lr\_model<- lm(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1)

new\_test1<-data.frame(ntest1[c(2:22)])

pred\_lr<-predict(lr\_model,newdata=new\_test1)

mse\_lr<-mean((ntest1$Actuals-pred\_lr)^2)

# Ridge

x\_train<-as.matrix(ntrain1[2:22])

y\_train<-(ntrain1$Actuals)

y\_train<-jitter(y\_train,factor = 1,amount=NULL)

x\_test<-as.matrix(ntest1[c(2:22)])

y\_test<-(ntest1$Actuals)

alpha0.fit<-cv.glmnet(x\_train,y\_train,type.measure="mse",alpha=0,grouped = FALSE)

alpha0.predicted<-predict(alpha0.fit,s=alpha0.fit$lambda.1se,newx=x\_test)

mse\_r<-mean((y\_test-alpha0.predicted)^2)

# lasso

alpha1.fit<-cv.glmnet(x\_train,y\_train,type.measure="mse",alpha=1,grouped = FALSE)

alpha1.predicted<-predict(alpha1.fit,s=alpha1.fit$lambda.1se,newx=x\_test)

mse\_l<-mean((y\_test-alpha1.predicted)^2)

# DT

tree<-ctree(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1,controls=ctree\_control(mincriterion = 0.9))

p<-predict(tree,ntest1[2:22])

mse\_DT=mse(ntest1[[1]],p)

# RF

rf<-randomForest(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrain1)

p1<-predict(rf,ntest1[2:22])

mse\_RF=mse(ntest1[[1]],p1)

# Arima

xi<-ts(ntrain1$Actuals,frequency=4,start=c(2014,1))

pi<-auto.arima(xi)

q<-forecast(pi,h=3)

q1<-data.frame(q)

y1<-subset(ntest1,select=c(1))

row.names(y1)<-c("2018 Q3","2018 Q4","2019 Q1")

mse\_a<-mean((y1$Actuals-q1$Point.Forecast)^2)

df2<-data.frame(UID=u1[i,],MSE\_LR=mse\_lr,MSE\_Lasso=mse\_l,MSE\_Ridge=mse\_r,MSE\_DT=mse\_DT,MSE\_RF=mse\_RF,MSE\_Arima=mse\_a,BEST='',Pred\_19Q2='',Pred\_19Q3='')

d<-colnames(df2[2:7])[which.min(df2[2:7])]

df2$BEST<-sub('MSE\_','',d)

df2

#for predicting actuals for 19Q2 and 19Q3

ntrainp<-subset(data2,UID==u1[i,] & (Period=='14-Q1'| Period=='14-Q2'| Period=='14-Q3'| Period=='14-Q4'| Period=='15-Q1'| Period=='15-Q2'| Period=='15-Q3'| Period=='15-Q4'| Period=='16-Q1'| Period=='16-Q2'| Period=='16-Q3'| Period=='16-Q4'| Period=='17-Q1'| Period=='17-Q2'| Period=='17-Q3'| Period=='17-Q4'| Period=='18-Q1'| Period=='18-Q2'| Period=='18-Q3'| Period=='18-Q4'| Period=='19-Q1'),select=c(11:32))

ntestp<-subset(data2,UID==u1[i,] & (Period=='19-Q2'| Period=='19-Q3'),select=c(11:32))

x\_trainp<-as.matrix(ntrainp[2:22])

y\_trainp<-(ntrainp$Actuals)

y\_trainp<-jitter(y\_trainp,factor = 1,amount=NULL)

x\_testp<-as.matrix(ntestp[c(2:22)])

y\_testp<-(ntestp$Actuals)

if(df2$BEST=='LR'){

modelp<- lm(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp)

new\_testp<-data.frame(ntestp[c(2:22)])

predp<-predict(modelp,newdata=new\_testp)

df2$Pred\_19Q2<-predp[[1]]

df2$Pred\_19Q3<-predp[[2]]

} else if(df2$BEST=='Lasso'){

alpha1.fitp<-cv.glmnet(x\_trainp,y\_trainp,type.measure="mse",alpha=1,grouped = FALSE)

alpha1.predictedp<-predict(alpha1.fitp,s=alpha1.fitp$lambda.1se,newx=x\_testp)

df2$Pred\_19Q2<-alpha1.predictedp[[1]]

df2$Pred\_19Q3<-alpha1.predictedp[[2]]

} else if(df2$BEST=='Ridge'){

alpha0.fitp<-cv.glmnet(x\_trainp,y\_trainp,type.measure="mse",alpha=0,grouped = FALSE)

alpha0.predictedp<-predict(alpha0.fitp,s=alpha0.fitp$lambda.1se,newx=x\_testp)

df2$Pred\_19Q2<-alpha0.predictedp[[1]]

df2$Pred\_19Q3<-alpha0.predictedp[[2]]

} else if(df2$BEST=='DT'){

tree1<-ctree(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp,controls=ctree\_control(mincriterion = 0.9))

pt<-predict(tree1,ntestp[2:22])

df2$Pred\_19Q2<-pt[[1]]

df2$Pred\_19Q3<-pt[[2]]

} else if(df2$BEST=='RF'){

rf1<-randomForest(Actuals~Market\_Share\_Units+Market\_Share\_Revenue+Market\_Size\_Units+Market\_Size\_Revenue+MEI\_CPI\_Inflation+MEI\_Exports+MEI\_Fixed\_Investment+MEI\_GDP+MEI\_Government\_Growth+MEI\_Imports+MEI\_Industrial\_Production+MEI\_Merchandise\_Exports+MEI\_Merchandise\_Imports+MEI\_Nominal\_Retail\_Sales+MEI\_Private\_Consumption+MEI\_Real\_Retail\_Sales+MEI\_WPI\_Inflation+MEI\_Price\_Index+MEI\_Trade\_GDP\_Ratio+MEI\_Merchandise\_Trade\_GDP\_Ratio+MEI\_Real\_Nominal\_Sales,ntrainp)

pf<-predict(rf1,ntestp[2:22])

df2$Pred\_19Q2<-pf[[1]]

df2$Pred\_19Q3<-pf[[2]]

} else{

xi1<-ts(ntrainp$Actuals,frequency=4,start=c(2014,1))

pi1<-auto.arima(xi1)

qi1<-forecast(pi1,h=2)

q2<-data.frame(qi1)

df2$Pred\_19Q2<-q2$Point.Forecast[[1]]

df2$Pred\_19Q3<-q2$Point.Forecast[[2]]

}

df1<-rbind(df1,df2)

i=i+1

}

df1

write\_xlsx(df1,"C:\\Users\\saloni\\Downloads\\Final\_output\_with\_bestmodel.xlsx")