Name: Manasvi Surname: Ghanta Student Id: 19306751 Kaggle_NickName: Manasvi Ghanta The below part of R code will check 3 different models namely ETS, AUTO-ARIMA and PROPHET using functions from the R software. Each of the models takes in the values other than the last 18 from each of the 999 monthly time series and uses the last 18 observations for testing the forecasting performance which will be done by the forecast() function with h as the number of steps to forecast. Each model estimates its parameters based on the errors and likelihood value and returns the best output by minimizing the errors and maximising the likelihood. For the prophet model, the time series is converted to a dataframe for the input into the prophet model. FOr prophet model, there must be dates mentioned from when to start. The last 18 predicted observations are extracted from one more for loop for testing the forecasting performance. ```{r}

part 1

library(fpp2) library(Metrics) library(TSstudio) library(prophet) h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17))

ets model

train.data.ts<-ts(data=train.data,frequency = 12) ets.function.tscv <- function(x, h){forecast(ets(x, model = "ZZZ"), h=h)} ets.model <- tsCV(train.data.ts, ets.function.tscv, h=1) ets.model.forecast<-forecast(ets.model,h=h) mae.ets<-mae(test.data,ets.model.forecast$mean)

# arima model

arima.function.tscv <- function(x, h){forecast(auto.arima(x), h=h)} arima.model <- tsCV(train.data.ts, arima.function.tscv,h=1) arima.model.forecast<- forecast(arima.model,h=h) mae.arima<-mae(test.data,arima.model.forecast$mean)

prophet model

df.for.prophet <- ts_to_prophet(ts.obj = train.data.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<- make_future_dataframe(prophet.model,periods=18) predict.prophet.model<-predict(prophet.model, forecast.prophet.model) predict.prophet.required<- predict.prophet.model$yhat i=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(i in i:length(predict.prophet.required)){ predict.prophet.values[j]<-predict.prophet.required[i] j=j+1 } mae.prophet<-mae(test.data,predict.prophet.values) index = 1 df <- cbind(index, as.matrix(mae.ets),as.matrix(mae.arima),as.matrix(mae.prophet)) colnames(df)<-c("File Number", "ETS","Arima MAE","Prophet MAE")

index = 1 for (i in 2:100){ print(i) name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<- window(load.data,start=(length(load.data)-17)) train.data.ts<-ts(train.data,frequency = 12) #ets.model ets.function.tscv <- function(x, h){forecast(ets(x, model = "ZZZ"), h=h)} ets.model <- tsCV(train.data.ts, ets.function.tscv, h=1) ets.model.forecast<-forecast(ets.model,h=h) mae.ets<-mae(test.data,ets.model.forecast$mean) #arima model arima.function.tscv <- function(x, h){forecast(auto.arima(x), h=h)} arima.model <- tsCV(train.data.ts, arima.function.tscv, h=1) arima.model.forecast<- forecast(arima.model,h=h) mae.arima<-mae(test.data,arima.model.forecast$mean) #prophet model # df.for.prophet <- ts_to_prophet(ts.obj = train.data.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18) predict.prophet.model<- predict(prophet.model, forecast.prophet.model) predict.prophet.required<-predict.prophet.model$yhat k=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(k in k:length(predict.prophet.required)){ predict.prophet.values[j]<-predict.prophet.required[k] j=j+1 } mae.prophet<- mae(test.data,predict.prophet.values) df<-rbind(df,cbind(as.integer(nrow(df)+index),mae.ets, mae.arima,mae.prophet))

}

write.table(df, file ="my_part1.csv", col.names = c("File Number","MAE Ets", "MAE Arima","MAE Prophet"), sep=",", row.names=FALSE)

The overall best algorithm is considered to the AUTO-ARIMA model as it has lower MAE when computed than the other two models.

The second part of the R code includes all the data from the files as the training data and computes the next 18 forecasts for each
```{r}
#2nd part
h=18 # forecast horizon
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
data.ts <- ts(data = load.data, frequency = 12)
#ets model
ets.model <- ets(data.ts,model="ZZZ")
ets.forecast.model<-forecast(ets.model,h=h)
#arima model
ar.model<-auto.arima(y=data.ts)
fc.ar.model<-forecast(ar.model,h=h)
#prophet model
data.prophet <- ts(data = load.data, frequency = 12)
df.for.prophet <- ts_to_prophet(ts.obj = data.prophet, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet,weekly.seasonality = TRUE,daily.seasonality=TRUE)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
```

```
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
index = 1:h
df <- cbind(index, as.matrix(ets.forecast.model$mean),as.matrix(fc.ar.model$mean),as.matrix(predict.prophet.values))
colnames(df)<-c("index", "ETS Forecasts","ARIMA Forecasts","Prophet Forecasts")

#From the second file
for (i in 2:999){
  name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="")
  name<- paste(name,".csv",sep="")
  #print(name)
  load.data<- as.matrix(read.csv(name))#load data
  data.ts <- ts(data = load.data, frequency = 12)
  #ets model
  ets.model <- ets(data.ts,model="ZZZ")
  ets.forecast.model<-forecast(ets.model,h=h)
  #arima model
  ar.model<-auto.arima(y=data.ts)
  fc.ar.model<-forecast(ar.model,h=h)
  #prophet model
  data.prophet <- ts(data = load.data, frequency = 12)
  df.for.prophet <- ts_to_prophet(ts.obj = data.prophet, start = as.Date("2000-01-01"))
  prophet.model<-prophet(df.for.prophet,weekly.seasonality = TRUE,daily.seasonality=TRUE)
  forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
  predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
  predict.prophet.required<-predict.prophet.model$yhat
  i=length(predict.prophet.required)-18
  j=0
  predict.prophet.values<-c(1:18)
  for(i in i:length(predict.prophet.required)){
    predict.prophet.values[j]<-predict.prophet.required[i]
    j=j+1
  }

  predictions.ets <- as.matrix(ets.forecast.model$mean)
  predictions.arima<-as.matrix(fc.ar.model$mean)
  predictions.prophet<-as.matrix(predict.prophet.values)

  #save the forecast
  df<-rbind(df,cbind(as.integer(nrow(df)+index),predictions.ets,predictions.arima,predictions.prophet) )

}

write.table(df, file ="my_part2.csv",
          col.names = c("Id", "ETS Forecasts", "ARIMA Forecasts","Prophet Forecasts"),
          sep=",",
          row.names=FALSE)
```

The third part basically checks on the function's hyperparameters and the changes that have been done for improving the performance of the forecasting with the three different models. ```{r}

Part 3

library(Metrics) h=18 index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<- window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12)

autoplot(data.ts)

ets model

```
ets.model <- ets(data.ts, model="ZZZ",alpha=0.8, damped=TRUE, phi=0.9,lambda=1,biasadj = TRUE,ic="aicc",allow.multiplicative.trend = TRUE,bounds="admissible")
ets.forecast.model<-forecast(ets.model,h=h)
```

autoplot(ets.forecast.model)

```
mae.ets.adj<-mae(test.data,ets.forecast.model$mean)
```

arima model

data.ts %>% diff() %>% ggtsdisplay(main="")

```
arima.model.adj<-arima(data.ts, include.mean = FALSE, transform.pars = FALSE)
```

print(arima.model.adj)

```
arima.forecast.model<-forecast(arima.model.adj,h=h) mae.arima.adj<-mae(test.data,arima.forecast.model$mean)
```

prophet

```
df.for.prophet <- ts_to_prophet(ts.obj = data.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<-
make_future_dataframe(prophet.model,periods=18) predict.prophet.model<-predict(prophet.model, forecast.prophet.model) predict.prophet.required<-
predict.prophet.model$yhat i=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(i in i:length(predict.prophet.required)){
predict.prophet.values[j]<-predict.prophet.required[i] j=j+1 } mae.prophet.adj<-mae(test.data,predict.prophet.values) index = 1 df <- cbind(index,
as.matrix(mae.ets.adj),as.matrix(mae.arima.adj),as.matrix(mae.prophet.adj)) colnames(df)<-c("File Number", "ETS MAE","Arima MAE","Prophet MAE")
```

~~~
 Firstly, few time series have been plotted to check different components of the time series. It is noticed that most of the time ser
As there is no trend seen in most of the time series, a phi value of 0.9 is taken just in case if any of the time series forecast th

The main thing for the ARIMA model is to have the time series to be stationary. Most of the time series do not experience a changing
The mean is not included for the model even though there is no difference performed for the time series.

There are no improvements seen for prophet model in terms of MAE, so the default hyperparameters are set for a better forecast perfo
```{r}
for(i in 2:10){
  index<-1
  name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="")
  name <- paste(name,".csv",sep="")
  load.data<- as.matrix(read.csv(name))
  train.data<-window(load.data,start=1, end=(length(load.data)-18))
  test.data<-window(load.data,start=(length(load.data)-17))
  data.ts <- ts(data = train.data, frequency = 12)
  #autoplot(data.ts)
  #ets model
  ets.model <- ets(data.ts, model="ZZZ",alpha=0.8, damped=TRUE,  phi=0.9,lambda=1,biasadj = TRUE,ic="aicc",allow.multiplicative.tren
  ets.forecast.model<-forecast(ets.model,h=h)
  mae.ets.adj<-mae(test.data,ets.forecast.model$mean)

  #arima model
  #data.ts %>% diff() %>% ggtsdisplay(main="")
  arima.model.adj<-arima(data.ts, include.mean = FALSE, transform.pars = FALSE)
  #print(arima.model.adj)
  arima.forecast.model<-forecast(arima.model.adj,h=h)
  mae.arima.adj<-mae(test.data,arima.forecast.model$mean)
  #prophet model
  df.for.prophet <- ts_to_prophet(ts.obj = data.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
mae.prophet.adj<-mae(test.data,predict.prophet.values)
index = 1
  df<-rbind(df,cbind(as.integer(nrow(df)+index),mae.ets.adj, mae.arima.adj,mae.prophet.adj) )

colnames(df)<-c("File Number", "ETS MAE","Arima MAE","Prophet MAE")

}

write.table(df, file ="my_part3.csv",
         col.names = c("File Number","MAE Ets", "MAE Arima","MAE Prophet"),
         sep=",",
          row.names=FALSE)
~~~

```{r}

# rough code trying on prophet function

library("TSstudio") library("prophet") library("fpp2") name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="") name<- paste(name,".csv",sep="") y1<- as.matrix(read.csv(name)) data1 <- ts(data = y1, frequency = 12) df <- ts_to_prophet(ts.obj = data1, start = as.Date("2000-01-01")) m<- prophet(df,growth="linear", n.changepoints = 30,weekly.seasonality = TRUE, daily.seasonality = TRUE, seasonality.mode = 'additive') future<- make_future_dataframe(m,periods=18) forecast<-predict(m,future)
```

yhat from the output gives the predicted value for the prophet model.

mae.prophet<-mae(m$history$y,forecast$yhat)

~~~
Part 4 of the R code selects the best 3 and worst 3 from each of the models. Further, a residual analysis is performed along with th
```{r}
#part 4
#ets model
library(Metrics)
library(fpp2)
h=18
index<-1
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
ets.model <- ets(data.ts, model="ZZZ")
ets.forecast.model<-forecast(ets.model,h=h)#forecasts
mae.ets.adj<-mae(test.data,ets.forecast.model$mean)#computes mae
mse.ets.adj<-mse(test.data,ets.forecast.model$mean)#computes mse
rmse.ets.adj<-rmse(test.data,ets.forecast.model$mean)#computes rmse
index = 1
df <- cbind(index, as.matrix(mae.ets.adj),as.matrix(mse.ets.adj),as.matrix(rmse.ets.adj))
colnames(df)<-c("File Number", "MAE","MSE","RMSE")

for(i in 2:999){
  index<-1
  name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="")
  name <- paste(name,".csv",sep="")
  load.data<- as.matrix(read.csv(name))
  train.data<-window(load.data,start=1, end=(length(load.data)-18))
  test.data<-window(load.data,start=(length(load.data)-17))
  data.ts <- ts(data = train.data, frequency = 12)
  ets.model <- ets(data.ts, model="ZZZ")
  ets.forecast.model<-forecast(ets.model,h=h)
  mae.ets.adj<-mae(test.data,ets.forecast.model$mean)
  mse.ets.adj<-mse(test.data,ets.forecast.model$mean)
  rmse.ets.adj<-rmse(test.data,ets.forecast.model$mean)
index = 1
df<-rbind(df,cbind(as.integer(nrow(df)+index),mae.ets.adj,mse.ets.adj,rmse.ets.adj ))
colnames(df)<-c("File Number", "MAE","MSE","RMSE")
}
order.df<-df[order(df[,2]),]
~~~

For the ETS model, considering the residual analyis, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) are computed with the forecasting performance for all the 999 monthly time series files. As the above mentioned methods MSE, MAE and RMSE includes the residuals in the formula these methods have been used for selecting the first 3 best files and last three best files. The data is sorted in an ascending order as we need to look for the lowest errors possible for the model to be a better fit. Generally, forecast errors are those which are obtained by computing the difference between the forecast and the actual value. As an example for how it includes residuals, MAE formula is mentioned below

$$ {\displaystyle {\mbox{MAE}}={\frac {1}{n}}\sum *{t=1}^{n}\left|{\frac {A*{t}-F_{t}}{1}}\right|} $$ So, coming to what three files are considered best according to the ETS model will be the files train809 with 0.03382446 MAE, train818 with 0.06270706 MAE, train366 with 0.06638814 MAE as they have the least Errors of all. The files which are considered to be the worst for the ETS algorithm will be train882 with 3.075851 MAE, train876 with 3.306402 MAE, and train692 with 4.922794 as they have highest errors.

```{r}

# First best file using ETS

# 809 file
```

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",809,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) first.ets.ts<-ts(data=load.data,frequency = 12) autoplot(first.ets.ts)+ylab("Values") ggseasonplot(first.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 809 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) ets.model <- ets(data.ts, model="ZZZ") ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)

~~~
 For the above time series after plotting, it is seen that there is an increasing trend and no seasonality is observed from the time

Checking on the seasonal plot, there is a linear increase accross the months of the year with no overlaps
On checking the residuals using the forecasting for the ets model, the residuals seem like white noise with no particular trend but

```{r}
#Second best file using ETS
#818 file
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",818,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 818 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
ets.model <- ets(data.ts, model="ZZZ")
ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)
~~~

Coming to this time series, both the time series plot and the seasonal plots looks similar to that of the train 809 time series. There is a similar analysis observed with the residual plots as well.

```{r}

# Third best file using ETS

# 366 file

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",366,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) second.ets.ts<-ts(data=load.data,frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 366 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) ets.model <- ets(data.ts, model="ZZZ") ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)
```

~~~
 This time series observes seasonality between time 2 and 4 with a little bit of decreasing trend after time 4 till 6 and then flatte

The high peaks from the time series are noted in the seasonal plot as well which seem to be similar with the periods 2 and 3. There 

Checking the residual analysis, there is seasonality observed in the residuals in the first plot between 2 and 4 which is similar to

```{r}
#First poor file using ETS
#882 file
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",882,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 882 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
ets.model <- ets(data.ts, model="ZZZ")
ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)
~~~

This time series observes a high increase between time 2 and 4 and decreasing with ups and dwons from then. There is no particular trend or seasonality observed overall from this time series plot. The seasonal plots show some overlaps between periods with high increase in the period 3 and decrease in the period 5. The residuals are seen as normally distributed from the three plots. The first plot gives that the residuals are randomly distributed. The second ACF plot gives that there is no correlation observed with all the residual values lying inside the blue bands. And finally the third plot shows that the data is normally distributed with the symmetry.

```{r}

# Second poor file using ETS

# 876 file

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",876,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) second.ets.ts<-ts(data=load.data,frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 876 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) ets.model <- ets(data.ts, model="ZZZ") ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)
```

~~~
 This time series as well has an increasing and decreasing trend between the time values just like the previous one.
The seasonal plot as well has overlaps with satisfying the time periods from the time series plots
Finally, checking the residuals, it is seen that there is a random distribution (white noise) of residuals from the first plot. The

```{r}
#Third poor file using ETS
#692 file
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",692,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
#time series plot
autoplot(second.ets.ts)+ylab("Values")
#seasonal plot
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 692 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
ets.model <- ets(data.ts, model="ZZZ",alpha=0.8, damped=TRUE, phi=0.9,lambda=1,biasadj = TRUE,ic="aicc",allow.multiplicative.trend =
ets.forecast.model<-forecast(ets.model,h=h)

checkresiduals(ets.forecast.model)
~~~

This time series has seasonality and no particular linear trend. There are two peaks noticed at time 4 and 10. Considering the seasonal plot, it is seen that there is a huge increase in period 9 and there are overlaps observed in other periods. Also, in the 4th period, there is a huge peak observed. Checking the residual plots, the plots are randomly distributed and it is to the mean line 0 with few peaks coming into existence due to the peaks from the time series plot.The ACF plot has not much correlation observed and the residuals are perfect having the lines inside the blue bands. There is a correlation observed between 5 and 7. A normal plot has been observed for the residuals with having symmetry from the third plot.

```{r}

# Selection of 3 best and 3 poor performing files using arima model

# part 4

# arima model

library(Metrics) h=18 index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) arima.model.adj<-auto.arima(data.ts) arima.forecast.model<-forecast(arima.model.adj,h=h) mae.arima.adj<-mae(test.data,arima.forecast.model$mean) mse.arima.adj<-mse(test.data,arima.forecast.model$mean) rmse.arima.adj<-rmse(test.data,arima.forecast.model$mean) index = 1 df <- cbind(index, as.matrix(mae.arima.adj),as.matrix(mse.arima.adj),as.matrix(rmse.arima.adj)) colnames(df)<-c("File Number", "MAE","MSE","RMSE")

for(i in 2:999){ index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) arima.model.adj<-auto.arima(data.ts) arima.forecast.model<-forecast(arima.model.adj,h=h) mae.arima.adj<-mae(test.data,arima.forecast.model$mean) mse.arima.adj<-mse(test.data,arima.forecast.model$mean) rmse.arima.adj<-rmse(test.data,arima.forecast.model$mean) index = 1 df<-rbind(df,cbind(as.integer(nrow(df)+index),mae.arima.adj,mse.arima.adj,rmse.arima.adj )) colnames(df)<-c("File Number", "MAE","MSE","RMSE") } order.df<-df[order(df[,2]),]
```

~~~
 The three files that are considered best after checking the errors came to be the time series with the data in the files train809 wi

```{r}
##First best file using Arima
#809
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",809,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 809 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
arima.model.adj<-auto.arima(data.ts)
arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)
~~~

There is a linear trend observed for this time series with no seasonality seen. The seasonal plot also has no overlaps and there are straight lines Coming to the residuals, The residuals are normally distributed seeing the three plots. The first plot tells that the residuals are randomly distributed and the normal plot has symmetry. The points in the ACF plot are inside the blue bands. There is correlation observed between 11 and 31.

```{r}

# Second best file using Arima

# 810

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",810,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) second.ets.ts<-ts(data=load.data,frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 810 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) arima.model.adj<-auto.arima(data.ts) arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)

```
 This time series is almost similar to the previous one other than the value from ACF plot goin outside the blue bands.There is not m

```{r}
#Third best file using Arima
#584
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",584,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 584 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
arima.model.adj<-auto.arima(data.ts)
arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)
```

The time series has observed a decreasing trend from time 5 and flattened later on. There is seasonality observed in this time series. The high peaks and ups and downs from the time series plots are the reasons for the seasonal plot to have the overlaps. The residuals do not observe any particular trend and are randomly distributed. But the ACF plot shows that there is a high correlation observed with going over the blue bands which is the confidence interval. Finally the normal plot doesnt have symmetry leading to that the residuals are not normally distributed. ```{r}

First Poor file using Arima

169

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",169,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) second.ets.ts<-ts(data=load.data,frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 169 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) arima.model.adj<-auto.arima(data.ts) arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)

~~~
 The time series plot observed a seasonality trend and no particular overall linear trend.
The seasonality plot shows overlaps between periods and high peak from the time series plot can be understood in the period 2.
The residuals do not observe any particualr trend but there might be seasonality observed and not being a white noise. The ACF plot

```{r}
#Second Poor file using Arima
#882
h=18
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",882,sep="")
name<- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
second.ets.ts<-ts(data=load.data,frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 882 Time series")
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
arima.model.adj<-auto.arima(data.ts)
arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)
~~~

The time series has an increasing and a decreasing behaviour with no seasonality in the time series. The overlap has been observed in the seasonal plot. Checking the residual plots, the residuals are kind of normally distributed. The residuals are randomly distributed. From the ACF plot, all the points lie within the blue bands. ```{r}

Third Poor file using Arima

890

h=18 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",890,sep="") name<- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) second.ets.ts<-ts(data=load.data,frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 890 Time series") train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) data.ts <- ts(data = train.data, frequency = 12) arima.model.adj<-auto.arima(data.ts) arima.forecast.model<-forecast(arima.model.adj,h=h)

checkresiduals(arima.forecast.model)

~~~
 This time series has an increasing trend overall with seasonality between time 3 and 6. There is a decreasing trend observed from ti
The seasonal plot has overlaps, decreasing, and increasing lines in accordance with the time series plot
The residuals are randonly distributed with no particular trend seen in the first plot. The ACF plot shows there is no high correlat




```{r}
#Selects 3 best and three poor performing files using prophet
#part 4
#prophet model
library(Metrics)
library(prophet)
~~~

```
library(TSstudio)
h=18
index<-1
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",1,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
data.ts <- ts(data = train.data, frequency = 12)
df.for.prophet <- ts_to_prophet(ts.obj = data.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
mae.prophet.adj<-mae(test.data,predict.prophet.values)
mse.prophet.adj<-mse(test.data,predict.prophet.values)
rmse.prophet.adj<-rmse(test.data,predict.prophet.values)
index = 1
df <- cbind(index, as.matrix(mae.prophet.adj),as.matrix(mse.prophet.adj),as.matrix(rmse.prophet.adj))
colnames(df)<-c("File Number", "MAE","MSE","RMSE")

for(i in 2:999){
  index<-1
  name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",i,sep="")
  name <- paste(name,".csv",sep="")
  load.data<- as.matrix(read.csv(name))
  train.data<-window(load.data,start=1, end=(length(load.data)-18))
  test.data<-window(load.data,start=(length(load.data)-17))
  data.ts <- ts(data = train.data, frequency = 12)
  df.for.prophet <- ts_to_prophet(ts.obj = data.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
mae.prophet.adj<-mae(test.data,predict.prophet.values)
mse.prophet.adj<-mse(test.data,predict.prophet.values)
rmse.prophet.adj<-rmse(test.data,predict.prophet.values)
index = 1
df<-rbind(df,cbind(as.integer(nrow(df)+index),mae.prophet.adj,mse.prophet.adj,rmse.prophet.adj ))
colnames(df)<-c("File Number", "MAE","MSE","RMSE")
}
order.df<-df[order(df[,2]),]
```

The best three run files using the prophet algorithm are train584 with 0.06119216 MAE, train979 with 0.12532804 MAE, train919 with 14082209 MAE. On the other hand, the poorly run files are train35 with 53.29192 as MAE, train61 with 55.41867 MAE, and train37 with 69.62170 MAE

```{r}

# First Best file using Prophet
```

584

library(Metrics) h=18 index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",584,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<- window(load.data,start=(length(load.data)-17)) second.ets.ts <- ts(data = train.data, frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 584 Time series") df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18) predict.prophet.model<-predict(prophet.model, forecast.prophet.model) predict.prophet.required<-predict.prophet.model$yhat i=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(i in i:length(predict.prophet.required)){ predict.prophet.values[j]<-predict.prophet.required[i] j=j+1 } checkresiduals(predict.prophet.required)

~~~
 The time series and seasonal plot are already seen before.
Coming to the residuals using the prophet model, there is seasonality and decreasing trend observed which can be understood why from

```{r}
#Second Best file using Prophet
#979
library(Metrics)
h=18
index<-1
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",979,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
second.ets.ts <- ts(data = train.data, frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 979 Time series")
df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
checkresiduals(predict.prophet.required)
~~~

There is an increasing trend seen in the time series plot with no seasonality trend. The lines do overlap in the seasonality plot but there is not much thing to note. The residuals are not normally distributed as there is a linear increasing trend with seasonality observed in the first plot. the ACF plot is decreasing and there is a high correlation observed with a pattern. Thus, leads to not having a normal plot. ```{r}

Third Best file using Prophet

919

library(Metrics) h=18 index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",919,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<- window(load.data,start=(length(load.data)-17)) second.ets.ts <- ts(data = train.data, frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 919 Time series") df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18) predict.prophet.model<-predict(prophet.model, forecast.prophet.model) predict.prophet.required<-predict.prophet.model$yhat i=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(i in i:length(predict.prophet.required)){ predict.prophet.values[j]<-predict.prophet.required[i] j=j+1 } checkresiduals(predict.prophet.required)

~~~
 The interpepation of this set of time series in terms of residual plots, time series or the seasonal plots are similar to that of th

```{r}
#First Poor file using Prophet
#35
library(Metrics)
h=18
index<-1
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",35,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
second.ets.ts <- ts(data = train.data, frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 35 Time series")
df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
checkresiduals(predict.prophet.required)
~~~

The time series plot shows a decreasing trend with seasonality observed. The high peaks in the seasonality plot are noticed and this might be due to the high peaks from the time series plot with overlaps.
Coming to the residual plots, again there is a decreasing trend and flattening out observed in the residual plot and ACF plot leading to no normal plot. The ACF plot shows that there is high correlation with points going beyond the blue bands

```{r}

# Second Poor file using Prophet

# 61

library(Metrics) h=18 index<-1 name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",61,sep="") name <- paste(name,".csv",sep="") load.data<- as.matrix(read.csv(name)) train.data<-window(load.data,start=1, end=(length(load.data)-18)) test.data<-window(load.data,start=(length(load.data)-17)) second.ets.ts <- ts(data = train.data, frequency = 12) autoplot(second.ets.ts)+ylab("Values") ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) + ggtitle("Seasonal plot: 61 Time series") df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01")) prophet.model<-prophet(df.for.prophet) forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18) predict.prophet.model<-predict(prophet.model, forecast.prophet.model) predict.prophet.required<-predict.prophet.model$yhat i=length(predict.prophet.required)-18 j=0 predict.prophet.values<-c(1:18) for(i in i:length(predict.prophet.required)){ predict.prophet.values[j]<-predict.prophet.required[i] j=j+1 } checkresiduals(predict.prophet.required)
```

~~~~
 The time series plot has no particular trend seen but there is seasonality seen.
The seasonal plot and the residual analysis is similar to the previous one.

```{r}
#Third Poor file using Prophet
#37
library(Metrics)
h=18
index<-1
name <- paste("/Users/manasvighanta/Desktop/Forecasting/Assignment_Kaggle/train/train",37,sep="")
name <- paste(name,".csv",sep="")
load.data<- as.matrix(read.csv(name))
train.data<-window(load.data,start=1, end=(length(load.data)-18))
test.data<-window(load.data,start=(length(load.data)-17))
second.ets.ts <- ts(data = train.data, frequency = 12)
autoplot(second.ets.ts)+ylab("Values")
ggseasonplot(second.ets.ts, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: 37 Time series")
df.for.prophet <- ts_to_prophet(ts.obj = second.ets.ts, start = as.Date("2000-01-01"))
prophet.model<-prophet(df.for.prophet)
forecast.prophet.model<-make_future_dataframe(prophet.model,periods=18)
predict.prophet.model<-predict(prophet.model, forecast.prophet.model)
predict.prophet.required<-predict.prophet.model$yhat
i=length(predict.prophet.required)-18
j=0
predict.prophet.values<-c(1:18)
for(i in i:length(predict.prophet.required)){
  predict.prophet.values[j]<-predict.prophet.required[i]
  j=j+1
}
checkresiduals(predict.prophet.required)
~~~~

The time series observed seasonality in the data and no particular trend leading to the seasonality plot with high peaks in it. Again, the analysis of the residual plots will be the similar one to the above mentioned for the prophet model.

To start with, the prophet model for any of the time series is considered as a bad method as the gaussian assumptions of the residuals are not being met. Also, the MAE is seen so high for any time series especially for the worst performed ones using the prophet model. The time series train584 has been selected by ARIMA Model as the third best time series with MAE of 0.04 approx while using prophet model, the MAE of the train584 time series is 0.06 approx and is selected as the first best model without gaussian assumptions being met and with high correlation of the residuals for the prophet model. Secondly, it is seen that the best selected models by ETS method have one value with high correlation and the residuals are not normally distributed while the worst selected once have low correlation and are normally distributed. The residuals are normally distributed with the best selected models using the ARIMA model unlike the ETS model. The first two best selected models have a linear trend observed similar to the ETS model. There is seasonality in the time series for the worst selected time series by this model. Also, the lines from the ACF plots for the worst selected time series lie within the the blue bands while there is one highly correlated one for the best selected models similar to the ETS model. It is interesting to note that both the models ETS and ARIMA have given the file train809 as the best time series with MAE of 0.03 approx for both the models.