#Assignment23\_Session23

#Problem

#1. Perform the below given activities:

#a. Take Apple Stock Prices from Yahoo Finance for last 90 days

#b. Predict the Stock closing prices for next 15 days.

#c. Submit your accuracy

#d. After 15 days again collect the data and compare with your forecast

#Answers

#\*\*\*\*NOTE\*\*\*\*

#APPL1 is my dataset file

df<- AAPL1

df$Date <- as.Date(df$Date)

data = ts(df$Close)

test = data[60:76]

data = data[1:59]

plot(data, main= "Daily Close Price")

class(data)

#This tells you that the data series is in a time series format

start(data)

#This is the start of the time series

end(data)

#This is the end of the time series

frequency(data)

#The

summary(data)

plot(data)

#This will plot the time series

abline(reg=lm(data~time(data)))

# This will fit in a line

boxplot(data~cycle(data))

#Box plot across months will give us a sense on seasonal effect

data = ts(df$Close, frequency = 10)

plot(data, main = "Daily Close Price")

decompose(data)

decompose(data, type = "multi")

par(mfrow=c(1,2))

plot(decompose(data, type = "multi"))

# creating seasonal forecast

library(forecast)

par(mfrow=c(1,1))

seasonplot(data)

# lags

lag(data,10)

lag.plot(data)

# Partial auto correlation

pac <- pacf(data)

pac$acf

#The blue line above shows significantly different values than zero. Clearly, the graph above has a cut off on

#PACF curve after 1st lag which means this is mostly an AR(1) process.

# Auto correlation

ac <- acf(data)

ac$acf

#the decay of ACF chart is very slow, which means that the population is not stationary

# we now intend to regress on the difference of logs rather than log directly.

#Let's see how ACF and PACF curve come out after regressing on the difference

# looking at ACF and PACF graph it is clear that the time series is not stationary

#------------------------------------------

pacf(diff(log(data)))

acf(diff(log(data)))

#now its correct

#----------------------------------------------

# deseasonlise the time series

tbl <- stl(data, 'periodic')

stab <- seasadj(tbl)

seasonplot(stab, 12)

# unit root for stationarity

# The Augmented Dicky Fuller Test for

library(tseries)

adf.test(data)

# P value is greater than 0.05 now, hence we fail to reject the null hypo

# there is unit root in time series hence the time series is not stationary

acf(log(data))

pacf(log(data))

acf(diff(log(data)))

pacf(diff(log(data)))

#main part start

data = ts(na.omit(AAPL1$Close ), frequency=10)

decomp = stl(data, s.window="periodic") #decompose

deseasonal\_cnt <- seasadj(decomp)

plot(decomp)

adf.test(data, alternative = "stationary")

#since it's p value is 0.14 which is greater than 0.05

#we have to do further processing by changing the value out of(p,d,q) of d.

Acf(data, main='')

Pacf(data, main='')

#thus still acf is not good

#thus we change d again n again so that we get desired p value

data = diff(deseasonal\_cnt, differences = 1)

plot(data)

adf.test(data, alternative = "stationary")

data = diff(deseasonal\_cnt, differences = 2)

plot(data)

adf.test(data, alternative = "stationary")

#now since p value is 0.01,concludes it is stationary

Acf(data, main='ACF for Differenced Series')

Pacf(data, main='PACF for Differenced Series')

#they seems correct,now start modelling

# Automatic ARIMA Model

model2 <- auto.arima(deseasonal\_cnt,seasonal = FALSE)

model2

tsdisplay(residuals(model2), lag.max=15, main='Seasonal Model Residuals')

#tsdisply helps in display overall of various things

plot(forecast(model2, h=15))

accuracy(model2)

#MAPE 1.303

#----------------------------------------------

# more running model on deseasonal\_cnt(deseasonal data)

model3 <- arima(deseasonal\_cnt, order=c(1,2,7))

model3

tsdisplay(residuals(model3), lag.max=15, main='Seasonal Model Residuals')

plot(forecast(model3, h=15))

accuracy(model3)

#MAPE 1.180

#-------------------------------------------------

# taking random order

model4 <- arima(deseasonal\_cnt, order = c(4,2,7))

model4

tsdisplay(residuals(model4), lag.max=15, main='Seasonal Model Residuals')

accuracy(model4)

plot(forecast(model4, h=15))

#MAPE 1.098

#---------------------------------------------------

# taking random order

model5 <- arima(deseasonal\_cnt, order = c(4,2,4))

model5

tsdisplay(residuals(model5), lag.max=15, main='Seasonal Model Residuals')

accuracy(model5)

plot(forecast(model5, h=15))

#MAPE 1.117

#---------------------------------------------------

# taking random order

model6 <- arima(deseasonal\_cnt, order = c(3,2,5))

model6

tsdisplay(residuals(model6), lag.max=15, main='Seasonal Model Residuals')

accuracy(model6)

plot(forecast(model6, h=15))

#MAPE 1.172

#---------------------------------------------------

# taking random order

model7 <- arima(deseasonal\_cnt, order = c(0,2,1))

model7

tsdisplay(residuals(model7), lag.max=15, main='Seasonal Model Residuals')

accuracy(model7)

plot(forecast(model7, h=15))

#MAPE 1.274

#---------------------------------------------------

# taking random order

model8 <- arima(deseasonal\_cnt, order = c(1,2,0))

model8

tsdisplay(residuals(model8), lag.max=15, main='Seasonal Model Residuals')

accuracy(model8)

plot(forecast(model8, h=15))

#MAPE 1.596

#---------------------------------------------------

# Holt Winters Exponential Smoothing Model

model9 <- HoltWinters(deseasonal\_cnt, gamma = F)

summary(model9)

tsdisplay(residuals(model9), lag.max=15, main='Seasonal Model Residuals')

plot(forecast(model9, h=15))

accuracy(forecast(model9, h=15))

#MAPE 1.344

#-----------------------------------------------------

# ETS

model10 <- ets(deseasonal\_cnt)

summary(model10)

tsdisplay(residuals(model10), lag.max=15, main='Seasonal Model Residuals')

plot(forecast(model10, h=15))

accuracy(forecast(model10, h=15))

#MAPE 1.302

#---------------------------------------------------------------

# model4 ( ARIMA) is most accurate with MAPE 1.098

#---------------------------------------------------------------

# Making predictions for next 15 days

predicted <- forecast(model4, 15)

predicted

# comparing data with forecast

predicted$residuals[60:76]