# Assignment 23

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

Sol :-

df = read.csv('AAPL.csv')

head(df)

str(df)

new\_date <- as.Date(df$Date)

new\_date

str(df)

format(new\_date,format="%B %d %Y")

# %d - day as number 1-31

# %a - weekday such as Mon

# %A- complete day name ex.Monday

# %m - month as a number

# %b - short form of month Jan, Feb

# %B - full form of month, January

# %y - two digit year

# %Y- four digit year

data = ts(df$Close,frequency =12)

plot(data,main="Monthly Closing Prices")

# Additive Time Series

# Trend + Seasonality+ Cyclicity+ error

# Multiplicative Time Series

## Trend \* Seasonality \* Cyclicity \* error

# additive model is easy to explain, easy to forecast and interpret

# multiplicate models can be converted to additive models using log of the time series

log(data)

# assumption for time series forecst:

#1- the time series should be stationary

# Identify the stationarity of a time series

#1- mean value of the time series is constant over time, the trend should not be present in the series

#2- the variance does not increase over time

#3- the seasonality impact is minimal, deseasonalization of the time series data

decompose(data) # default method is additive

decompose(data, type='multi')

par(mfrow=c(1,2))

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

library(forecast)

seasonplot(data)

lag(data,10)

lag.plot(data)

# Calculation of Autocorrelation and Partial Autocorrelation

data

ac<-acf(data)

ac$acf

# data time series may not have stationarity

pac<-pacf(data)

pac$acf

# looking at the ACF and PACF graph we can conclude that the time series is not stationary

model <- lm(data~c(1:length(data)))

summary(model)

plot(resid(model),type='l')

# the series is not stationary

# deseasonalize the time series

tbl <- stl(data,'periodic')

stab<-seasadj(tbl)

seasonplot(stab,12)

# statistically we need to test out if the series is stationary or not

# Augmented Dickey Fuller Test

library(tseries)

adf.test(data)

# if the p-value is less than 0.05, then the time series is stationary, else not

# Time Series Forecasting Models

# Simple Exponential Smoothing

# Double Expo. Smoothing

# Tripple Expo. Smoothing

# AR-I-MA model

#PACF- p

#diff - d

#ACF- q

model2<-auto.arima(data)

accuracy(model2)

plot(forecast(model2,h=12))

adf.test(diff(data))

plot(diff(data))

diff(data,differences = 3)

#running a model on diff data

model3<-auto.arima(diff(data))

accuracy(model3)

acf(diff(data))

pacf(diff(data))

#taking random order

model4 <- Arima(diff(data),order=c(4,0,5))

model4

accuracy(model4)

model5 <- Arima(diff(data),order=c(4,0,4))

model5

accuracy(model5)

model6<-Arima(data,order=c(3,0,5))

model6

accuracy(model6)

model7<-Arima(diff(data),order=c(4,0,4))

model7

accuracy(model7)

model8<-Arima(diff(data),order=c(0,0,1))

model8

accuracy(model8)

model9<-Arima(diff(data),order=c(1,0,0))

model9

accuracy(model9)

model10<-Arima(diff(data),order=c(1,0,1))

model10

accuracy(model10)

model11<-Arima(diff(data),order=c(1,0,2))

model11

accuracy(model11)

model12<-Arima(diff(data),order=c(1,1,3))

model12

accuracy(model12)

# MAPE = mean absolute percentage error (should be < 10%) for a good model

par(mfrow=c(1,2))

plot(forecast(model5,h=12))

plot(log(data))

# Holt Winters Exponential Smoothing Model

# if series is stationary then use simple exponential smoothing model

model4<-HoltWinters(data,beta = F, gamma = F)

summary(model4)

model4

library(forecast)

plot(forecast(model4,12))

# Holt Winters Exponential Smoothing Model

# if series is not stationary and only trend component is present, then use double exponential smoothing model

model5<-HoltWinters(data,gamma = F)

summary(model5)

model5

plot(forecast(model5,12))

plot(log(data))

# Holt Winters Exponential Smoothing Model

# if series is not stationary and trend, seasonality component is present, then use tripple exponential smoothing model

model6<-HoltWinters(data)

summary(model6)

model6

plot(log(data))

plot(forecast(model6,12))

# MAPE

# Automatic Exponential Smoothing Model

model7<-ets(data)

summary(model7)

accuracy(model7)

plot(log(data))

plot(forecast(model7,12))