# packages to install

#install.packages('forecast')

#install.packages('tseries')

#install.packages('lubridate') #use for date manipulation

library(forecast)

library(tseries)

library(lubridate)

library(zoo)

library(ISLR)

library(caret)

library(randomForest)

library(dplyr)

library(e1071)

library(car)

library(pROC)

library(binaryLogic)

library(rpart)

library(rpart.plot)

library(Hmisc)

library(ggplot2)

setwd("~/R/MBAD 6211/MBAD6211\_DataSets")

CO2Emiss <- read.csv("cleaned\_CO2\_Concentrations\_Monthly\_3.18.22.csv",header=TRUE,row.names=1)

#Initial Data Exploration

summary(CO2Emiss)

str(CO2Emiss)

nrow(CO2Emiss) #determine total number of observations

head(CO2Emiss)

range(CO2Emiss$parts.per.million, na.rm = TRUE) #determine range of ppm

#Convert full.date from CHR to DATE format so can graph

CO2Emiss$full.date <- mdy(CO2Emiss$full.date)

str(CO2Emiss)

#Remove unnecessary columns

CO2Emiss <- CO2Emiss %>% select(-c(country, indicator, date, month))

str(CO2Emiss)

#Initial Visualization + turn data into tseries

CO2.ts <- ts(CO2Emiss$parts.per.million, start=c(1958,3),frequency = 12)

plot(CO2.ts, xlab="Date",ylab="CO2 Emissions (Parts per Million)",type="l")

autoplot(decompose(CO2.ts))

# prepare test & training data:

nTest <- 228 #30% of the total observations (762)

nTrain <- length(CO2.ts)- nTest

train.CO2.ts <- window(CO2.ts,start=c(1958,3),end=c(1958,nTrain)) #historical data

test.CO2.ts <- window(CO2.ts,start=c(1958,nTrain+1),end=c(1958,nTrain+nTest)) #recent data

## Model 1: Linear Trend Model

CO2.lm <- tslm(CO2.ts~trend)

summary(CO2.lm)

# Model 1: Linear Training Model

train.CO2.lm <- tslm(train.CO2.ts~trend)

summary(train.CO2.lm)

train.CO2.lm.pred <- forecast(train.CO2.lm,h=nTest,level=0.95)

# Visualize the Linear Trend Model

par(mfrow = c(1, 1)) #c(row, column)

plot(train.CO2.lm.pred, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030),main = "", flty = 2)

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(train.CO2.lm.pred$fitted, lwd = 2, col = "blue")

lines(test.CO2.ts)

# plot the residuals

par(mfrow = c(1, 1))

plot(train.CO2.lm.pred$residuals, ylim = c(-25, 25), ylab = "Forecast Errors", xlab = "Time", bty = "l", xaxt = "n", xlim = c(1955,2030),main = "")

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(test.CO2.ts -train.CO2.lm.pred$mean, lwd = 2, col = "blue")

# Evaluate model performance

accuracy(train.CO2.lm.pred,test.CO2.ts)

## Model 2: Polynomial Trend

train.CO2.poly <- tslm(train.CO2.ts ~ trend + I(trend^2))

summary(train.CO2.poly)

train.CO2.poly.pred <- forecast(train.CO2.poly, h = nTest, level = 0.95)

accuracy(train.CO2.poly.pred,test.CO2.ts)

# Visualize the polynomial trend model

par(mfrow = c(1, 1)) #c(row, column)

plot(train.CO2.poly.pred, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030),main = "", flty = 2)

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(train.CO2.poly.pred$fitted, lwd = 2, col = "blue")

lines(test.CO2.ts)

## Model 3: with seasonality

train.CO2.season <- tslm(train.CO2.ts ~ season)

summary(train.CO2.season)

train.CO2.season.pred <- forecast(train.CO2.season, h = nTest, level = 0)

accuracy(train.CO2.season.pred,test.CO2.ts)

# Visualize the Seasonality model

par(mfrow = c(1, 1)) #c(row, column)

plot(train.CO2.season.pred, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030),main = "", flty = 2)

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(train.CO2.season.pred$fitted, lwd = 2, col = "blue")

lines(test.CO2.ts)

## Model 4: with trend and seasonality

train.CO2.trend.season <- tslm(train.CO2.ts ~ trend + I(trend^2) + season)

summary(train.CO2.trend.season)

train.CO2.trend.season.pred <- forecast(train.CO2.trend.season, h = nTest, level = 0)

accuracy(train.CO2.trend.season.pred,test.CO2.ts)

# Visualize the trend and seasonality model

par(mfrow = c(1, 1)) #c(row, column)

plot(train.CO2.trend.season.pred, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030),main = "", flty = 2)

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(train.CO2.trend.season.pred$fitted, lwd = 2, col = "blue")

lines(test.CO2.ts)

## Model 5: Simple Moving Average

sma <- rollmean(train.CO2.ts, k = 12, align = "right")

# obtain the last moving average in the training period

last.sma <- tail(sma, 1)

# create forecast based on last MA

sma.pred <- ts(rep(last.sma, nTest), start = c(1958, nTrain + 1),

end = c(1958, nTrain + nTest), freq = 12)

# plot the series

plot(train.CO2.ts, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030),main = "")

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(sma, lwd = 2, col = "blue")

lines(sma.pred, lwd = 2, col = "blue", lty = 2)

lines(test.CO2.ts)

accuracy(sma.pred, test.CO2.ts)

## Model 6: Simple Exponential Smoothing

CO2.ses <- ses(train.CO2.ts, alpha = 0.2, h=228)

autoplot(CO2.ses) +

autolayer(fitted(CO2.ses), series="Fitted") +

autolayer(test.CO2.ts)+

ylab("CO2 Emissions (Parts per Million)") + xlab("Date")

accuracy(CO2.ses,test.CO2.ts)

# # Identify optimal alpha parameter

# ses.alpha <- seq(.01, .99, by = .01)

# ses.MAPE <- NA

# for(i in seq\_along(ses.alpha)) {

# fit <- ses(train.CO2.ts, alpha = ses.alpha[i], h = 36)

# ses.MAPE[i] <- accuracy(fit, test.CO2.ts)[2,5]

# }

#

# # convert to a data frame and identify min alpha value

# ses.alpha.fit <- data\_frame(ses.alpha, ses.MAPE)

# ses.alpha.min <- filter(ses.alpha.fit, ses.MAPE == min(ses.MAPE))

#

# # plot MAPE vs. alpha

# ggplot(ses.alpha.fit, aes(ses.alpha, ses.MAPE)) +

# geom\_line() +

# geom\_point(data = ses.alpha.min, aes(ses.alpha, ses.MAPE), size = 2, color = "blue")

## Model 7:Holt-Winters exponential smoothing

CO2.hwin <- ets(train.CO2.ts, model = "AAA")

# create predictions

CO2.hwin.pred <- forecast(CO2.hwin, h = nTest, level = 0.95)

accuracy(CO2.hwin.pred, test.CO2.ts)

# plot the series

plot(CO2.hwin.pred, ylim = c(300, 430), ylab = "CO2 Emissions (Parts per Million)", xlab = "Date",

bty = "l", xaxt = "n", xlim = c(1955,2030), main = "", flty = 2)

axis(1, at = seq(1955, 2030, 1), labels = format(seq(1955, 2030, 1)))

lines(CO2.hwin.pred$fitted, lwd = 2, col = "blue")

lines(test.CO2.ts)

## Model 8: ARIMA

# Homoscedasticity?

# No, variances are the same over time

plot(CO2.ts)

# Stationary?

# No, increasing mean over time

# Fix by taking the difference

CO2.ts2 <- diff(CO2.ts)

plot(CO2.ts2)

# Random Walk?

# Use Phillips-Perron Unit Root Test to check

# If p-value is significant, reject the null hypothesis (i.e., data is not random walk)

PP.test(CO2.ts2)

# p-value = 0.01, therefore data doesn't have random walk

# ACF test for White Noise

# ACF shows correlation between y\_t and lagged terms y\_(t-h)

# The figure suggests seasonal lagged autocorrelation

acf(CO2.ts2,main="CO2 Emissions (Parts per Million)")

CO2.ARIMAfit <- auto.arima(CO2.ts, approximation=FALSE,trace=TRUE)

summary(CO2.ARIMAfit)

# Use the best ARIMA model to forecast future emissions

CO2.ARIMA.pred <- predict(CO2.ARIMAfit,n.ahead=228)

CO2.ARIMA.pred

# Plot the data

par(mfrow = c(1,1))

plot(CO2.ts,type='l',xlim=c(1955,2040),ylim=c(300,460),xlab = 'Date',ylab = 'CO2 Emissions (Parts per Million)')

lines(CO2.ARIMA.pred$pred,col='blue')

lines(CO2.ARIMA.pred$pred+2\*CO2.ARIMA.pred$se,col='orange')

lines(CO2.ARIMA.pred$pred-2\*CO2.ARIMA.pred$se,col='orange')

checkresiduals(CO2.ARIMAfit)