# ETS Seasonal

library(forecast)

dataset1 <- maml.mapInputPort(1) # class: data.frame

dataset2 <- maml.mapInputPort(2) # class: data.frame

seasonality<-12

labels <- as.numeric(dataset1$data)

timeseries <- ts(labels,frequency=seasonality)

model <- ets(timeseries)

numPeriodsToForecast <- ceiling(max(dataset2$time)) - ceiling(max(dataset1$time))

numPeriodsToForecast <- max(numPeriodsToForecast, 0)

forecastedData <- forecast(model, h=numPeriodsToForecast)

forecastedData <- as.numeric(forecastedData$mean)

#output <- data.frame(cbind(dataset2,forecastedData))

output <- data.frame(time=dataset2$time,forecast=forecastedData)

data.set <- output

maml.mapOutputPort("data.set");

# Metrics

library(forecast)

masefun <- function(observed, predicted){

error = 0;

if (length(observed) != length(predicted)) {

return (NA);

} else if (length(observed) == 0 || length(predicted) == 0) {

return (NA);

}

else {

denom = (sum(abs(observed[2:length(observed)] - observed[1:(length(observed) - 1)])))/(length(observed) - 1)

error = sum((abs(observed-predicted)) / denom)/length(observed);

}

return (error);

}

smape <- function(observed, predicted){

error = 0;

if (length(observed) != length(predicted)) {

return (NA);

} else if (length(observed) == 0 || length(predicted) == 0) {

return (NA);

}

else {

error = sum((abs(observed-predicted)) / (observed+predicted))/length(observed);

# denom = (sum(abs(observed[2:length(observed)] - observed[1:(length(observed) - 1)])))/(length(observed) - 1)

# error = sum((abs(observed-predicted)) / denom)/length(observed);

}

return (100.0\*error);

}

dataset1 <- maml.mapInputPort(1) # class: data.frame

time <- as.numeric(dataset1$time)

observed\_data <- as.numeric(dataset1$data)

forecast <- as.numeric(dataset1$forecast)

plot(time,observed\_data,type="l",col="blue",xlab="Time",ylab="Data",lwd=1.5)

lines(time,forecast,col="red",lwd=1.5)

legend("topleft",legend = c("Original Data","Average Seasonal ETS and ARIMA Forecast"),bty=c("n","n"),lty=c(1,1),pch=16,col=c("blue","red"))

forecast\_data\_testwindow <- as.numeric(forecast[(which(!is.na(forecast)))])

actual\_data\_testwindow <- as.numeric(observed\_data[(which(!is.na(forecast)))])

mase <- masefun(actual\_data\_testwindow,forecast\_data\_testwindow)

smape <- smape(actual\_data\_testwindow,forecast\_data\_testwindow)

arima\_acc <- data.frame(Method=as.character("average seasonal arima & ets"),accuracy(forecast\_data\_testwindow,actual\_data\_testwindow),MASE=mase,sMAPE=smape)

arima\_acc$Method <- as.character(arima\_acc$Method)

data.set <- arima\_acc

lapply(data.set,class)

maml.mapOutputPort("data.set");