Predict_413_Sec55_Homework_3

Singh, Gurjeet

March 18, 2018

Contents

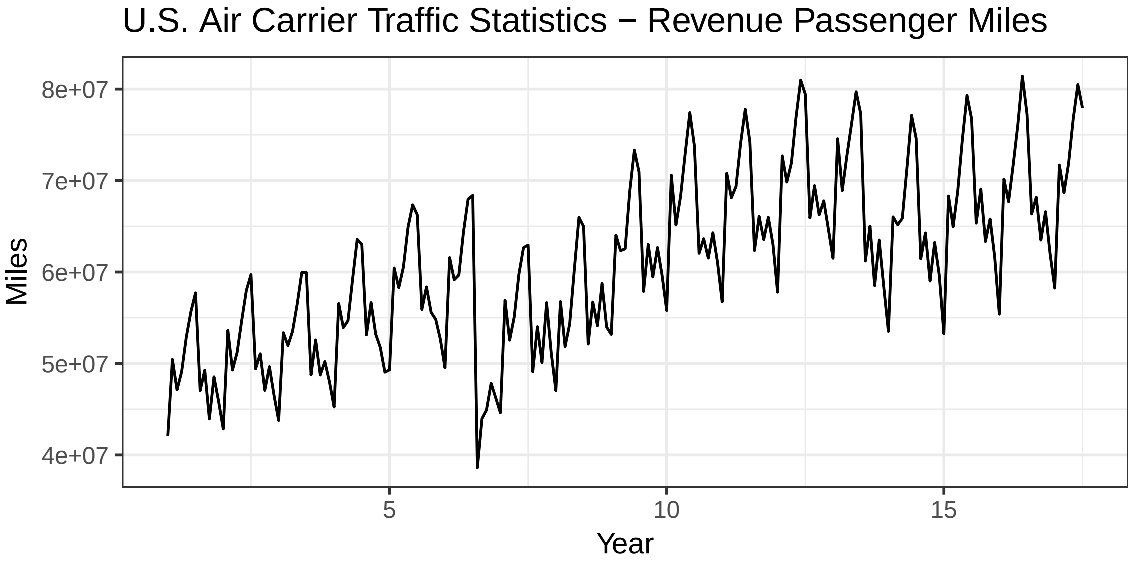
1. Introduction

For homework #3, we picked a time series and build three models learned in this course. We have selected to build Auto ARIMA, ETS, and neaural network models We will test the performance of each model and select the best model.

2. Data

We have selected the U.S Air Carrier Traffic Statistics – Revenue Passenger Miles for this project.

The figure below shows that this data has strong seasonality within each year. Along with the seasonality, there is also an upward trend to it. We notice that every summer there's high traffic i.e. peak times. During the months of January and February, we see a dip in the traffic. This makes sense because a lot of people are back to their normal routine after the holidays. Therefore, not many people are travel in those months.



2.1 Split Data - Training/Test set

We split the data into the training set and test set. As per the requirement, we will hold out only 6 months of data for the test.

3. Model

For this exercise, we have created three (3) different models. We have created ARIMA and ETS models with full dataset as well as training and test set.

3.1 Auto Models - ARIMA & ETS Full Data

When comparing MAE and MASE, we notice that the Auto ARIMA performed better than the other two models. When comparing RMSE, ETS model is the clear winner. Since we are testing on the same data, there is a high likelihood that model is over-fitting. Hence, we would try another approach to build models with the training data (in sample data) and test the model with the test set (out of sample data).

Table 1: Metrics for each model using full data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA(2,0,0)(0,1,1)[12] with drift	17726.37	1895929	1139228	-0.0876403	1.997818	0.3670247	-0.0024269
ETS(A,N,A)	123403.47	1881491	1256336	0.1346370	2.231370	0.4047536	0.0182516
NNAR(1,1,2)[12]	101908.45	3145409	2153886	-0.1572497	3.668165	0.6939169	0.4765724

3.2 Auto Models - Train & Test Data

Next, we created all three models again but using the training set. We test the accuracy of each model using the test data. The table below shows the metrics for each model. Based on the results, we can confirm that our previous models built on full dataset were over-fitting.

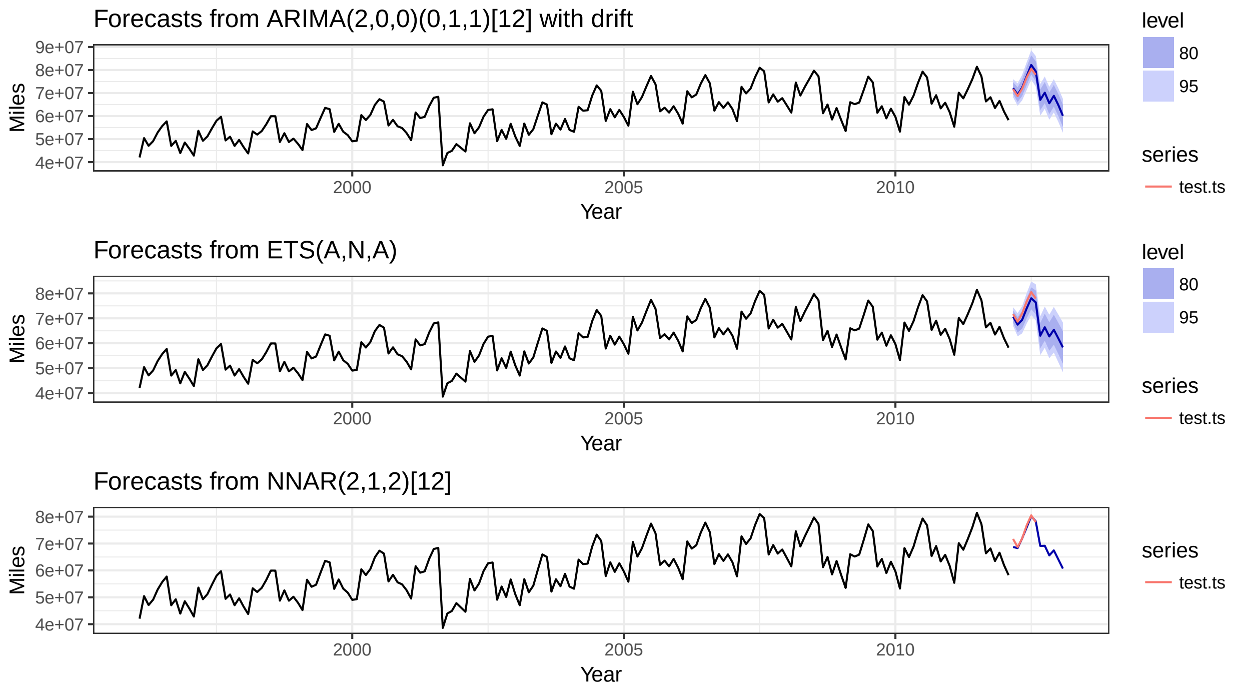
In the result below, when comparing MAE, MAPE, and MASE for test results, we clearly see that Neural Network model performed the best. Therefore, we will select this model as our final model.

Table 2: Metrics for each model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Auto Arima - Train	16700.7772	1923840	1161344.6	-0.0964374	2.043157	0.3653068	-0.0019718
Auto Arima - Test	-887716.1863	1026463	887716.2	-1.1647129	1.164713	0.2792356	0.3591187
Auto ETS - Train	115236.1013	1904285	1273352.1	0.1193085	2.269276	0.4005394	0.0192389
Auto ETS - Test	1912761.2958	2031558	1912761.3	2.5449616	2.544962	0.6016688	0.3539079
Auto Neural Network - Train	506.8029	3219613	2227626.1	-0.3511328	3.825861	0.7007111	0.4944573
Auto Neural Network - Test	675298.2739	1282113	810053.1	0.9314070	1.106301	0.2548063	-0.0369644

4. Forecast

The figure below shows the forecast for each of our models. We forecasted for 12 months so we could see the trend after the test period as well. It appears that neural network model is very accurate in predicting the forecast because both the forecasted and test data overlay each other. This can be observed in the bottom graph. We also notice the similar decreasing trend for the next 6 months as we saw in the previous months.



5. Conclusion

In conclusion, we would select the neural model because it performed better and also the forecast is pretty accurate.

RMarkdown code is available on my GitHub page (https://github.com/Singh-Gurjeet) in the Time Series and Forecasting repository.

6. Appendix I: R Code

```r
knitr::opts_chunk$set(echo = TRUE, warning = F, message = F)
library(tidyverse)
library(fpp)
#Import the data
US.air.traffic  <- read.csv("us-air-carrier-traffic-statistic.csv")
colnames(US.air.traffic) <- c("YearMonth", "Revenue_Miles")
autoplot(ts(US.air.traffic[,"Revenue_Miles"], frequency = 12)) +
  ggtitle(paste0("U.S. Air ", "Carrier Traffic Statistics ",
                 "- Revenue Passenger Miles")) +
  xlab("Year")+ ylab("Miles")+ theme_bw()
# #Split the data into training set and test set
data.ts<-ts(US.air.traffic[,"Revenue_Miles"], frequency=12)
train.ts<-ts(US.air.traffic[1:193,"Revenue_Miles"],end=c(2012,2),frequency=12)
test.ts<-ts(US.air.traffic[194:199,"Revenue_Miles"],start=c(2012,3),frequency=12)
set.seed(123)
#Auto Arima
traffic.arima <- auto.arima(data.ts)
#ETS AUTO Model
```

```r
traffic.ets <- ets(data.ts)
#Neural Network auto selection model
fit_nn <- nnetar(data.ts, lambda=0)
#Metrics
aa.model <- accuracy(traffic.arima)
est.model <- accuracy(traffic.ets)
nn.model <- accuracy(fit_nn)
rownames(aa.model) <- "ARIMA(2,0,0)(0,1,1)[12] with drift"
rownames(est.model) <- "ETS(A,N,A) "
rownames(nn.model) <- "NNAR(1,1,2)[12]"
knitr::kable(rbind(aa.model, est.model, nn.model),
             caption = "Metrics for each model using full data")
#Auto Arima using training dataset
traffic.arima.train <- auto.arima(train.ts)
fcast.arima.train <- forecast(traffic.arima.train, h=12)
acc.arima <- accuracy(fcast.arima.train, test.ts)
rownames(acc.arima) <- c("Auto Arima - Train","Auto Arima - Test")
#Auto ETS using training dataset
traffic.ets.train <- ets(train.ts)
fcast.ets.train <- forecast(traffic.ets.train, h=12)
acc.ets <- accuracy(fcast.ets.train, test.ts)
rownames(acc.ets) <- c("Auto ETS - Train","Auto ETS - Test")
#Auto ETS using training dataset
traffic.nn.train <- nnetar(train.ts)
fcast.nn.train <- forecast(traffic.nn.train, h=12)
acc.nn <- accuracy(fcast.nn.train, test.ts)
rownames(acc.nn) <- c("Auto Neural Network - Train","Auto Neural Network - Test")
knitr::kable(rbind(acc.arima[,1:7], acc.ets[,1:7], acc.nn[,1:7]),
             caption = "Metrics for each model")
AA_plot <- autoplot(fcast.arima.train) + autolayer(test.ts) +
            xlab("Year")+ ylab("Miles")+ theme_bw()
ETS_plot <- autoplot(fcast.ets.train) + autolayer(test.ts) +
            xlab("Year")+ ylab("Miles")+ theme_bw()
NN_plot <- autoplot(fcast.nn.train) + autolayer(test.ts) +
            xlab("Year")+ ylab("Miles")+theme_bw()
gridExtra::grid.arrange(AA_plot,ETS_plot,NN_plot, nrow = 3)
```