Capstone project Sales prediction

Shivani Gupta

07/08/2021

# > Setting working directory

path<-"D:/data science/Capstone Project"  
setwd(path)

# Reading the data file

Sales\_data<-read.csv("sales\_case\_study.csv", header=T)  
head(Sales\_data, 10)

## SKU ISO\_Week Sales Season  
## 1 ProductA 2018-01 0 WINTER  
## 2 ProductA 2018-02 0 WINTER  
## 3 ProductA 2018-03 0 WINTER  
## 4 ProductA 2018-04 6988 WINTER  
## 5 ProductA 2018-04 6988 WINTER  
## 6 ProductA 2018-05 6743 WINTER  
## 7 ProductA 2018-06 4112 WINTER  
## 8 ProductA 2018-07 5732 WINTER  
## 9 ProductA 2018-08 NA WINTER  
## 10 ProductA 2018-09 5559 SPRING

#Subsetting the dataframe into 3 SKUs

ProductA<-subset(Sales\_data, SKU=="ProductA")  
ProductB<-subset(Sales\_data, SKU=="ProductB")  
ProductC<-subset(Sales\_data, SKU=="ProductC")

**EDA**

# Cleaning the Data set

*Initial zero removal* There are some SKU’s for which initial week’s sales values are 0. It means sales started only after that period. Those weeks needs to be removed before fitting the data into the model.

*considering Initial weeks to be till 5th week* *We remove zero sales upto 5th week for all SKUs if any.*

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

filter(ProductA, Sales=="0")

## SKU ISO\_Week Sales Season  
## 1 ProductA 2018-01 0 WINTER  
## 2 ProductA 2018-02 0 WINTER  
## 3 ProductA 2018-03 0 WINTER

ProductA<-ProductA[-c(1,2,3),]  
  
filter(ProductB, Sales=="0")

## SKU ISO\_Week Sales Season  
## 1 ProductB 2018-04 0 WINTER  
## 2 ProductB 2018-13 0 SPRING  
## 3 ProductB 2018-14 0 SPRING  
## 4 ProductB 2018-15 0 SPRING  
## 5 ProductB 2018-16 0 SPRING  
## 6 ProductB 2018-17 0 SPRING  
## 7 ProductB 2018-18 0 SPRING  
## 8 ProductB 2018-19 0 SPRING  
## 9 ProductB 2018-27 0 SUMMER  
## 10 ProductB 2018-28 0 SUMMER  
## 11 ProductB 2018-30 0 SUMMER  
## 12 ProductB 2018-31 0 SUMMER  
## 13 ProductB 2018-32 0 SUMMER  
## 14 ProductB 2018-40 0 AUTUMN  
## 15 ProductB 2018-41 0 AUTUMN  
## 16 ProductB 2018-43 0 AUTUMN  
## 17 ProductB 2018-44 0 AUTUMN  
## 18 ProductB 2018-45 0 AUTUMN

# No initial zeros in Product B as NA values will be replaced by average on later step  
  
filter(ProductC, Sales=="0")

## [1] SKU ISO\_Week Sales Season   
## <0 rows> (or 0-length row.names)

# There are no initial zeros in Product C

*Duplicate Value Treatment* Removing duplicate rows

duplicated(ProductA)

## [1] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE

ProductA[duplicated(ProductA),]

## SKU ISO\_Week Sales Season  
## 5 ProductA 2018-04 6988 WINTER  
## 17 ProductA 2018-15 10012 SPRING

ProductA<-distinct(ProductA)  
  
duplicated(ProductB)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE FALSE FALSE

ProductB[duplicated(ProductB),]

## SKU ISO\_Week Sales Season  
## 63 ProductB 2018-08 219 WINTER

ProductB<-distinct(ProductB)  
  
duplicated(ProductC)

## [1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

ProductC[duplicated(ProductC),]

## SKU ISO\_Week Sales Season  
## 113 ProductC 2018-15 5533 SPRING

ProductC<-distinct(ProductC)

*Missing Value Treatment*

sum(is.na(ProductA))

## [1] 4

Averages<-ProductA %>% group\_by(Season) %>% summarise(average = mean(Sales, na.rm=TRUE))  
ProductA[5,3]<-9600.125  
ProductA[15,3]<-9027.154  
ProductA[26,3]<-5942.091  
ProductA[27,3]<-5942.091  
  
sum(is.na(ProductB))

## [1] 3

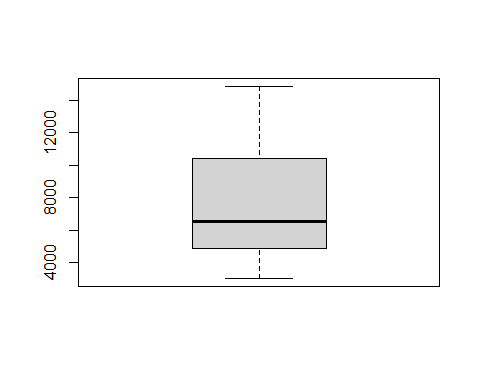
AveragesB<-ProductB %>% group\_by(Season) %>% summarise(average = mean(Sales, na.rm=TRUE))  
ProductB[1,3]<-397.8750  
ProductB[2,3]<-397.8750  
ProductB[3,3]<-397.8750  
  
sum(is.na(ProductC))

## [1] 0

# No NA values present in Product C

*Outlier treatment* Product A

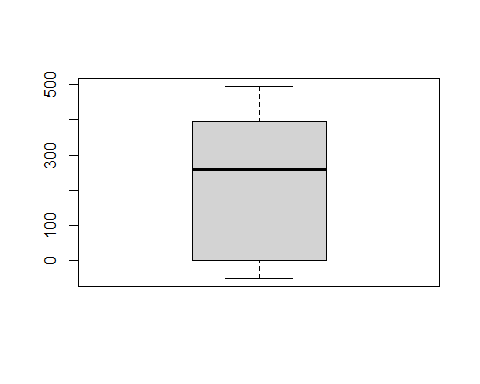
boxplot(ProductA$Sales)



# No outlier in Product A

Product B

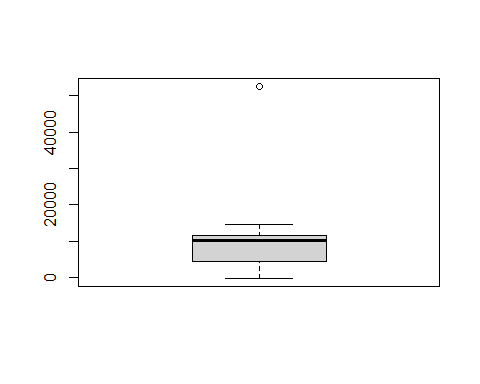
boxplot(ProductB$Sales)



# No outlier in Product B

# Replacing outlier with mean in Product C

boxplot(ProductC$Sales)



AveragesC<-ProductC %>% group\_by(Season) %>% summarise(average = mean(Sales, na.rm=TRUE))  
outliers <- boxplot(ProductC$Sales, plot=FALSE)$out  
ProductC$Sales<-replace(ProductC$Sales, ProductC$Sales==52524,14072.385)

*Replacing negative values with zero*

ProductA[ProductA$Sales<0,]

## [1] SKU ISO\_Week Sales Season   
## <0 rows> (or 0-length row.names)

ProductB[ProductB$Sales<0,]

## SKU ISO\_Week Sales Season  
## 12 ProductB 2018-12 -50 SPRING  
## 29 ProductB 2018-29 -45 SUMMER  
## 42 ProductB 2018-42 -23 AUTUMN

ProductB$Sales<-replace(ProductB$Sales, ProductB$Sales<0, 0)  
  
ProductC[ProductC$Sales<0,]

## SKU ISO\_Week Sales Season  
## 7 ProductC 2018-17 -111 SPRING  
## 8 ProductC 2018-18 -149 SPRING  
## 9 ProductC 2018-19 -163 SPRING  
## 10 ProductC 2018-20 -119 SPRING

ProductC$Sales<-replace(ProductC$Sales, ProductC$Sales<0, 0)

#Checking summary and structure

summary(ProductA)

## SKU ISO\_Week Sales Season   
## Length:49 Length:49 Min. : 3036 Length:49   
## Class :character Class :character 1st Qu.: 4874 Class :character   
## Mode :character Mode :character Median : 6568 Mode :character   
## Mean : 7519   
## 3rd Qu.:10410   
## Max. :14853

str(ProductA)

## 'data.frame': 49 obs. of 4 variables:  
## $ SKU : chr "ProductA" "ProductA" "ProductA" "ProductA" ...  
## $ ISO\_Week: chr "2018-04" "2018-05" "2018-06" "2018-07" ...  
## $ Sales : num 6988 6743 4112 5732 9600 ...  
## $ Season : chr "WINTER" "WINTER" "WINTER" "WINTER" ...

summary(ProductB)

## SKU ISO\_Week Sales Season   
## Length:52 Length:52 Min. : 0.0 Length:52   
## Class :character Class :character 1st Qu.: 0.0 Class :character   
## Mode :character Mode :character Median :259.5 Mode :character   
## Mean :221.1   
## 3rd Qu.:393.5   
## Max. :495.0

str(ProductB)

## 'data.frame': 52 obs. of 4 variables:  
## $ SKU : chr "ProductB" "ProductB" "ProductB" "ProductB" ...  
## $ ISO\_Week: chr "2018-01" "2018-02" "2018-03" "2018-04" ...  
## $ Sales : num 398 398 398 0 446 ...  
## $ Season : chr "WINTER" "WINTER" "WINTER" "WINTER" ...

summary(ProductC)

## SKU ISO\_Week Sales Season   
## Length:42 Length:42 Min. : 0 Length:42   
## Class :character Class :character 1st Qu.: 4415 Class :character   
## Mode :character Mode :character Median :10192 Mode :character   
## Mean : 8244   
## 3rd Qu.:11592   
## Max. :14521

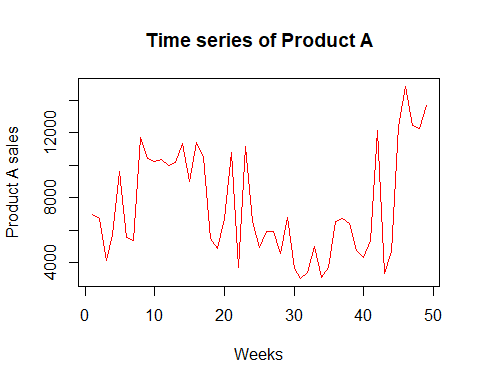
str(ProductC)

## 'data.frame': 42 obs. of 4 variables:  
## $ SKU : chr "ProductC" "ProductC" "ProductC" "ProductC" ...  
## $ ISO\_Week: chr "2018-11" "2018-12" "2018-13" "2018-14" ...  
## $ Sales : num 5495 6330 6144 6383 5533 ...  
## $ Season : chr "SPRING" "SPRING" "SPRING" "SPRING" ...

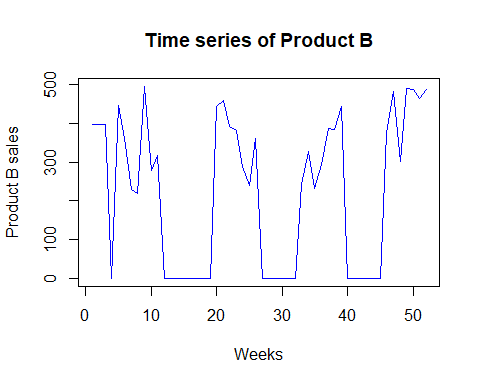
# Visualising Data

*Product A*

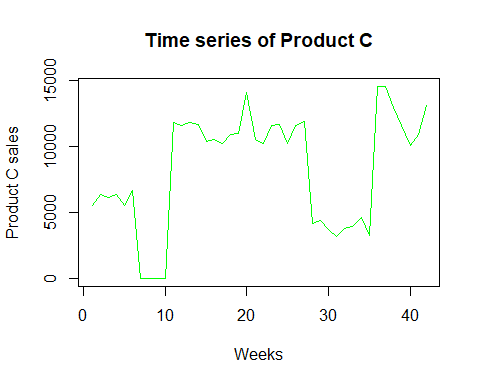
plot.ts(ProductA$Sales, col="Red", main="Time series of Product A", ylab="Product A sales", xlab="Weeks")

 *Product B*

plot.ts(ProductB$Sales, col="Blue", main="Time series of Product B", ylab="Product B sales", xlab="Weeks")

 *Product C*

plot.ts(ProductC$Sales, col="Green", main="Time series of Product C", ylab="Product C sales", xlab="Weeks")



#Dividing into Training And Testing Dataset

trainA<-ProductA[1:38,]  
TestA<-ProductA[39:49,]  
  
trainB<-ProductB[1:41,]  
TestB<-ProductB[42:52,]  
  
trainC<-ProductC[1:31,]  
TestC<-ProductC[32:42,]

# Fitting into model ARIMA

#install.packages("forecast")  
library(forecast)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

fitA<-auto.arima(as.ts(trainA$Sales), stepwise=FALSE, approximation=FALSE)  
fitB<-auto.arima(as.ts(trainB$Sales), stepwise=FALSE, approximation=FALSE)  
fitC<-auto.arima(as.ts(trainC$Sales), stepwise=FALSE, approximation=FALSE)  
  
  
ForecastA <-forecast(fitA,h=11)  
ForecastB <-forecast(fitB,h=11)  
ForecastC <-forecast(fitC,h=11)  
  
  
accuracy(ForecastA,TestA$Sales)

## ME RMSE MAE MPE MAPE MASE  
## Training set -63.58775 2330.254 1795.271 -11.30124 29.95325 0.8735753  
## Test set 3252.24100 5394.183 4494.402 14.37814 45.15375 2.1869671  
## ACF1  
## Training set -0.051369  
## Test set NA

accuracy(ForecastB, TestB$Sales)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -3.254217 155.0201 125.0011 -Inf Inf 1.203127 -0.0190718  
## Test set 97.384994 218.9796 206.9214 -Inf Inf 1.991604 NA

accuracy(ForecastC, TestC$Sales)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 61.94718 2814.795 1852.785 -Inf Inf 1.222697 0.04304937  
## Test set 3073.56723 4818.061 4087.098 13.81509 41.97441 2.697175 NA

# Forecasted data using ARIMA

Fr\_salesA<-data.frame(ForecastA)  
Fr\_salesB<-data.frame(ForecastB)  
Fr\_salesC<-data.frame(ForecastC)  
  
output\_table<-read.csv("output.csv")  
Fr\_salesA[,1]

## [1] 5852.577 5852.577 5852.577 5852.577 5852.577 5852.577 5852.577 5852.577  
## [9] 5852.577 5852.577 5852.577

SKU<-output\_table$SKU  
ISO\_week<-output\_table$ISO\_Week  
Pred\_Arima<-c(Fr\_salesA$Point.Forecast,Fr\_salesB$Point.Forecast,Fr\_salesC$Point.Forecast)  
output<-data.frame(cbind(SKU,ISO\_week,Pred\_Arima))  
print(output)

## SKU ISO\_week Pred\_Arima  
## 1 ProductA 2018-42 5852.57717848706  
## 2 ProductA 2018-43 5852.57717848706  
## 3 ProductA 2018-44 5852.57717848706  
## 4 ProductA 2018-45 5852.57717848706  
## 5 ProductA 2018-46 5852.57717848706  
## 6 ProductA 2018-47 5852.57717848706  
## 7 ProductA 2018-48 5852.57717848706  
## 8 ProductA 2018-49 5852.57717848706  
## 9 ProductA 2018-50 5852.57717848706  
## 10 ProductA 2018-51 5852.57717848706  
## 11 ProductA 2018-52 5852.57717848706  
## 12 ProductB 2018-42 95.1183691737332  
## 13 ProductB 2018-43 146.055997014545  
## 14 ProductB 2018-44 173.334029140075  
## 15 ProductB 2018-45 187.941914669547  
## 16 ProductB 2018-46 195.764704768927  
## 17 ProductB 2018-47 199.953952063855  
## 18 ProductB 2018-48 202.197370674748  
## 19 ProductB 2018-49 203.398762406813  
## 20 ProductB 2018-50 204.042129670892  
## 21 ProductB 2018-51 204.386664617688  
## 22 ProductB 2018-52 204.571169388433  
## 23 ProductC 2018-42 4255.91404243336  
## 24 ProductC 2018-43 5063.49963112302  
## 25 ProductC 2018-44 5657.5301816511  
## 26 ProductC 2018-45 6094.47742278904  
## 27 ProductC 2018-46 6415.87990178331  
## 28 ProductC 2018-47 6652.29186134767  
## 29 ProductC 2018-48 6826.18788977892  
## 30 ProductC 2018-49 6954.09947580817  
## 31 ProductC 2018-50 7048.18658054609  
## 32 ProductC 2018-51 7117.39362857644  
## 33 ProductC 2018-52 7168.29981290867

# Fitting data into Model ETS

modelA<-ets(trainA$Sales)  
modelB<-ets(trainB$Sales)  
modelC<-ets(trainC$Sales)  
  
PredictA<-predict(modelA, h=11)  
PredictB<-predict(modelB, h=11)  
PredictC<-predict(modelC, h=11)  
  
accuracy(PredictA,TestA$Sales)

## ME RMSE MAE MPE MAPE MASE  
## Training set -61.32686 2329.721 1813.403 -11.41182 30.29308 0.8823982  
## Test set 3295.52809 5420.392 4498.337 15.01142 44.88702 2.1888820  
## ACF1  
## Training set -0.03874678  
## Test set NA

accuracy(PredictB, TestB$Sales)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -12.38388 167.1241 108.0049 -Inf Inf 1.039539 0.007062978  
## Test set 239.10879 324.3193 269.3768 -Inf Inf 2.592732 NA

accuracy(PredictC, TestC$Sales)

## ME RMSE MAE MPE MAPE MASE  
## Training set -84.18031 2970.083 1521.921 -Inf Inf 1.004352  
## Test set 6148.70032 7536.322 6150.275 52.61385 52.66289 4.058715  
## ACF1  
## Training set 0.0001862073  
## Test set NA

#Forecasted data using ets

Pr\_salesA<-data.frame(PredictA)  
Pr\_salesB<-data.frame(PredictB)  
Pr\_salesC<-data.frame(PredictC)  
  
  
  
SKU<-output\_table$SKU  
ISO\_week<-output\_table$ISO\_Week  
Pred\_Arima<-c(Fr\_salesA$Point.Forecast,Fr\_salesB$Point.Forecast,Fr\_salesC$Point.Forecast)  
output<-data.frame(cbind(SKU,ISO\_week,Pred\_Arima))  
output$Pred\_ets<-c(Pr\_salesA$Point.Forecast,Pr\_salesB$Point.Forecast,Pr\_salesC$Point.Forecast)  
print(output)

## SKU ISO\_week Pred\_Arima Pred\_ets  
## 1 ProductA 2018-42 5852.57717848706 5809.29009  
## 2 ProductA 2018-43 5852.57717848706 5809.29009  
## 3 ProductA 2018-44 5852.57717848706 5809.29009  
## 4 ProductA 2018-45 5852.57717848706 5809.29009  
## 5 ProductA 2018-46 5852.57717848706 5809.29009  
## 6 ProductA 2018-47 5852.57717848706 5809.29009  
## 7 ProductA 2018-48 5852.57717848706 5809.29009  
## 8 ProductA 2018-49 5852.57717848706 5809.29009  
## 9 ProductA 2018-50 5852.57717848706 5809.29009  
## 10 ProductA 2018-51 5852.57717848706 5809.29009  
## 11 ProductA 2018-52 5852.57717848706 5809.29009  
## 12 ProductB 2018-42 95.1183691737332 41.61848  
## 13 ProductB 2018-43 146.055997014545 41.61848  
## 14 ProductB 2018-44 173.334029140075 41.61848  
## 15 ProductB 2018-45 187.941914669547 41.61848  
## 16 ProductB 2018-46 195.764704768927 41.61848  
## 17 ProductB 2018-47 199.953952063855 41.61848  
## 18 ProductB 2018-48 202.197370674748 41.61848  
## 19 ProductB 2018-49 203.398762406813 41.61848  
## 20 ProductB 2018-50 204.042129670892 41.61848  
## 21 ProductB 2018-51 204.386664617688 41.61848  
## 22 ProductB 2018-52 204.571169388433 41.61848  
## 23 ProductC 2018-42 4255.91404243336 3220.66332  
## 24 ProductC 2018-43 5063.49963112302 3220.66332  
## 25 ProductC 2018-44 5657.5301816511 3220.66332  
## 26 ProductC 2018-45 6094.47742278904 3220.66332  
## 27 ProductC 2018-46 6415.87990178331 3220.66332  
## 28 ProductC 2018-47 6652.29186134767 3220.66332  
## 29 ProductC 2018-48 6826.18788977892 3220.66332  
## 30 ProductC 2018-49 6954.09947580817 3220.66332  
## 31 ProductC 2018-50 7048.18658054609 3220.66332  
## 32 ProductC 2018-51 7117.39362857644 3220.66332  
## 33 ProductC 2018-52 7168.29981290867 3220.66332