Product Demand Prediction with Machine Learning-Guidelines

Name : Sam Suman Raj E

Regno : 721221106086

Email : samraj51170@gmail.com

|  |
| --- |
|  |

**INTRODUCTION**

What Is Our Ambition?

**We don't aim here for a perfect prediction, but rather to show that this dataset has a potential to make good forecasts.**

**Hence, we shall here stick to some trivial models, and invite the reader to pursuie this work if s.he is interested.**

**Our personnal goal is to train on a really raw dataset, whose structure is not fit yet for machine learning, and for which we can set:**

* **Our own objective : making forecast of next month worldwide orders for each product ;**
* **Our own criteria of succes : minimizing the mean forecasting error ;**

**Importing the Libraries**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**%matplotlib inline**

**import seaborn as sns**

**import plotly.express as px**

**import bokeh**

**from scipy import stats**

**import date time**

**import os**

**import pickle**

**import requests**

**Importing Data**

**df = pd.read\_csv(r"/input/productdemandforecasting/Historical Product Demand.csv")**

**df.head(5)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product\_Code | Warehouse | Product\_Category | Date | Order\_Demand |
| Product\_0993 | Whse\_J | Category\_028 | 2012/7/27 | 100 |
| Product\_0979 | Whse\_J | Category\_028 | 2012/1/19 | 500 |
| Product\_0979 | Whse\_J | Category\_028 | 2012/2/3 | 500 |
| Product\_0979 | Whse\_J | Category\_028 | 2012/2/9 | 500 |
| Product\_0979 | Whse\_J | Category\_028 | 2012/3/2 | 500 |
| Product\_0979 | Whse\_J | Category\_028 | 2012/4/19 | 500 |

Cleaning and preprocessing

rm(list=ls())

d <- read.table(file = "../input/Historical Product Demand.csv",h=T,sep=",")

d <- na.omit(d)

head(d)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Product­­­\_Code | Warehouse | Product Category | Date | Order\_Demand |
| count | 1048575 | 1048575 | 1048575 | 1037336 | 1048575 |
| Unique | 2160 | 4 | 33 | 1729 | 3828 |
| Top | 1359 | Whse\_J | 019 | 2013 | 1000 |
| freq | 16936 | 764447 | 481099 | 2075 | 112682 |

**Let us focus on the minimum number of product we need to focus on to capture the max number of orders. This step is imprtant if we want to focus on important products, that have a good chance to have enough orders for prediction.**

tab <- table(d["Product\_Code"])

tab <- tab[order(tab,decreasing = TRUE)]

plot(cumsum(tab)/dim(d)[1],main="Cumulated sum of number of orders",

xlab="# of products",ylab="Volume")

selection <- 670

#80% of orders (in volume) concern only 670 products of the 2160 in the dataset

selection <- tab[1:selection]

selection <- rownames(selection)

in\_selection <- function(prod\_id){

return(is.element(prod\_id,selection))

}

are\_in <- tapply(d["Product\_Code"][,1],INDEX=rep(1,dim(d)[1]),FUN = in\_selection)[[1]]

ds <- subset(d,are\_in,select = colnames(d))

ds <- droplevels(ds)

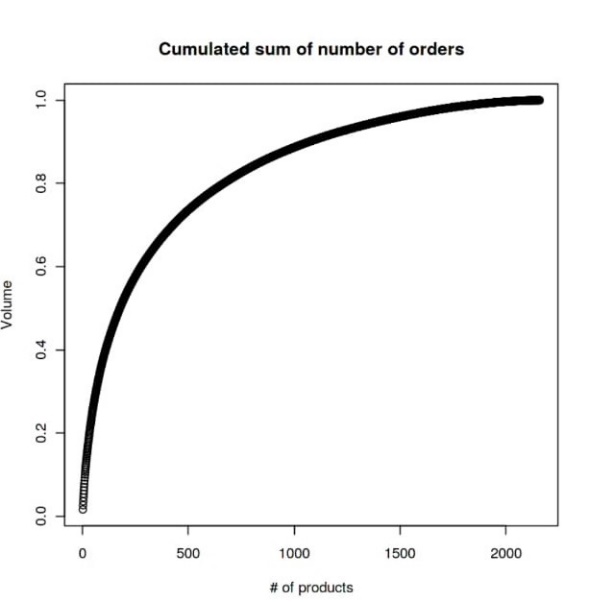
Product\_Category <- ds[c("Product\_Code","Product\_Category")]

Warehouse <- ds[c("Product\_Code","Warehouse")]

ds["Warehouse"] <- NULL

ds["Product\_Category"] <- NULL # of interest only later

ds$Order\_Demand <- as.numeric(ds$Order\_Demand)



**Before predicting anything, we want to group the command by month, as we try, for each product, to predict the next month global demand.**

ds["YeMo"] <- format(as.Date(ds$Date), format="%Y/%m/00")

ds$Date <- NULL

library(data.table)

dsa <- data.table(ds)

setkey(dsa, Product\_Code)

dsa <- dsa[, lapply(.SD, sum), by = list(Product\_Code,YeMo)]

head(dsa)

|  |
| --- |
| Product\_ Code Order\_Demand YeMo |
| Product\_0001 2012/01/00 20470 |
| Product\_0001 2012/05/00 10585 |
| Product\_0001 2012/08/00 24945 |
| Product\_0001 2012/02/00 21574 |
| Product\_0001 2012/03/00 18479 |
| Product\_0001 2012/04/00 12452 |

**Let us now create a dataset where, for each product considered, we have access to the Order\_Demand of each month (wether it is null or not). While doing so, we also gain some insight on the data.**

What is our range of dates ?

|  |
| --- |
| ye <- dsa$YeMo[order(dsa$YeMo)]  range(ye)  timeframe <- 8+5\*12+1 #69 months |

'2011/05/00' '2017/01/00'

We now code each month by an integer ID, and fill our time-series data-frame.

dates <- dsa$YeMo

YeMo\_to\_int <- function(bad\_date){

a <- strsplit(bad\_date,split="/")[[1]]

y <- (strtoi(a[1])-2011)\*12

m <- strtoi(a[2],10)-4

return(y+m)

}

# test:

#YeMo\_to\_int("2011/05/00")

#YeMo\_to\_int("2017/01/00") #Ok

dsa$timestamp <- unlist(lapply(dates, FUN= YeMo\_to\_int))

dsa$YeMo <- NULL

#test :

#sum(is.na(dsa$timestamp)) #ok

products <- as.character(unique(dsa$Product\_Code))

dtf <- data.frame(matrix(data = 0, nrow = length(products), ncol=timeframe), row.names = products)

for (pXm in 1:dim(dsa)[1]){

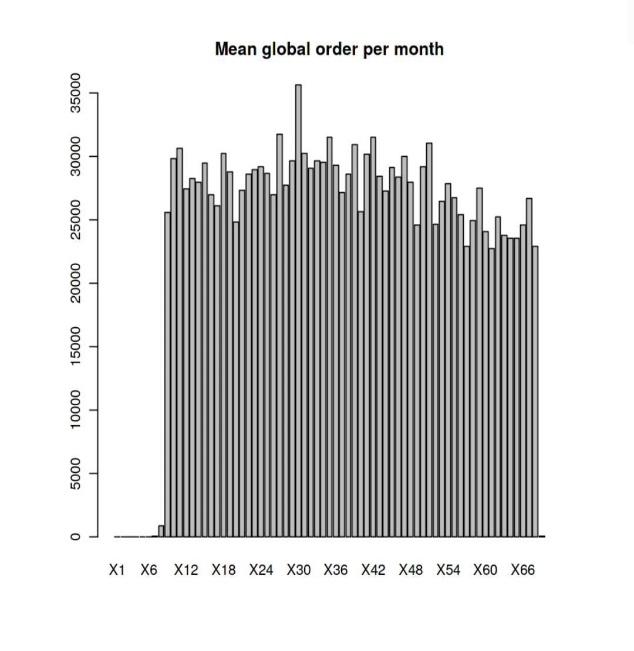
dtf[as.character(unlist(dsa[pXm,"Product\_Code"])),unlist(dsa[pXm,"timestamp"])]<-

unlist(dsa[pXm,"Order\_Demand"])

}

PRUNE OUR DATA

barplot(apply(dtf,2,mean),main="Mean global order per month")

****

We observe 9 months are order of magnitude below the others. They will not be very informative for our prediction.

*Product categories*

padded\_category\_data = {}

for category, category\_data in category\_yearly\_demand.groupby(level = 'Product\_Category'):

print(f"Category: {category}")

padded\_category\_data[category] = [0 for \_ in range(7)]

for year, total\_demand in category\_data.items():

index = ((year[0] - 2010) % 7) - 1

padded\_category\_data[category][index] = total\_demand

fig = plt.figure(figsize=(12, 25))

rows, cols = 11, 3

x = [2011, 2012, 2013, 2014, 2015, 2016, 2017]

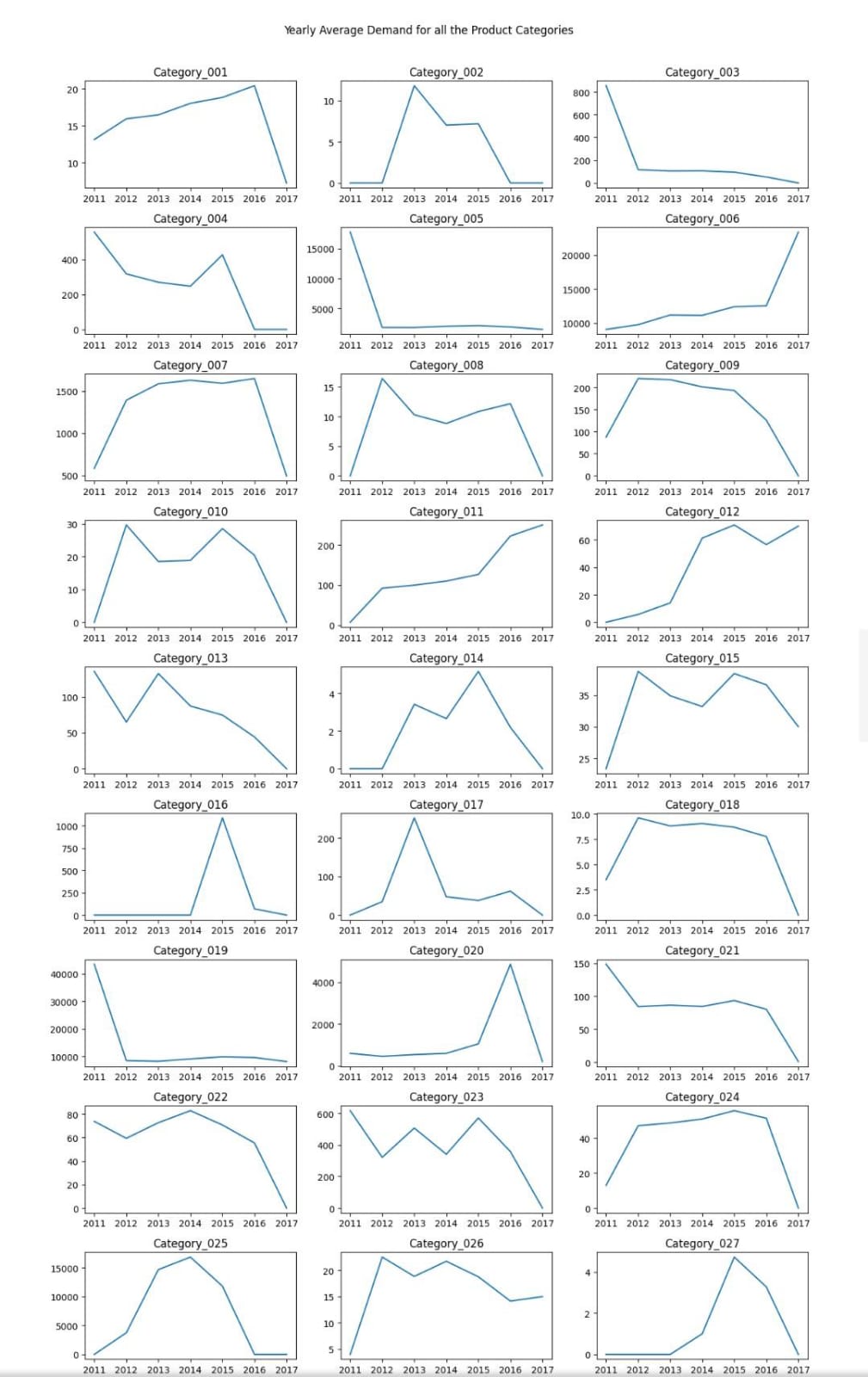
for title, data in padded\_category\_data.items():

ax = fig.add\_subplot(rows, cols, int(title[-2:]))

ax.plot(x, data)

ax.set\_title(title)

plt.tight\_layout()

fig.suptitle("Yearly Average Demand for all the Product Categories", y=1.02) ****

Yearwise Average Demand of all Product

padded\_yearly\_categories = {}

for year, year\_data in category\_yearly\_demand.groupby(level = 'Date'):

# print(f"Category: {category}")

padded\_yearly\_categories[year] = [0 for \_ in range(33)]

for category, total\_demand in year\_data.items():

# print(category)

index = (int(category[1][-2:]) % 33) - 1

padded\_yearly\_categories[year][index] = total\_demand

x = [i+1 for i in range(33)]

rows = len(padded\_yearly\_categories)

cols = 1

fig, axes = plt.subplots(nrows=rows, ncols=cols, figsize=(10, 20))

colors = plt.cm.viridis(np.linspace(0, 1, len(x)))

for i, (year, data) in enumerate(padded\_yearly\_categories.items()):

bars = axes[i].bar(x, data, color=colors)

axes[i].bar\_label(bars, labels=x, fontsize = 8)

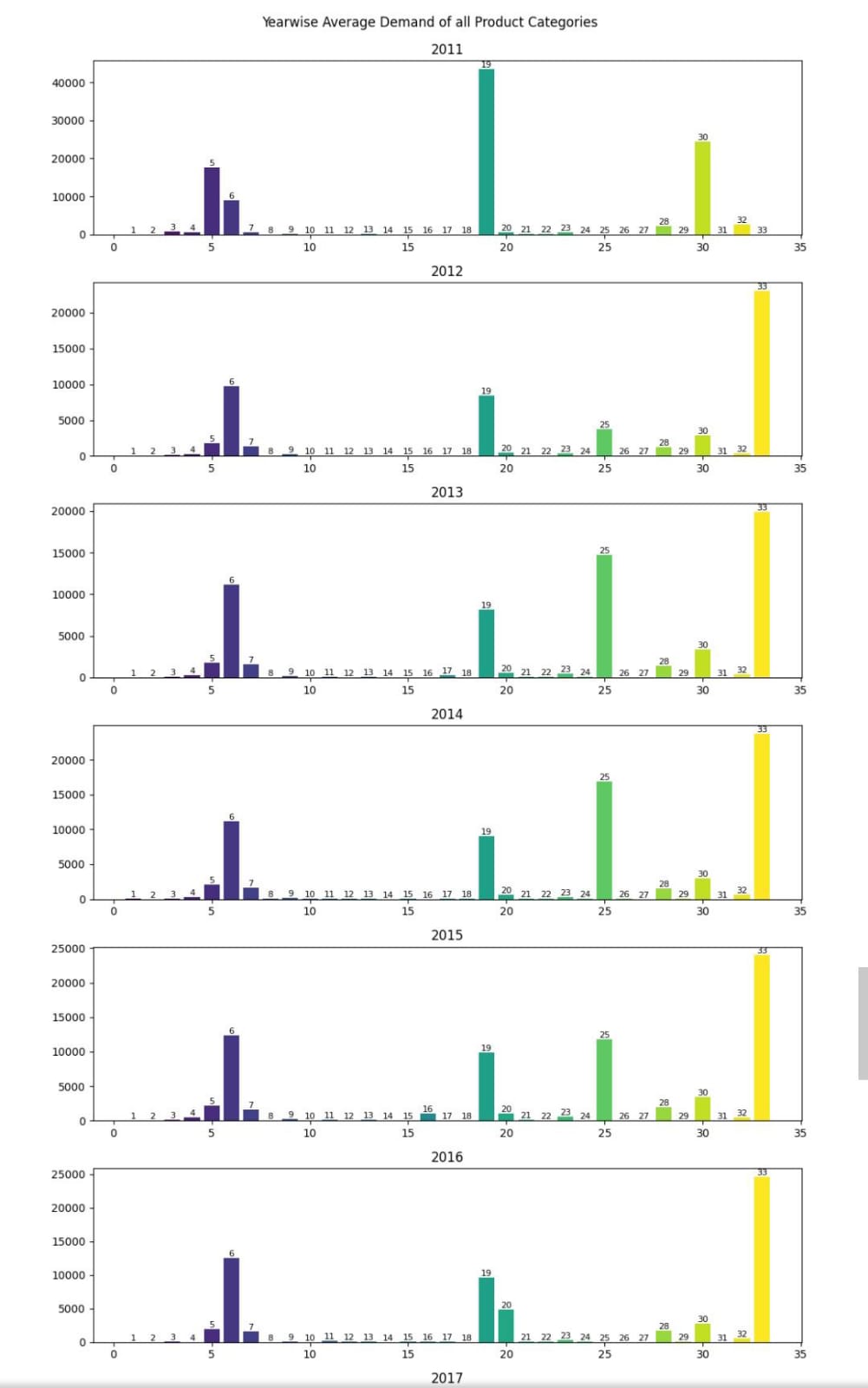
axes[i].set\_title(year)

axes[row\_idx, col\_idx].get\_yaxis().set\_visible(False)

plt.tight\_layout()

fig.suptitle("Yearwise Average Demand of all Product Categories", y=1.01)

plt.show()

****warehouse\_yearly\_demand = df.groupby([df.index.year, 'Warehouse'])['Order\_Demand'].mean()

Yearly Average Demand by Warehouse

demand\_data = {}

warehouses = []

years = [2011 + i for i in range(7)]

for warehouse, warehouse\_data in warehouse\_yearly\_demand.groupby(level='Warehouse'):

warehouses.append(warehouse)

demand\_data[warehouse] = [0 for i in range(7)]

for year, year\_data in warehouse\_data.items():

index = ((year[0] - 2010) % 7) - 1

demand\_data[warehouse][index] = year\_data

num\_warehouses = len(warehouses)

num\_years = len(years)

bar\_width = 0.15

fig, ax = plt.subplots(figsize=(12, 8))

x = np.arange(num\_years)

# Create a grouped bar chart

for i, warehouse in enumerate(warehouses):

x\_pos = x + i \* bar\_width

ax.bar(x\_pos, demand\_data[warehouse], width=bar\_width, label=warehouse)

ax.set\_xticks(x + (num\_warehouses - 1) \* bar\_width / 2)

ax.set\_xticklabels(years)

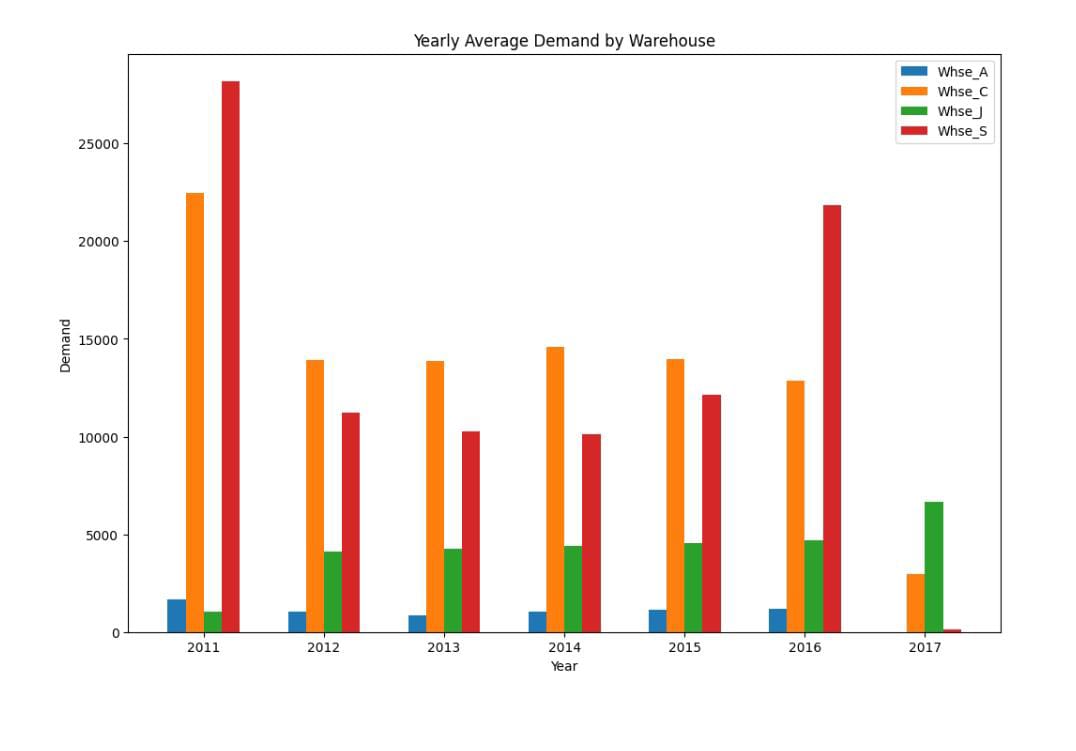
ax.set\_xlabel('Year')

ax.set\_ylabel('Demand')

ax.set\_title('Yearly Average Demand by Warehouse')

ax.legend()

plt.show()

****

Conclusion

Customers today expect effective products and hassle free on-time services. These expectations could not be met without a strong supply-chain that involves strategic planning that includes demand forecasting.

The solution in this white paper is a statistical and ML-based solution that creates timeseries regarding each product and its entitlements based on geographic locations. The inputs of renewal rates and holidays based on each country or region helped generate accurate results by count and rate-based forecast on weekly basis. These forecasts assist the business in parts procurements and help budget planning for each financial year.

The Demand Forecasting project was originally used by services finance and part planning teams. But it has the potential to broaden its horizon by expanding the scope of the forecasting project and changing the granularity of forecast with expanded end users.