INTRODUCTION

Supermarkets are large retail stores operated on a self-service basis, selling groceries, meats, dairy products and many more. The growth of supermarkets is increasing day by day and market competitions are also high. We have one similar dataset wherein we will be creating different time intelligence measure and using different visualization tools to understand and find the insights of sales of various products sold inside the supermarket. We will also calculate the profit margins on each and every product sold in the supermarket along with the product return rate, different KPI's and creating forecasting of the weekly profit for next three month using the visualization analytics options.

DATASET DESCRIPTION AND SOURCE

The dataset contains 8 xlsx file which are as mentioned below.

1. MavenMarket_Calendar – contains 1 Column and 731 rows pertaining to the date of sales of product
2. MavenMarket_Customers—contains 20 columns and 10282 rows pertaining to the customer details such as their age, sex, income etc.
3. MavenMarket_Products- contains 9 columns and 1561 rows pertaining to the product name, revenue, cost and weight
4. MavenMarket_Regions-contains 03 columns and 110 rows pertaining to the region where the product is sold
5. MavenMarket_Returns_1997-1998- contains 04 columns and 7088 rows pertaining to returns of the products
6. MavenMarket_Stores- contains 13 columns and 25 rows pertaining to stores, their opening and closing dates, regions they fall in etc.
7. MavenMarket_Transactions_1997- contains 06 columns and 86838 rows pertaining to transcation made in correspondence to the products
8. MavenMarket_Transactions_1998- contains 06 columns and 182884 rows pertaining to transcation made in correspondence to the products

DATA PREPARATION AND CLEANING

1. First and the fore most change the regional setting and the autodetect relationship options

Go to File ->options and settings -> options->current file ->Data load -> Deselect the autodetect new relation after data is loaded option (if you want to create relationship on your own)

Go to File ->options and settings -> options->current file ->Regional settings ->English (USA)

2. Connect the data tables, and transform, check the datatypes and clean the unnecessary data
3. Rename all the connected files as either a dimension or a fact table
4. **Calendar dimension** - Go to transform data, select the calendar dimension query, go to add columns, select date and add the day name, month name, start of month, start of year, start of quarter and quarter of the year.

   Add columns as mentioned below –

   Day of the week = WEEKDAY ('calendar dimension'[date],1)

   end of month = ENDOFMONTH ('calendar dimension'[date])

   Short day name = upper (left ('calendar dimension'[Day Name],3))

   Short month name = upper (left ('calendar dimension'[Month Name],3))

   Weekend = IF ('calendar dimension'[Short day name] ="SAT" || 'calendar dimension'[Short day name] ="SUN","Y","N")

   Year = YEAR ('calendar dimension'[date])

5. **customer dimension**- select the customer dimension query and add the following columns.

   Current age = DATEDIFF ('customer dimension'[birthdate], TODAY (), YEAR)

   full name = 'customer dimension'[first name] & " " & 'customer dimension'[last name]

   Has children = IF ('customer dimension'[total children] =0,"N","Y")

   House Number = LEFT ('customer dimension'[customer address], SEARCH (" ", 'customer dimension'[customer address])-1)

   Priority = IF (AND ('customer dimension'[homeowner]="SEARCH (dimension'[member card] ="Golden"),"high", "standard")

   short country = UPPER ((LEFT ('customer dimension'[country/region],3))

6. **Product dimension**- select the customer dimension query and add the following columns.

   price tier = IF ('products dimensions'[columns.]>3,"high", IF ('products dimensions'[columns.]>1,"mid","low"))

7. Connect the fact and the dimension table and create the model manually and keep a note to check the flow direction and relational types is one to many

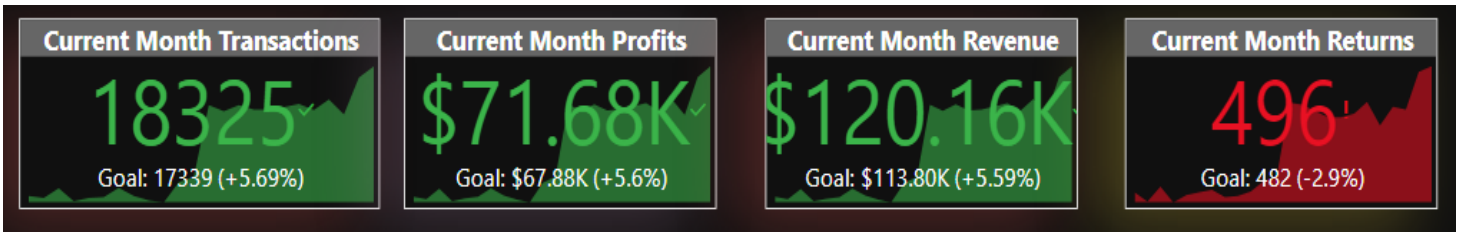
DAX MEASURES

DAX FUNCTION	M-CODE
AVERAGE ()	Average retail price = AVERAGE ('products dimensions'[product cost])
SUMX () RELATED ()	total revenue = SUMX ('transaction fact', 'transaction fact'[Quantity sold] *RELATED ('products dimensions'[columns.])) total cost = SUMX ('transaction fact', 'transaction fact'[Quantity sold] *RELATED ('products dimensions'[product cost])) adjusted revenue = SUMX ('transaction fact', 'transaction fact'[quantity]*'transaction fact'[adjusted price]) adjusted profit = 'transaction fact'[adjusted revenue]-'transaction fact'[total cost] adjusted price = 'transaction fact'[Average retail price] *(1+'Price adjustments'[Price adjustment value]))
CALCULATE () ALL ()	all orders = CALCULATE ('transaction fact'[total orders], ALL ('transaction fact')) all returns = CALCULATE ('transaction fact'[total returns], ALL ('returns fact'))
COUNTROWS ()	total returns = COUNTROWS ('returns fact') Total orders = COUNTROWS ('transaction fact')
SUM ()	Quantity sold = SUM ('transaction fact'[quantity]) Quantity returned = sum ('returns fact'[quantity])
CALCULATE ()	weekend transactions = CALCULATE ('transaction fact'[total orders],'calendar dimension'[Weekend]="Y")
	Revenue targets = [Last month revenue] *1.05 order target = [Last month orders] *1.1
DIVIDE ()	return rate = DIVIDE ([Quantity returned], [Quantity sold]) Profit margin = 'transaction fact'[total profit]/'transaction fact'[total revenue] % Weekend transactions = [weekend transactions]/'transaction fact'[total orders]

	% Weekend transactions = [weekend transactions]/'transaction fact'[total orders] [% of all orders] = 'transaction fact'[total orders]/'transaction fact'[all orders]
CALCULATE () DATEADD ()	Last month revenue = CALCULATE ('transaction fact'[total revenue], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month returns = CALCULATE ('transaction fact'[total returns], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month profit = CALCULATE ([total profit], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month orders = CALCULATE ('transaction fact'[total orders], DATEADD ('calendar dimension'[date], -1, MONTH))
CALCULATE () DATESINPERIOD ()	10-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -10, DAY)) 60-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -60, DAY)) 90-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -90, DAY))
CALCULATE () DATESYTD () DATESQTD () DATESMTD ()	YTD = CALCULATE ('transaction fact'[total revenue], DATESYTD ('calendar dimension'[date])) QTD = CALCULATE ('transaction fact'[total revenue], DATESQTD ('calendar dimension'[date])) MTD = CALCULATE ('transaction fact'[total revenue], DATESMTD ('calendar dimension'[date]))

OBJECTIVES AND GOALS

1. **Create KPI for (a) current month transaction (b) current month profits (c) current month revenue (d) current month returns**

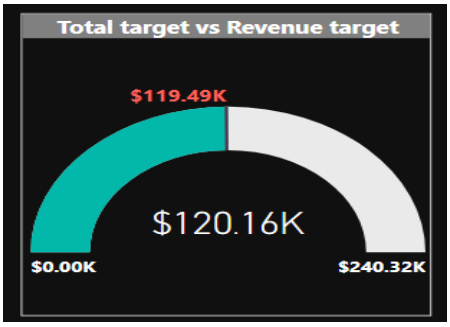


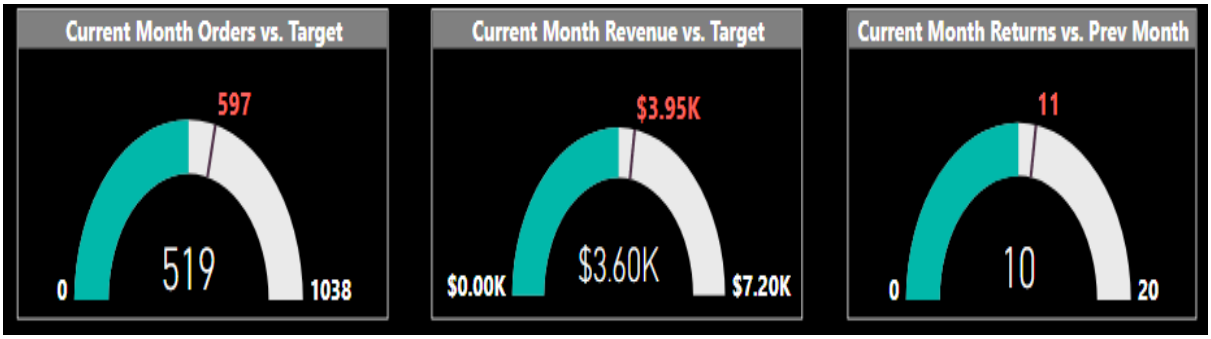
2. **Create time intelligence measures**

CALCULATE () DATEADD ()	Last month revenue = CALCULATE ('transaction fact'[total revenue], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month returns = CALCULATE ('transaction fact'[total returns], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month profit = CALCULATE ([total profit], DATEADD ('calendar dimension'[date], -1, MONTH)) Last month orders = CALCULATE ('transaction fact'[total orders], DATEADD ('calendar dimension'[date], -1, MONTH))

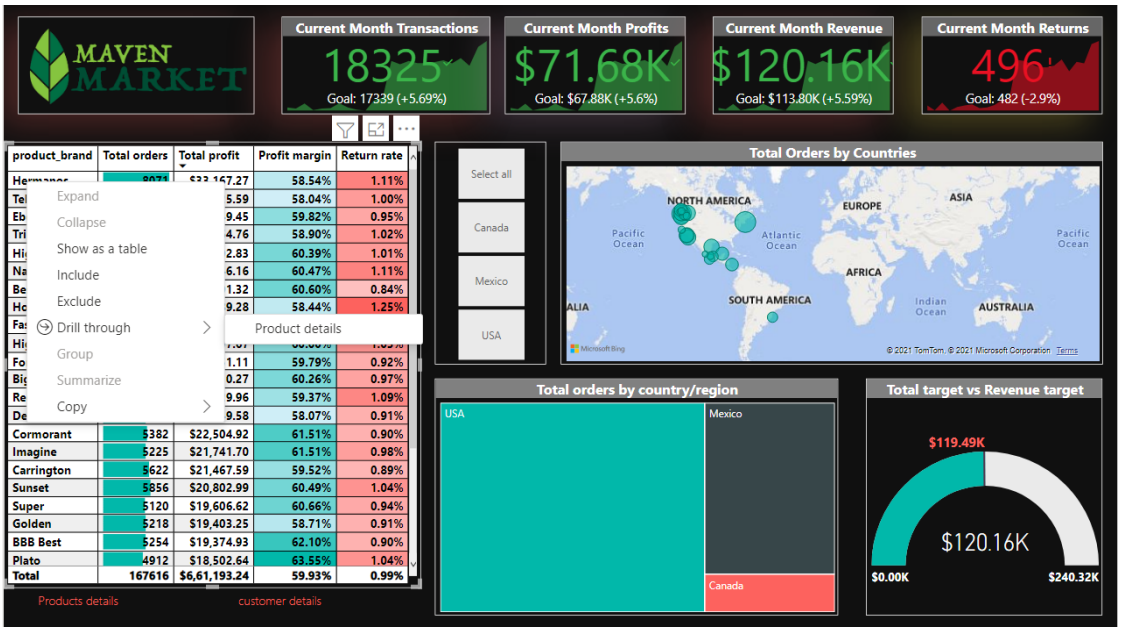
CALCULATE () DATESINPERIOD () 	10-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -10, DAY)) 60-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -60, DAY)) 90-day Rolling Revenue = CALCULATE ('transaction fact'[total revenue], DATESINPERIOD ('calendar dimension'[date], MAX ('calendar dimension'[date]), -90, DAY))
CALCULATE () DATESYTD () DATESQTD () DATESMTD ()	YTD = CALCULATE ('transaction fact'[total revenue], DATESYTD ('calendar dimension'[date])) QTD = CALCULATE ('transaction fact'[total revenue], DATESQTD ('calendar dimension'[date])) MTD = CALCULATE ('transaction fact'[total revenue], DATESMTD ('calendar dimension'[date]))

3. **Using gauge chart show (A) total target vs revenue target(B) Current vs target month order (c) current month vs target revenue (d) current vs previous month returns**

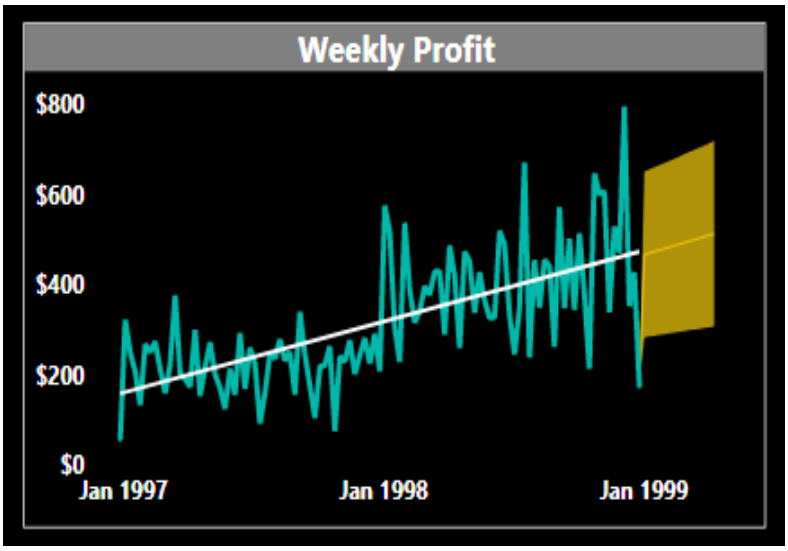


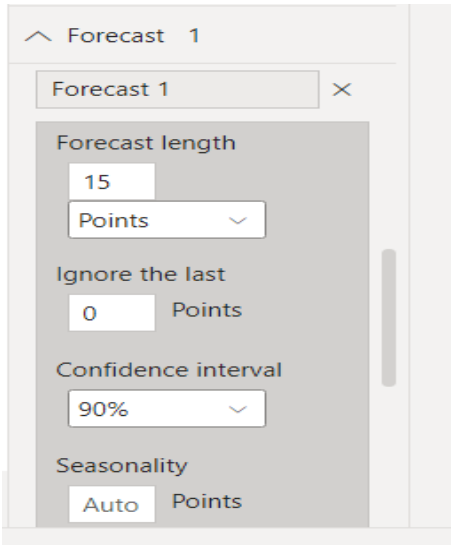


4. **Create a drill through to hover to the product details**



5. **creating forecasting of the weekly profit for next three month using the visualization analytics options**





6. **Customer with highest revenues, profit and orders.**

Full Name	Total orders	Total Revenue
Ida Rodriguez	290	$2,235.43
James Horvat	297	$2,121.31
Dawn Laner	301	$1,995.21
Mary Francis Benigar	275	$1,973.79
Wildon Cameron	233	$1,955.25
Aaron McDonnell	260	$1,907.90
Joann Mramor	251	$1,826.31
Eric Winters	257	$1,776.27
Merridee Archuleta	279	$1,763.58
Lucy Flowers	267	$1,750.14
Kristin Miller	239	$1,738.93
Scott Littleford	224	$1,716.15
Bernadette Marschang	265	$1,713.75
George Todero	262	$1,664.45
Emily Barela	205	$1,630.12
Frank Darrell	221	$1,628.36
Isaiah Heymsfield	197	$1,598.53
Mary Smith	223	$1,595.58
Patricia Towns	241	$1,585.12
Shirley Ortiz	226	$1,583.94
Total	6105	$43,444.59

Customer with highest total profit – IDA RODRIGUEZ

Customer with highest total Revenue – IDA RODRIGUEZ

Customer with highest orders – DAMN LANER

Future scope

There are a lot of insights which can be added to these reports in future, some of them are as listed Below

1. Customer tracking based on regions and country wise.
2. Effect of loyalty card to identify customers with more fragile of mindset
3. Effect of Customised pricing / promotions on the products sold