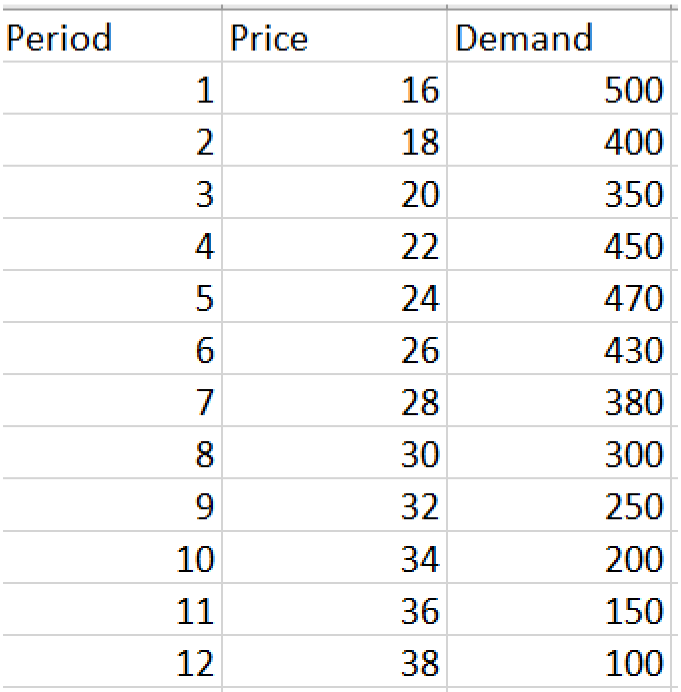
Retail Analytics Report:

Task-1:

Demand-based pricing:

our task is to help client optimize the price of the product. We have given the prices for different weeks and the associated demand for the same. We will try both the linear and logit model and see which one is a better fit to the given data. We will be using excel tool for this task.

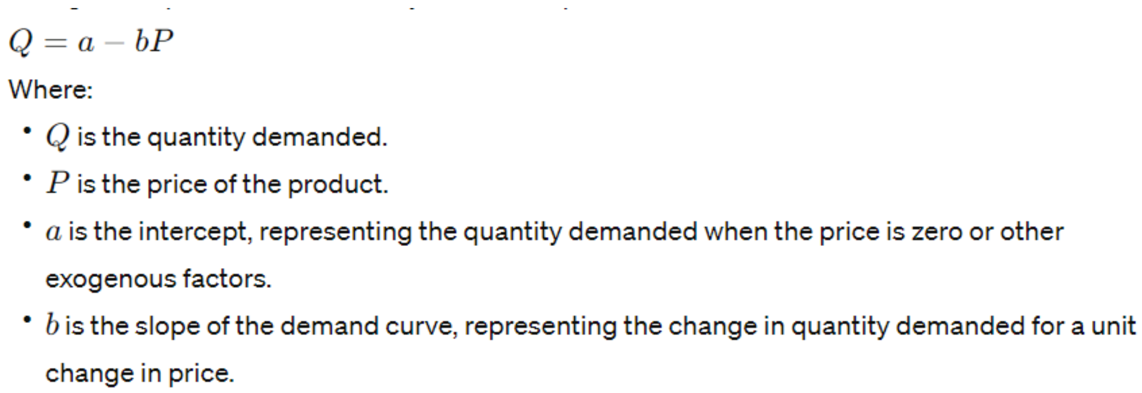


Period	Price	Demand
1	16	500
2	18	400
3	20	350
4	22	450
5	24	470
6	26	430
7	28	380
8	30	300
9	32	250
10	34	200
11	36	150
12	38	100

A price response function, also known as a demand function or demand curve, is a mathematical representation of the relationship between the price of a product and the quantity demanded by consumers. It describes how changes in price lead to changes in the quantity of a product that consumers are willing to purchase, all else being equal.

The two commonly used models to explain price response: Linear Price Response Model and Logit Model.

1. **Linear Price Response Model:** This model assumes that the change in demand for a product is directly proportional to the change in its price. Mathematically, it can be represented as:



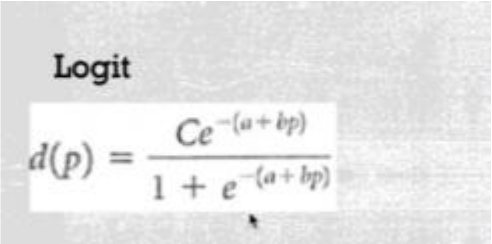
$$Q = a - bP$$

Where:

- Q is the quantity demanded.
- P is the price of the product.
- a is the intercept, representing the quantity demanded when the price is zero or other exogenous factors.
- b is the slope of the demand curve, representing the change in quantity demanded for a unit change in price.

In this model, the coefficient *b* is negative, indicating an inverse relationship between price and quantity demanded. The larger the absolute value of *b*, the more elastic the demand.

2. **Logit Model:** The Logit Model is commonly used in economics and marketing to analyze discrete choice behaviour, particularly in situations where consumers choose among a finite set of alternatives (like whether to purchase a product or not). It models the probability that an individual will choose a particular alternative based on a set of explanatory variables, including price.



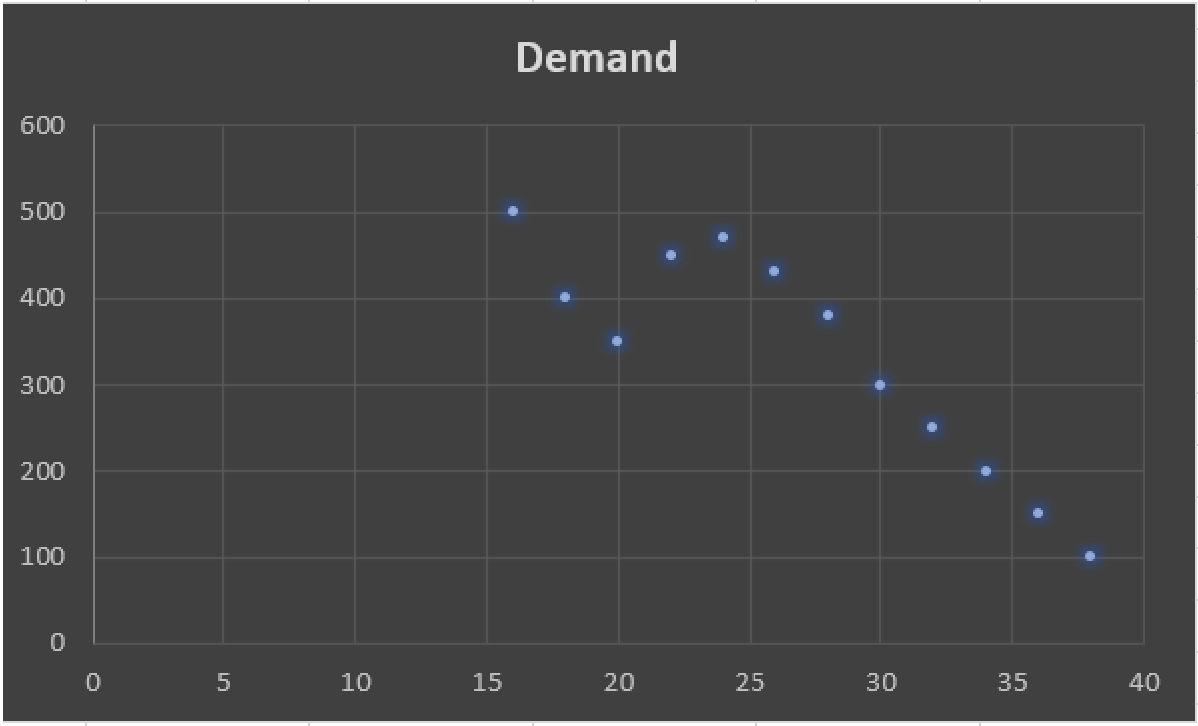
Logit

$$d(p) = \frac{Ce^{-(a+bp)}}{1 + e^{-(a+bp)}}$$

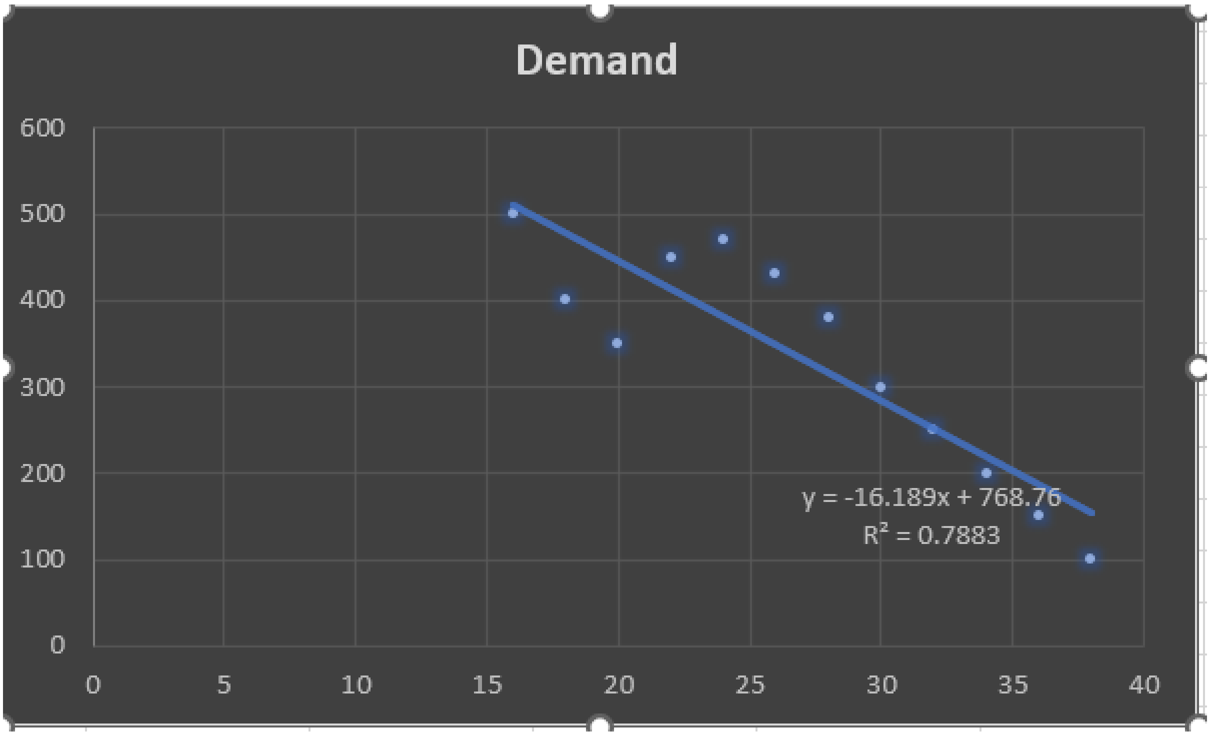
Here C= total population size, p is the price and a and b are the parameters.

Both of these models provide valuable insights into price response, albeit from different perspectives. The Linear Price Response Model is more straightforward and focuses on the direct relationship between price and quantity demanded, while the Logit Model considers pricing within the broader context of consumer choice behaviour.

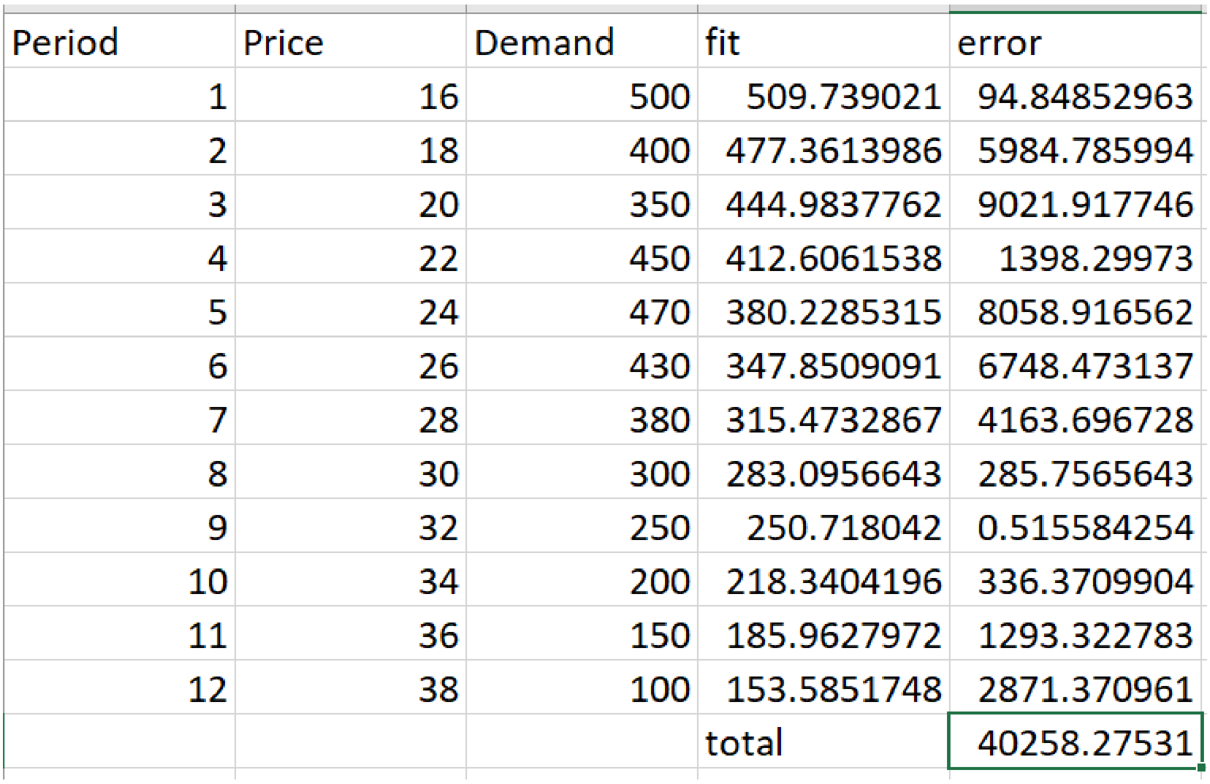
First we will visualize the demand vs price using a scatter chart.



We can infer that as the price decreases, the demand also decreases. But that decrease is not direct decrease. So may be some non-linearity or randomness happening. We will figure out using the price response function. We just added the trend line and the option to display the equation and r square value.

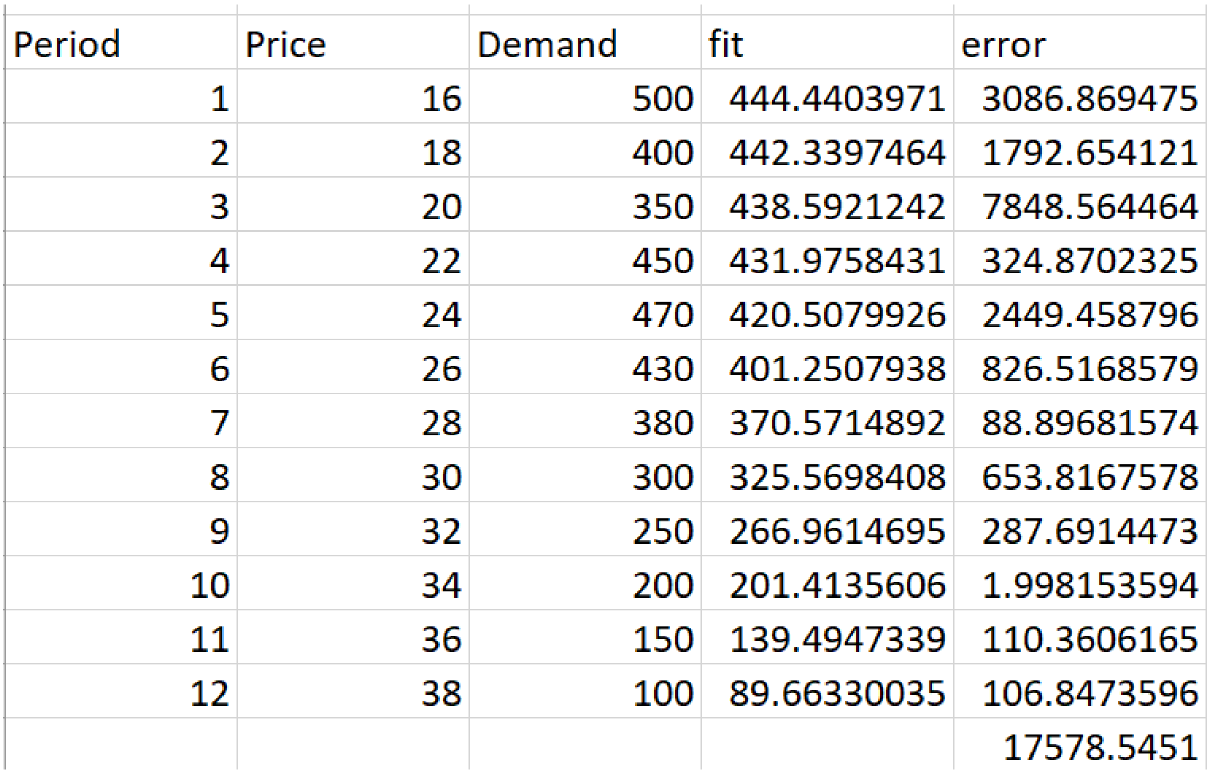


Next we calculated fit= price* coefficient of price(16.189)+ intercept(768.76). Error is the difference between the squares of demand and fit. We can use the values of intercept and co-efficient either from the chart or using Linest function. Now the total sum of squared error for the linear response function =40258.27531



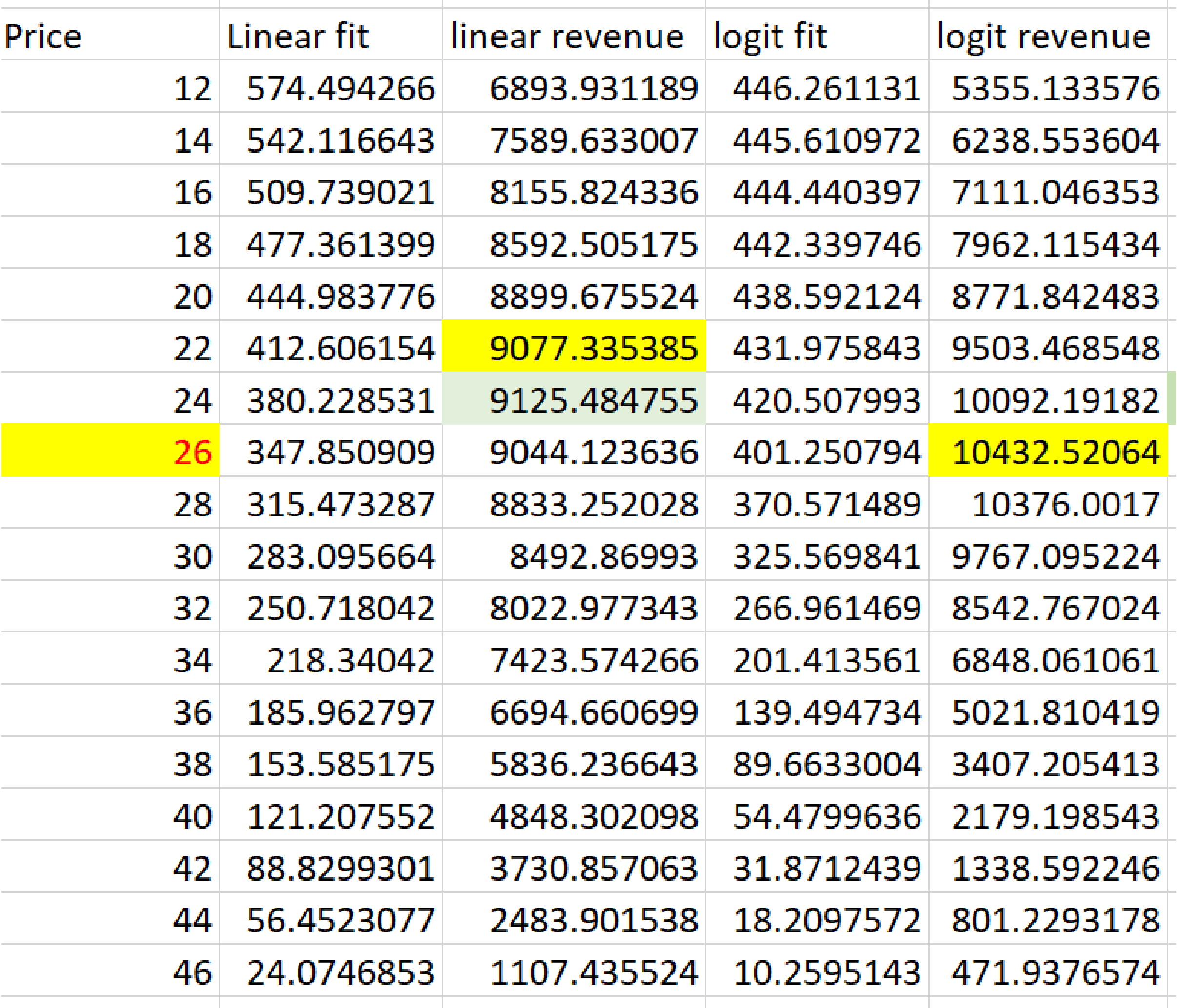
Period	Price	Demand	fit	error
1	16	500	509.739021	94.84852963
2	18	400	477.3613986	5984.785994
3	20	350	444.9837762	9021.917746
4	22	450	412.6061538	1398.29973
5	24	470	380.2285315	8058.916562
6	26	430	347.8509091	6748.473137
7	28	380	315.4732867	4163.696728
8	30	300	283.0956643	285.7565643
9	32	250	250.718042	0.515584254
10	34	200	218.3404196	336.3709904
11	36	150	185.9627972	1293.322783
12	38	100	153.5851748	2871.370961
			total	40258.27531

Next is using the logit function. For that we use solver tool. We relay on solver to estimate and optimize the parameters to minimize the error. First we calculated the fit using some random values for C, b and a. tHhen calculated the error= squared difference of demand-fit. So we get a total error. Now go to solver tool and say we want to minimize this total error by changing parameters.



Period	Price	Demand	fit	error
1	16	500	444.4403971	3086.869475
2	18	400	442.3397464	1792.654121
3	20	350	438.5921242	7848.564464
4	22	450	431.9758431	324.8702325
5	24	470	420.5079926	2449.458796
6	26	430	401.2507938	826.5168579
7	28	380	370.5714892	88.89681574
8	30	300	325.5698408	653.8167578
9	32	250	266.9614695	287.6914473
10	34	200	201.4135606	1.998153594
11	36	150	139.4947339	110.3606165
12	38	100	89.66330035	106.8473596
				17578.5451

So the logit model is a better fit than linear model due to less error. Next we simulate the price for linear and logistic function. Prices range from 12 to 46. Use both the functions and calculated the revenue for both. Revenue will be product of price* estimated demand.

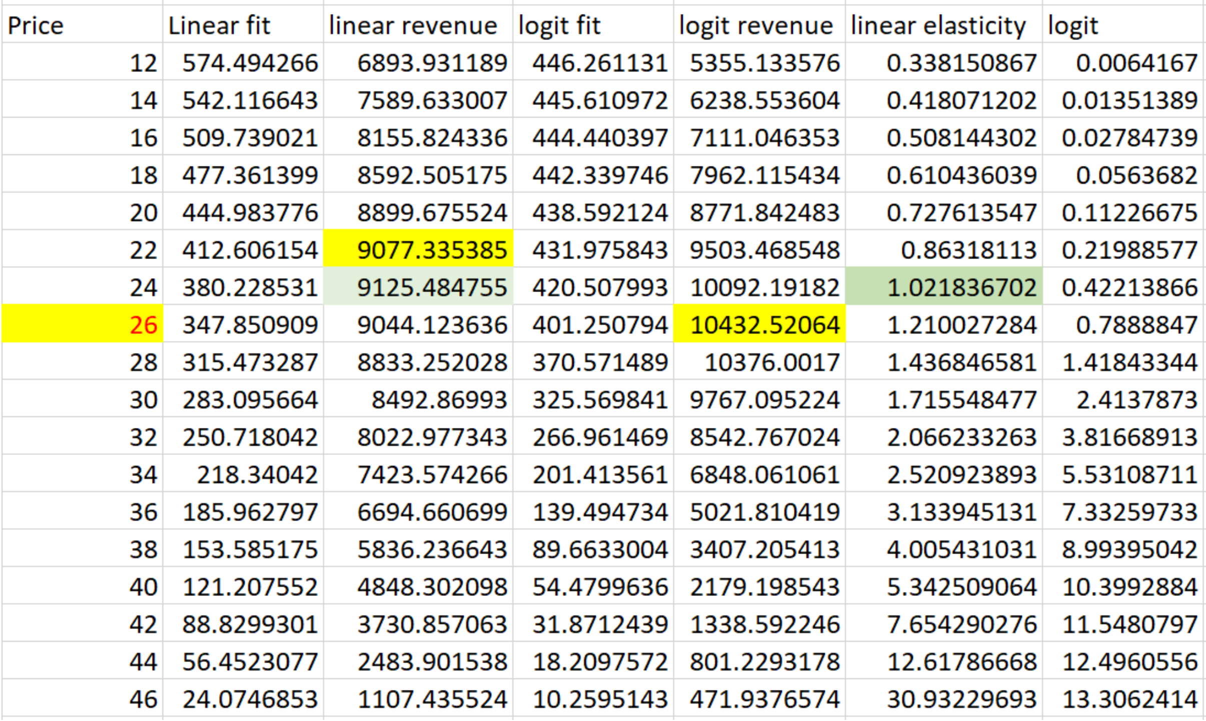


Price	Linear fit	linear revenue	logit fit	logit revenue
12	574.494266	6893.931189	446.261131	5355.133576
14	542.116643	7589.633007	445.610972	6238.553604
16	509.739021	8155.824336	444.440397	7111.046353
18	477.361399	8592.505175	442.339746	7962.115434
20	444.983776	8899.675524	438.592124	8771.842483
22	412.606154	9077.335385	431.975843	9503.468548
24	380.228531	9125.484755	420.507993	10092.19182
26	347.850909	9044.123636	401.250794	10432.52064
28	315.473287	8833.252028	370.571489	10376.0017
30	283.095664	8492.86993	325.569841	9767.095224
32	250.718042	8022.977343	266.961469	8542.767024
34	218.34042	7423.574266	201.413561	6848.061061
36	185.962797	6694.660699	139.494734	5021.810419
38	153.585175	5836.236643	89.6633004	3407.205413
40	121.207552	4848.302098	54.4799636	2179.198543
42	88.8299301	3730.857063	31.8712439	1338.592246
44	56.4523077	2483.901538	18.2097572	801.2293178
46	24.0746853	1107.435524	10.2595143	471.9376574

The highest fit of linear revenue(9077.33) is at price 22 and the highest fit of logit revenue(10432.52) is at price 26. So, the range of optimum price will be between the ranges of 22-26. But since our logit model fit better, we choose the price 26 as the optimized price.

Next, we calculated the elasticity of the products. Elasticity measures the sensitivity of demand in relation to the price. It is an indicator that tells us whether the product at this price is sensitive to change in price or not. If the product is sensitive to change in price it is called elastic then we need to decrease the price to get optimum revenue. Optimum revenue is the point where elasticity reaches 1 or the product is unit elastic i.e. the % change in demand is equal to % change in price. If elasticity <1, then increase the price to maximize the revenue till elasticity reaches 1.

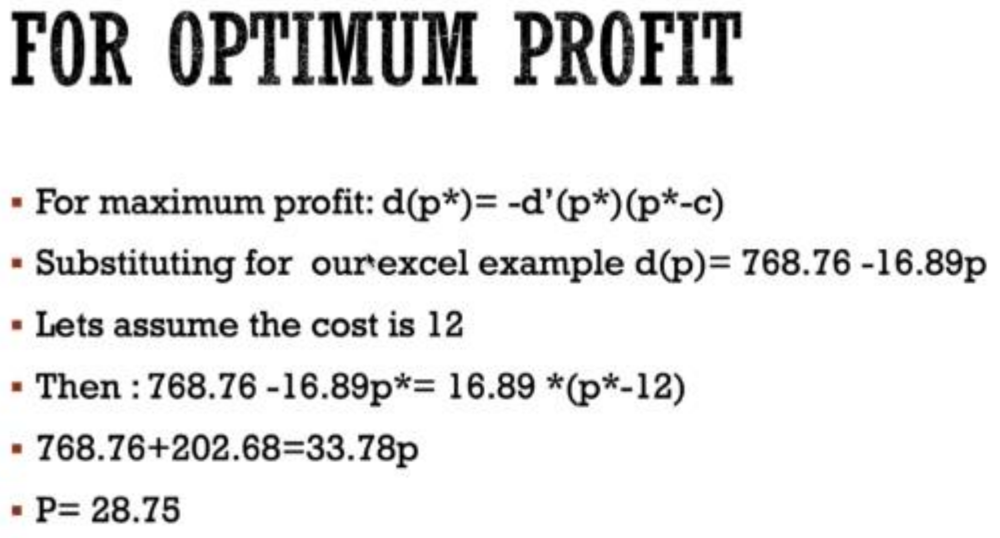
Linear elasticity = /-d'(p)*p/d(p) where d(p) is the price response function (linear fit value). For linear the derivative of response function = slope. So Linear elasticity = (slope*price)/ linear fit. For logit the elasticity formula= bp/(1+exp(-a+bp).



Price	Linear fit	linear revenue	logit fit	logit revenue	linear elasticity	logit
12	574.494266	6893.931189	446.261131	5355.133576	0.338150867	0.0064167
14	542.116643	7589.633007	445.610972	6238.553604	0.418071202	0.01351389
16	509.739021	8155.824336	444.440397	7111.046353	0.508144302	0.02784739
18	477.361399	8592.505175	442.339746	7962.115434	0.610436039	0.0563682
20	444.983776	8899.675524	438.592124	8771.842483	0.727613547	0.11226675
22	412.606154	9077.335385	431.975843	9503.468548	0.86318113	0.21988577
24	380.228531	9125.484755	420.507993	10092.19182	1.021836702	0.42213866
26	347.850909	9044.123636	401.250794	10432.52064	1.210027284	0.7888847
28	315.473287	8833.252028	370.571489	10376.0017	1.436846581	1.41843344
30	283.095664	8492.86993	325.569841	9767.095224	1.715548477	2.4137873
32	250.718042	8022.977343	266.961469	8542.767024	2.066233263	3.81668913
34	218.34042	7423.574266	201.413561	6848.061061	2.520923893	5.53108711
36	185.962797	6694.660699	139.494734	5021.810419	3.133945131	7.33259733
38	153.585175	5836.236643	89.6633004	3407.205413	4.005431031	8.99395042
40	121.207552	4848.302098	54.4799636	2179.198543	5.342509064	10.3992884
42	88.8299301	3730.857063	31.8712439	1338.592246	7.654290276	11.5480797
44	56.4523077	2483.901538	18.2097572	801.2293178	12.61786668	12.4960556
46	24.0746853	1107.435524	10.2595143	471.9376574	30.93229693	13.3062414

In case of linear, at the price=1, the product is inelastic. At price between 22 and 24 the elasticity reaches 1. So, this is the point of maximum revenue. For logit model elasticity reaches 1 between 26-28 so this is the point of optimum or maximum revenue.

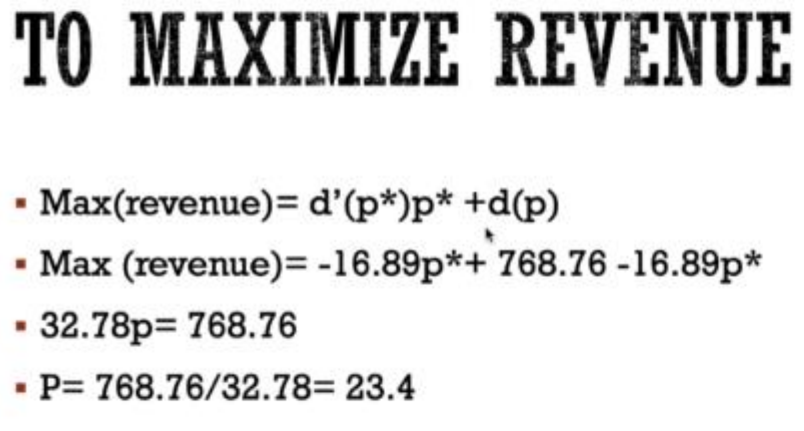
Now if the cost given, we can also calculate the point of maximum profit too.



FOR OPTIMUM PROFIT

- For maximum profit: $d(p^*)= -d'(p^*)(p^*-c)$
- Substituting for our excel example $d(p)= 768.76 -16.89p$
- Lets assume the cost is 12
- Then : $768.76 -16.89p^*= 16.89 *(p^*-12)$
- $768.76+202.68=33.78p$
- $P= 28.75$

For the optimum profit, the client needs to set the price =28.75



TO MAXIMIZE REVENUE

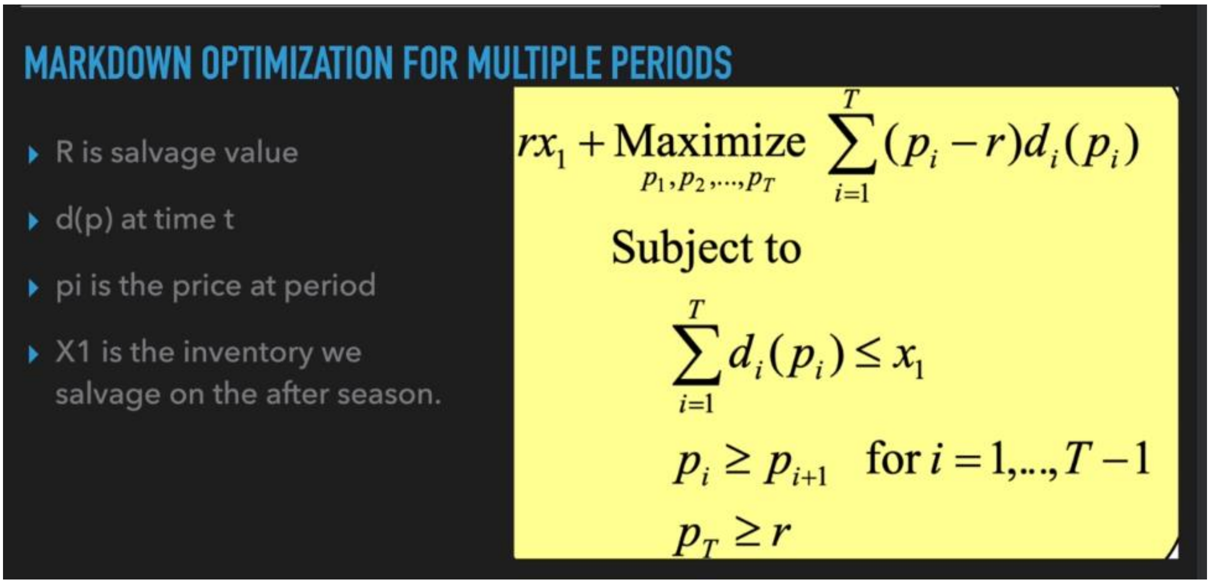
- $\text{Max(revenue)}= d'(p^*)p^* +d(p)$
- $\text{Max (revenue)}= -16.89p^*+ 768.76 -16.89p^*$
- $32.78p= 768.76$
- $P= 768.76/32.78= 23.4$

For the maximum revenue, the client needs to set the price =23.4

Task-2: Mark down optimization

Markdown optimization in retail analytics refers to the strategic adjustment of prices to maximize sales and profitability. Markdowns are temporary reductions in the selling price of products to stimulate sales, clear inventory, and maintain a healthy retail operation.

Salvage value is the price we set for when we write off the products and we want to get rid of that. When the season finished we want to get rid of our products so we set a minimal salvage value.



MARKDOWN OPTIMIZATION FOR MULTIPLE PERIODS

- R is salvage value
- d(p) at time t
- pi is the price at period
- X1 is the inventory we salvage on the after season.

$$rx_1 + \underset{p_1, p_2, \dots, p_T}{\text{Maximize}} \sum_{i=1}^{T} (p_i - r) d_i(p_i)$$

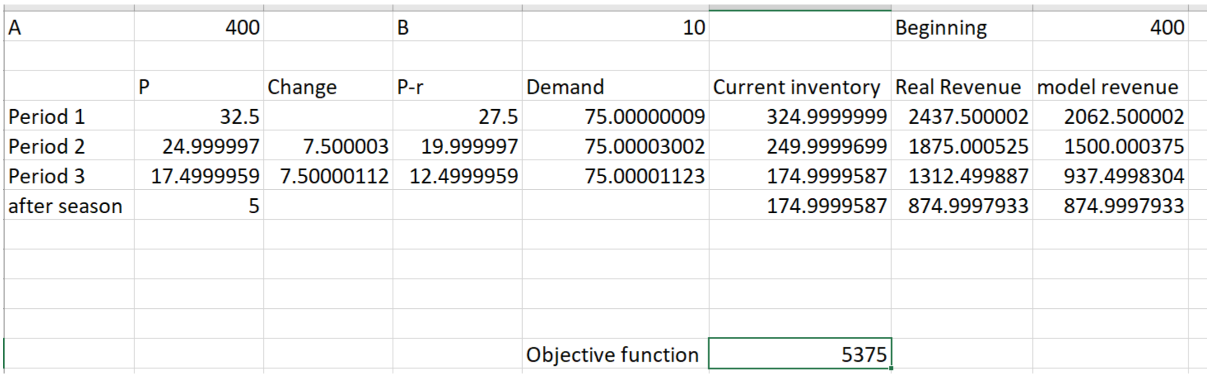
Subject to

$$\sum_{i=1}^{T} d_i(p_i) \le x_1$$

$$p_i \ge p_{i+1} \quad \text{for } i = 1, \dots, T-1$$

$$p_T \ge r$$

set salvage value=r=5, A=400,B=10 and Beginning inventory=400 Calculated the Change =pi+1-pi, p-r value and the Demand function=max(0, A-BP). Now calculated the current inventory which is the difference between the beginning inventory and the demand calculated for that period. Next calculated the real revenue= product of price and demand. The model revenue is the product of demand and p-r. The objective function will be sum of all the model revenue. Now we use solver tool to maximise this objective function by changing the price values and add the subject to conditions as constraints of the solver tool.

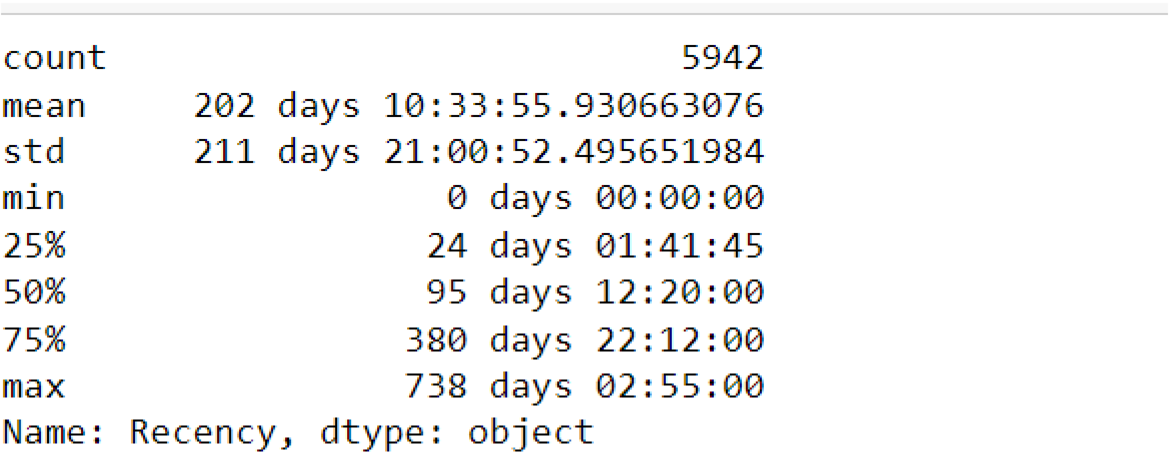


A	400		B	10		Beginning	400
	P	Change	P-r	Demand	Current inventory	Real Revenue	model revenue
Period 1	32.5		27.5	75.00000009	324.9999999	2437.500002	2062.500002
Period 2	24.999997	7.500003	19.999997	75.00003002	249.9999699	1875.000525	1500.000375
Period 3	17.4999959	7.50000112	12.4999959	75.00001123	174.9999587	1312.499887	937.4998304
after season	5				174.9999587	874.9997933	874.9997933
				Objective function	5375		

So, the price at period 1 will be 32.5 and expected to sell 75 and the real revenue will be 2437.50. At period 2 the company should decrease the price to 24.99 and further decrease to 17.5 at period 3. After the season is finished the company should sell the remaining inventory 175 at 5.

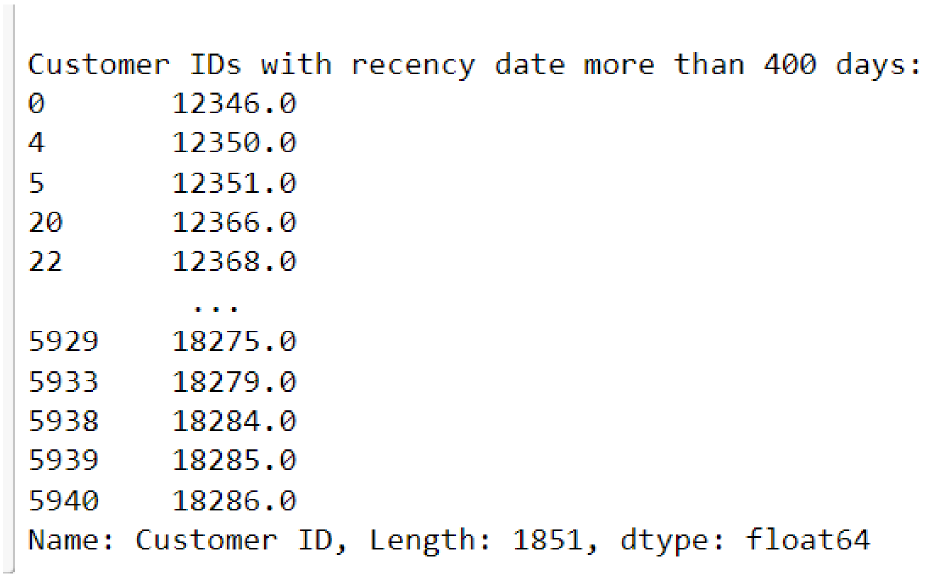
Task-3:

We have given a retail dataset given. We need to analyze the dataset and create some insights from it. We will be using pandas library for the analysis. We have carried put some basic cleaning operations like checking nulls, dropping duplicates, converting the columns to suitable datatypes, grouping by certain cols and calculating some aggregation metrics, creating a pivot table to provide a summary etc. Next is we will be working with dates. After some initial date manipulation tasks, we calculated the recency of the purchases of the customer. Recency means since how long did customer last purchased from us. It is calculated as the difference between the maximum date of the dataset and the last date purchased by each customer using group by function.



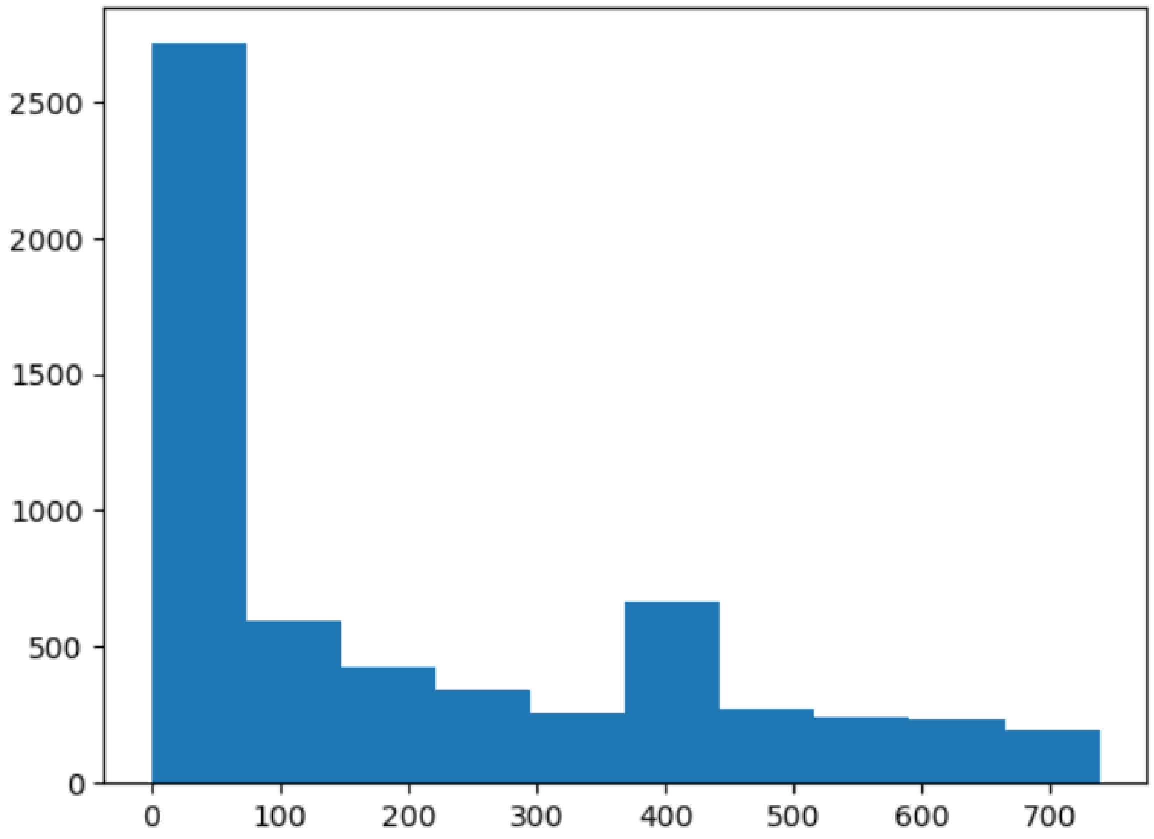
```
count                                 5942
mean      202 days 10:33:55.930663076
std       211 days 21:00:52.495651984
min                   0 days 00:00:00
25%                  24 days 01:41:45
50%                  95 days 12:20:00
75%                 380 days 22:12:00
max                 738 days 02:55:00
Name: Recency, dtype: object
```

We have lost the customers who has recency dates more than 400 days. Just we displayed those customer ids who stopped buying or lost or not purchased more than 400 days.



```
Customer IDs with recency date more than 400 days:
0       12346.0
4       12350.0
5       12351.0
20      12366.0
22      12368.0
          ...
5929    18275.0
5933    18279.0
5938    18284.0
5939    18285.0
5940    18286.0
Name: Customer ID, Length: 1851, dtype: float64
```

We have also created a histogram for recency days. From the histogram we infer that most of data have recent purchase history. Also it is right skewed so some customers have not purchased for long too.



Insights from Recency Analysis:

Summary Statistics:

- The analysis of recency among customers reveals significant variations in their purchasing behavior. On average, customers made purchases approximately 202 days ago, with a standard deviation of approximately 211 days. The minimum recency observed is 0 days, indicating that some customers made purchases very recently. On the other hand, the maximum recency is 738 days, reflecting a considerable gap between the most recent purchase and the present date.

Customer Behavior:

- The customer with the shortest recency (Customer ID: 12680.0) made their most recent purchase on December 9, 2011, resulting in a recency of 0 days, suggesting very recent activity.
- In contrast, the customer with the longest recency (Customer ID: 12636.0) made their most recent purchase on December 1, 2009, resulting in a recency of 738 days, indicating a significant gap since their last purchase.

Median Recency:

- The median recency across all customers is approximately 95 days, indicating that half of the customers made their most recent purchase within this timeframe.

Customer Segmentation:

- A substantial number of customers, totaling 2971 individuals, have recency greater than the median. This finding suggests that approximately 50.00% of customers have not made a purchase within the last 95 days, highlighting the importance of re-engagement strategies for this segment.

Implications:

- Understanding the distribution of recency enables businesses to segment their customer base effectively and tailor marketing strategies accordingly. Customers with shorter recency may respond well to promotions or reminders to encourage repeat purchases, while those with longer recency may require targeted efforts to re-engage them with the brand.

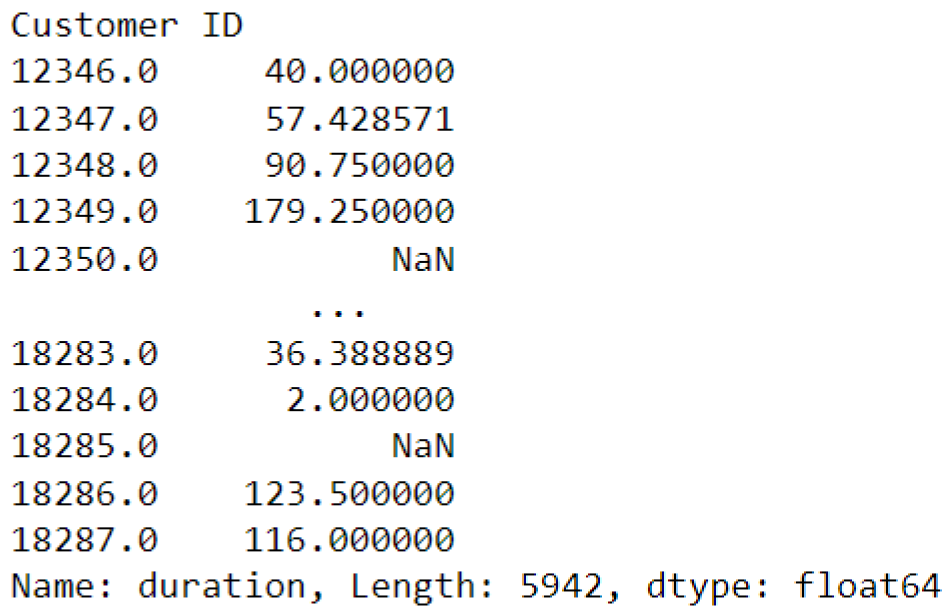
Conclusion:

- Analyzing recency provides valuable insights into customer engagement and purchasing patterns. By leveraging these insights, businesses can devise strategies to retain existing customers, re-engage inactive ones, and ultimately drive revenue growth and customer satisfaction.

Next is we will be modelling the inter arrival time. It means how frequently every customer arrives to buy something.

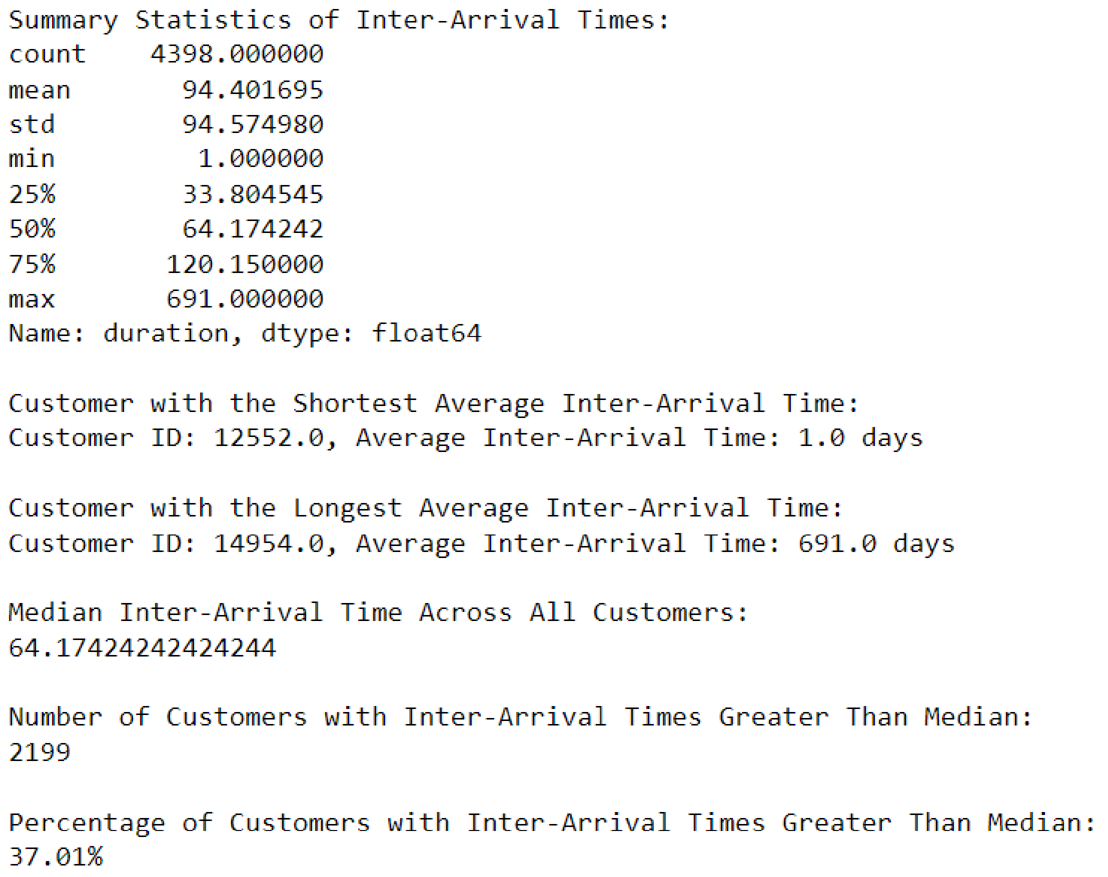
Customer Inter-Arrival Time Analysis Report:

In our analysis, we focused on understanding the time gaps between successive purchases made by each customer. This information is crucial for discerning customer behaviour patterns and establishing effective marketing strategies. We began by extracting unique customer IDs from our retail dataset. Afterward, we grouped the data by customer ID and purchase date, counting the occurrences to determine the frequency of purchases for each customer. Using the grouped data, we computed the duration between consecutive purchase dates for every customer. This allowed us to understand how much time elapsed, on average, between each purchase made by individual customers.



```
Customer ID
12346.0      40.000000
12347.0      57.428571
12348.0      90.750000
12349.0     179.250000
12350.0            NaN
               ...
18283.0      36.388889
18284.0       2.000000
18285.0            NaN
18286.0     123.500000
18287.0     116.000000
Name: duration, Length: 5942, dtype: float64
```

We have done some further analysis to extract some insights such as such as summary statistics of inter-arrival times, customers with the shortest and longest average inter-arrival times, median inter-arrival time across all customers, and the number/percentage of customers with inter-arrival times greater than the median.



```
Summary Statistics of Inter-Arrival Times:
count    4398.000000
mean       94.401695
std        94.574980
min         1.000000
25%        33.804545
50%        64.174242
75%       120.150000
max       691.000000
Name: duration, dtype: float64

Customer with the Shortest Average Inter-Arrival Time:
Customer ID: 12552.0, Average Inter-Arrival Time: 1.0 days

Customer with the Longest Average Inter-Arrival Time:
Customer ID: 14954.0, Average Inter-Arrival Time: 691.0 days

Median Inter-Arrival Time Across All Customers:
64.17424242424244

Number of Customers with Inter-Arrival Times Greater Than Median:
2199

Percentage of Customers with Inter-Arrival Times Greater Than Median:
37.01%
```

Insights from Inter-Arrival Time Analysis:

Summary Statistics:

- The analysis of inter-arrival times across all customers reveals valuable insights. On average, customers make purchases approximately every 94.4 days, with a standard deviation of 94.57 days, indicating a considerable variation in purchasing frequencies.
- The shortest inter-arrival time observed is just 1 day, indicating a customer who makes purchases almost daily, while the longest inter-arrival time is an astonishing 691 days, representing a significant gap between purchases for another customer.

Customer Behaviour:

- The customer with the shortest average inter-arrival time (Customer ID: 12552.0) demonstrates remarkably frequent purchasing behaviour, with an average time of just 1 day between purchases.
- Conversely, the customer with the longest average inter-arrival time (Customer ID: 14954.0) exhibits a starkly different behaviour, with an average time of 691 days between purchases, suggesting infrequent engagement with the retail offerings.

Median Inter-Arrival Time:

- The median inter-arrival time across all customers is approximately 64.17 days. This metric provides a central tendency measure, indicating that half of the customers have inter-arrival times shorter than this value, while the other half have longer inter-arrival times.

Customer Segmentation:

- A substantial portion of customers, totaling 2199 individuals, have inter-arrival times greater than the median. This finding suggests a significant segment of customers with less frequent purchasing behavior, representing approximately 37.01% of the total customer base.

Implications:

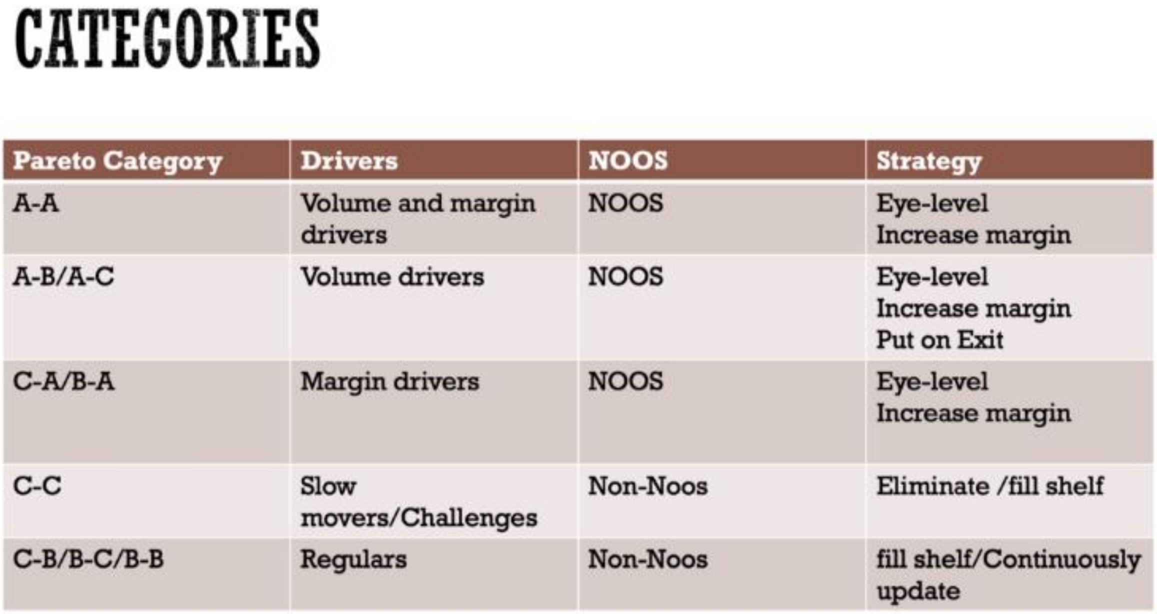
- Understanding the distribution of inter-arrival times enables businesses to segment their customer base effectively and tailor marketing strategies accordingly. Customers with short inter-arrival times may benefit from loyalty programs to incentivize frequent purchases, while those with longer inter-arrival times may require targeted campaigns to re-engage and retain their interest.

Conclusion:

- Analyzing inter-arrival times provides valuable insights into customer behavior and preferences. By leveraging these insights, businesses can optimize their marketing efforts, enhance customer satisfaction, and drive sustainable growth in the competitive retail landscape.

Task-4: Product placement inside the store

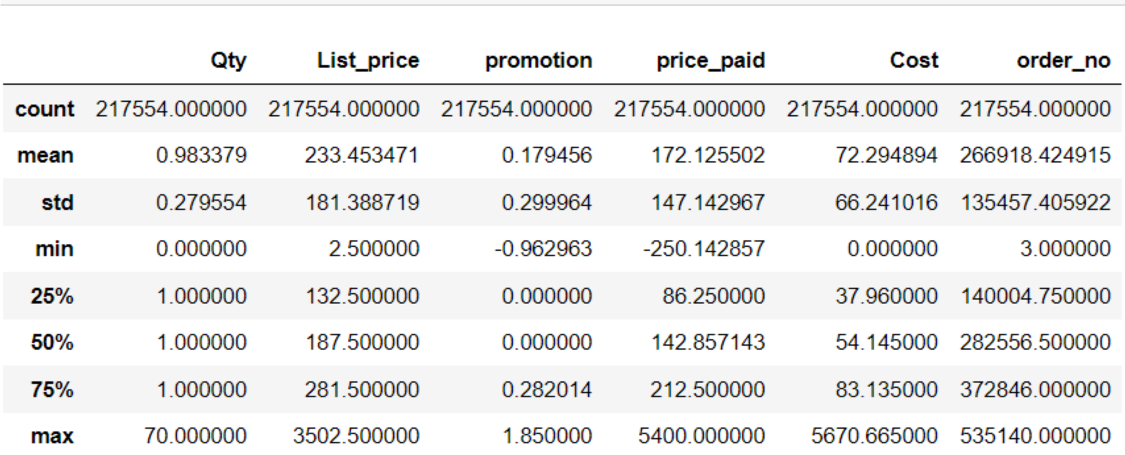
Our task is to find out those critical items that contribute most to the revenue so that the company can position it accordingly in the valuable shelf of the store. Based on their volume and margin we classify products to different categories and appropriate product placement strategies are implemented based on their categories. We have given a apparel dataset for the analytics.



CATEGORIES

Pareto Category	Drivers	NOOS	Strategy
A-A	Volume and margin drivers	NOOS	Eye-level Increase margin
A-B/A-C	Volume drivers	NOOS	Eye-level Increase margin Put on Exit
C-A/B-A	Margin drivers	NOOS	Eye-level Increase margin
C-C	Slow movers/Challenges	Non-Noos	Eliminate /fill shelf
C-B/B-C/B-B	Regulars	Non-Noos	fill shelf/Continuously update

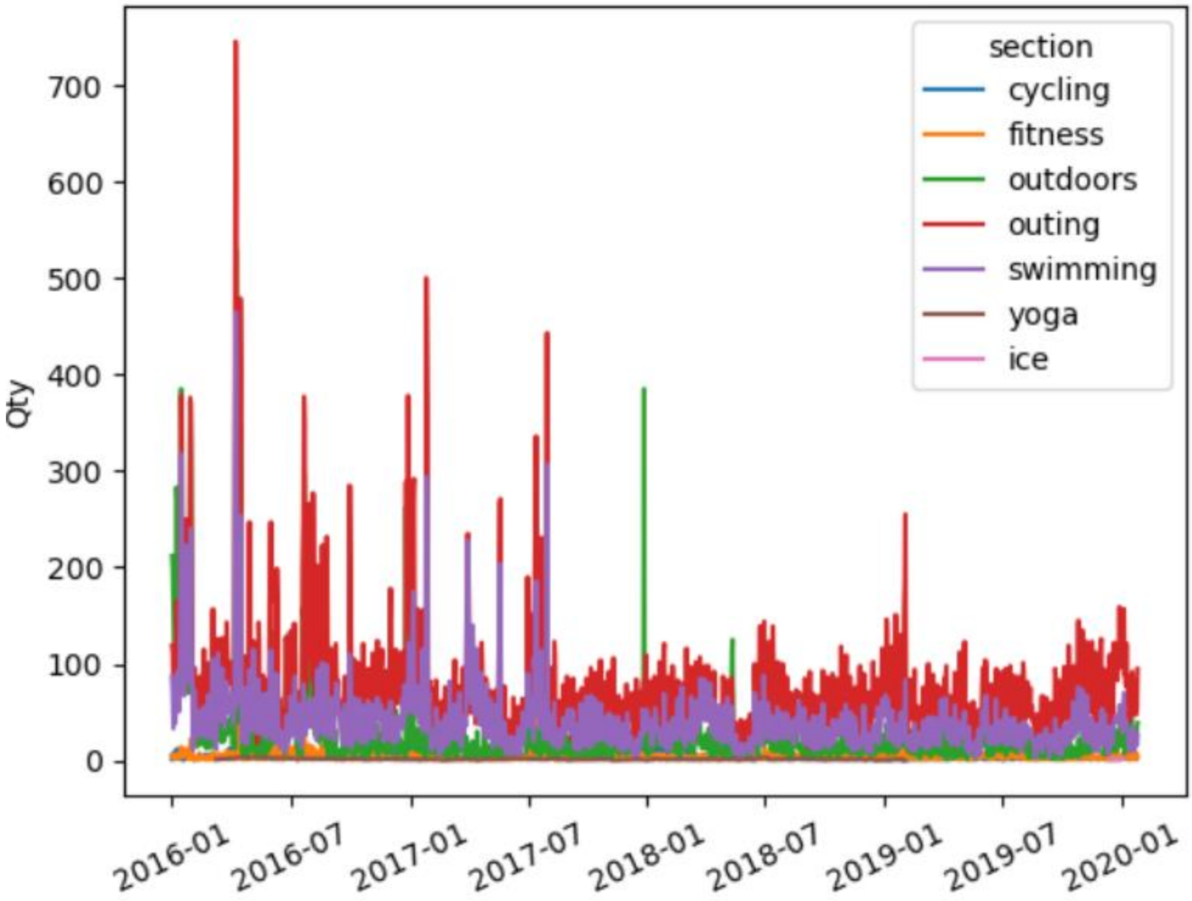
We will be exploring the dataset first using the basic methods.



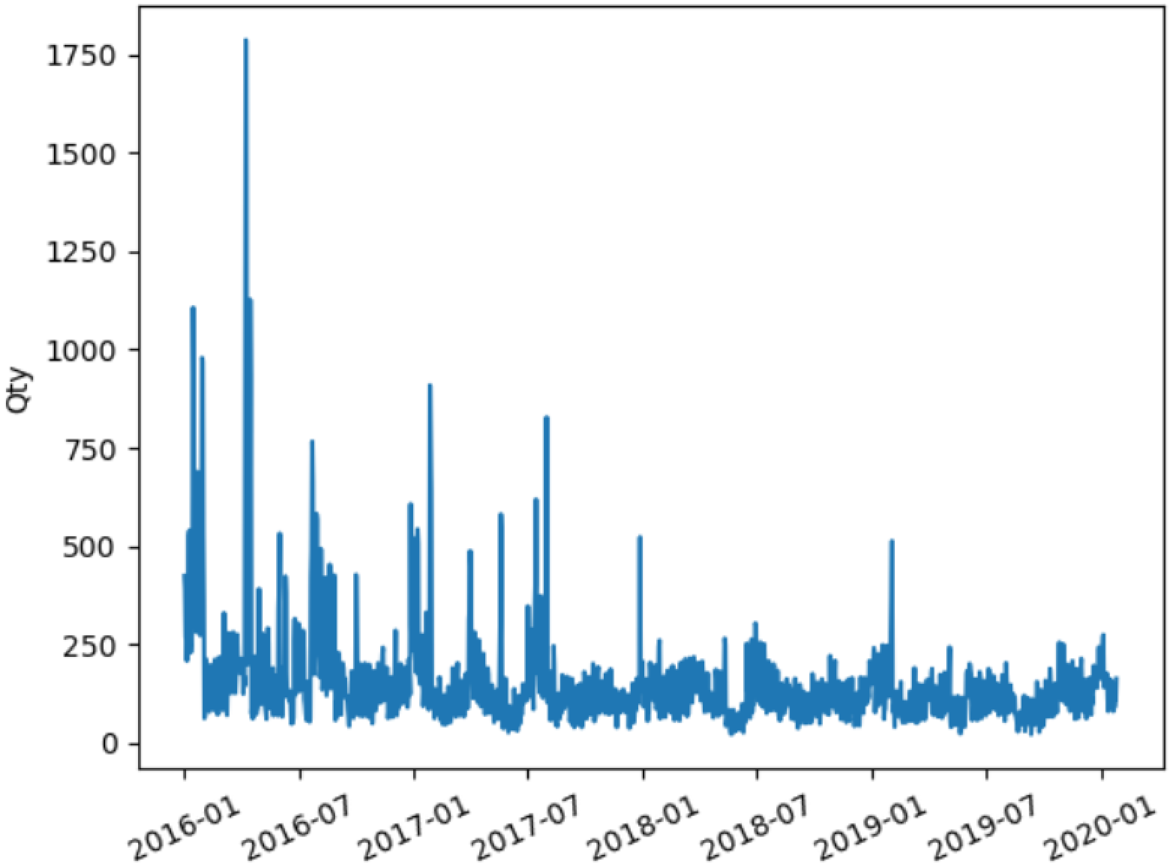
	Qty	List_price	promotion	price_paid	Cost	order_no
count	217554.000000	217554.000000	217554.000000	217554.000000	217554.000000	217554.000000
mean	0.983379	233.453471	0.179456	172.125502	72.294894	266918.424915
std	0.279554	181.388719	0.299964	147.142967	66.241016	135457.405922
min	0.000000	2.500000	-0.962963	-250.142857	0.000000	3.000000
25%	1.000000	132.500000	0.000000	86.250000	37.960000	140004.750000
50%	1.000000	187.500000	0.000000	142.857143	54.145000	282556.500000
75%	1.000000	281.500000	0.282014	212.500000	83.135000	372846.000000
max	70.000000	3502.500000	1.850000	5400.000000	5670.665000	535140.000000

We can infer that the minimum value of list price is 0 so some products are bought free or as bonus and the maximum list price is 3502. Also in some cases cost is higher than price may be the supplier want to get rid of the stock. Average promotion applied is 17% and a negative promotion indicates an increase in price. Also promotion is applied only for the 50% of products.

Then we have calculated the total sales of apparel for each section on each date and also the total sales of apparel on each date alone. We have plotted the 2 using line chart.



From this line chart we infer that swimming is dominated followed by outdoors in sales by section.



From this chart we infer that there is an up and down trend in averages sales.

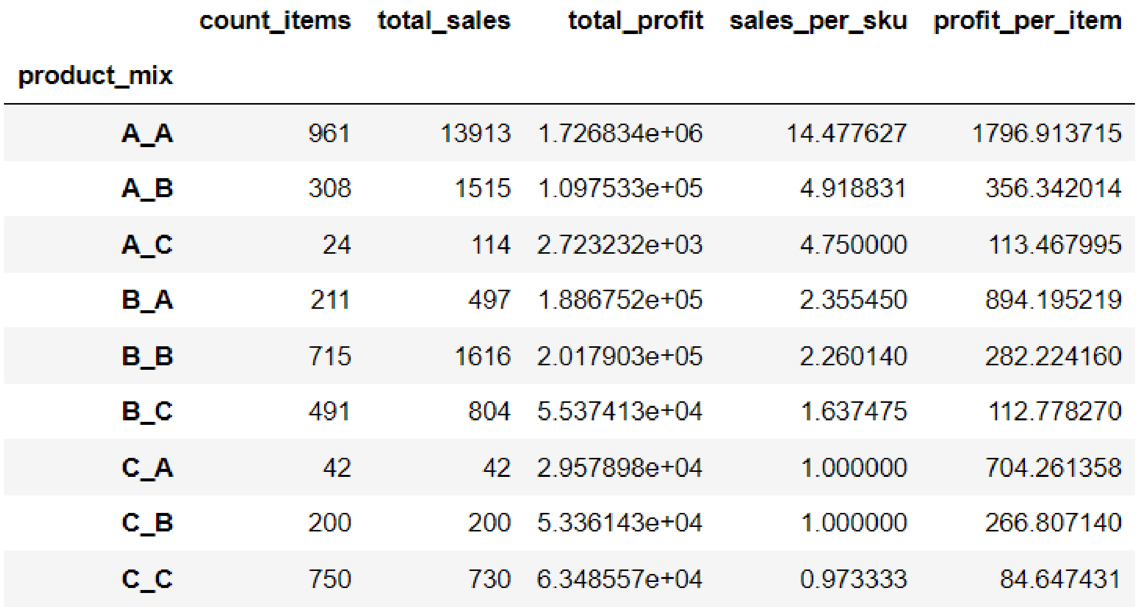
Next, we calculated some metrics like revenue which is equal to product of quantity and list price after discounting the promotion, the total cost which is equal to the product of quantity and cost and the total profit is difference between the revenue and the total cost. Next is identifying the volume drivers so that we can position the volume drivers at exit doors. We focus on the volume drivers of last quarter. We created a new column named "item_section" by combining information from existing columns: "description," "subfamily," "size," and "color." This new column likely provides a comprehensive description of each item, incorporating various attributes such as the item's name, category, size, and color. Then we grouped the data based on the combined item sections created earlier and calculated two aggregate metrics:

- **Total Profit**: It sums up the profit generated by each item section.
- **Total Sales Quantity**: It sums up the quantity of items sold for each item section.

- Next, we have performed the ABC analysis using inventorize package in python. The **productmix** function from the **inventorize** library is used to classify items into different product mixes based on their total sales and total profit. It likely assigns each item a classification based on predefined criteria, such as high sales and high profit, low sales and high profit, etc. Then Additional metrics are computed:

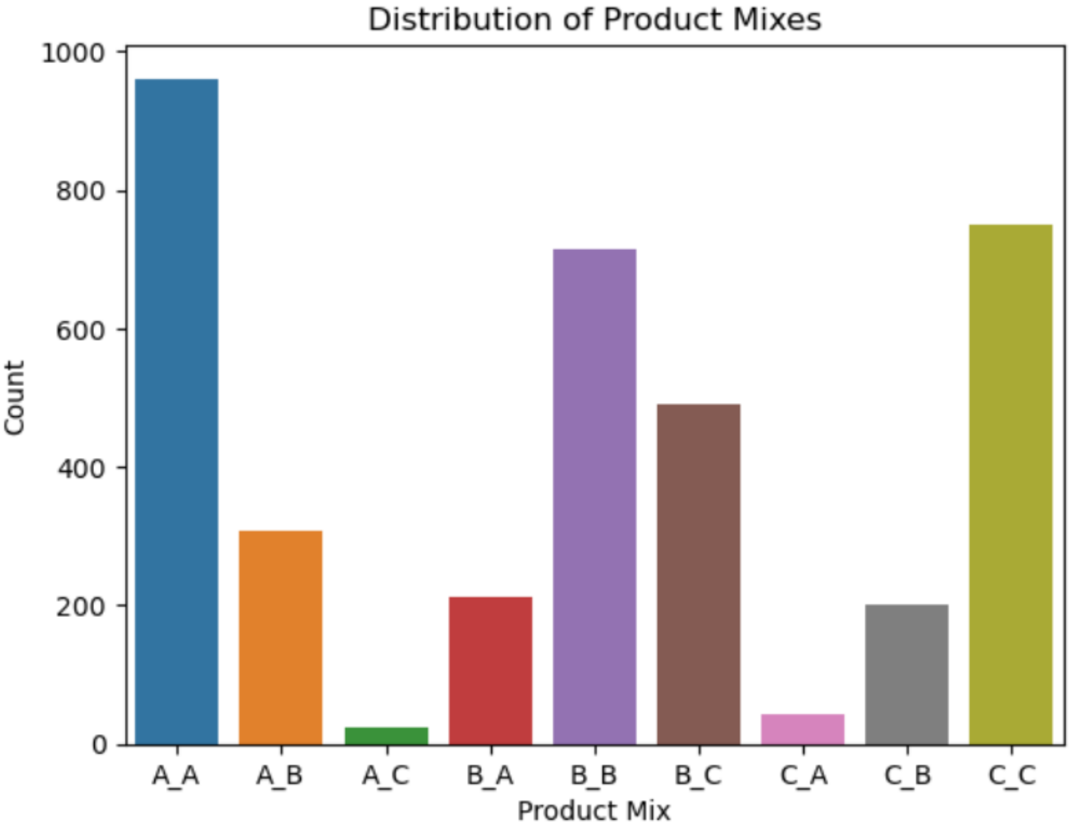
- **sales_per_SKU**: Total sales divided by the count of SKUs for each product mix.
- **profit_per_item**: Total profit divided by the count of SKUs for each product mix.

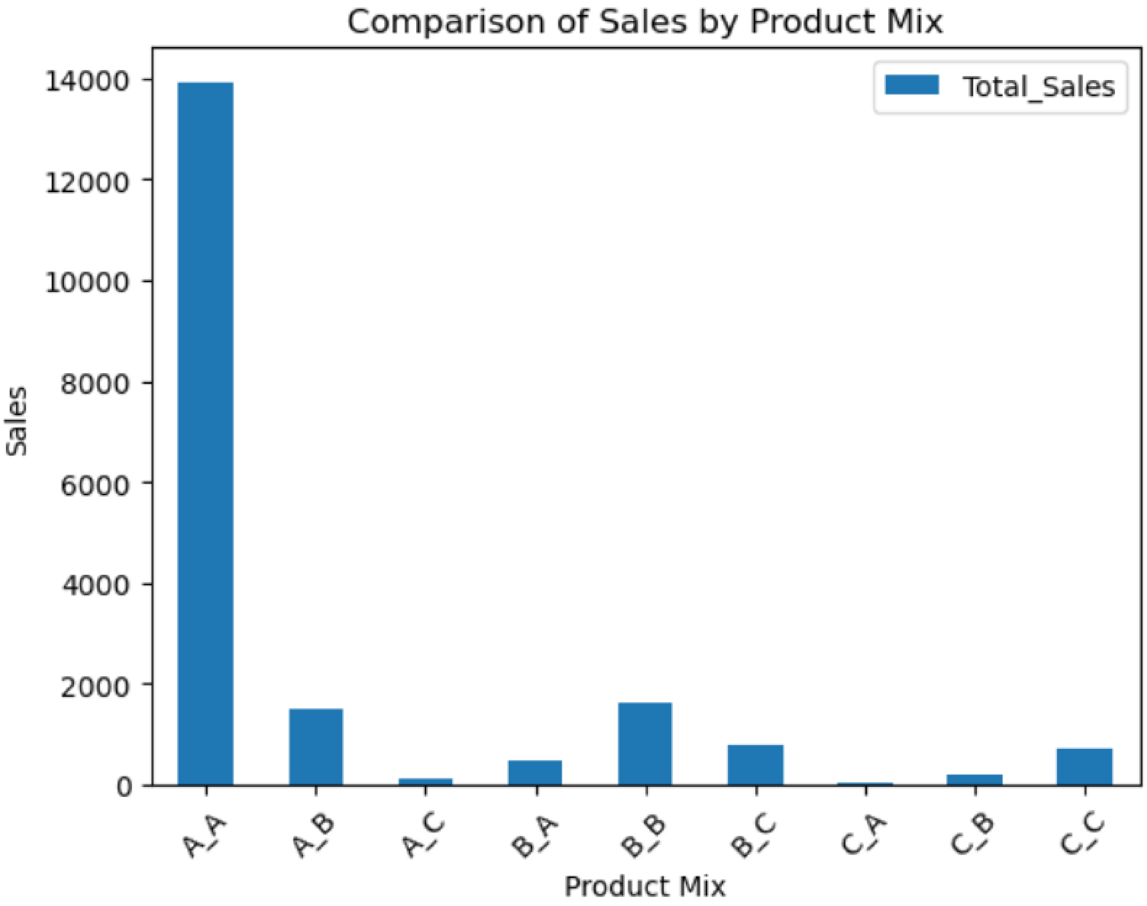
These metrics provide insights into the average sales and profit per SKU within each product mix. So we analysed the product mix by classifying items, grouping them based on their classifications, and computing various metrics to understand the performance of different product mixes in terms of sales and profit.



	count_items	total_sales	total_profit	sales_per_sku	profit_per_item
product_mix					
A_A	961	13913	1.726834e+06	14.477627	1796.913715
A_B	308	1515	1.097533e+05	4.918831	356.342014
A_C	24	114	2.723232e+03	4.750000	113.467995
B_A	211	497	1.886752e+05	2.355450	894.195219
B_B	715	1616	2.017903e+05	2.260140	282.224160
B_C	491	804	5.537413e+04	1.637475	112.778270
C_A	42	42	2.957898e+04	1.000000	704.261358
C_B	200	200	5.336143e+04	1.000000	266.807140
C_C	750	730	6.348557e+04	0.973333	84.647431

Product mix A-A has a significantly higher sales per SKU (14) compared to product mix A-B (5). Product mix A-A appears to be more lucrative in terms of both sales and profitability. Businesses may want to investigate what attributes or characteristics of products within this mix contribute to higher sales and profit. This analysis could inform decisions regarding product development, marketing strategies, or inventory management to capitalize further on the success of product mix A-A. product mix A-c should be placed in front as it is fast moving low margin but high volume.



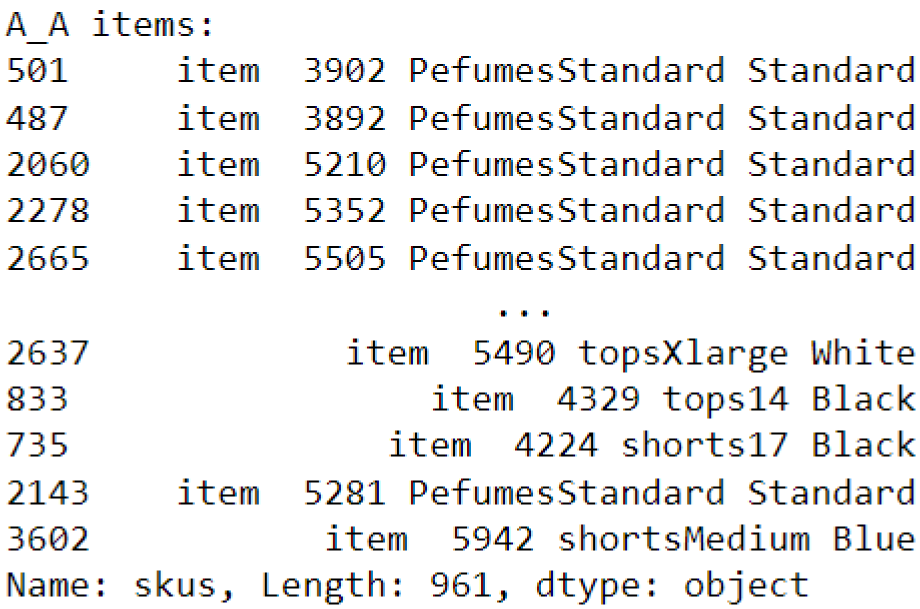


1. **Product Mix Distribution**:

- The product mix “A_A” and "C_C" has the highest count of items (750), followed by "B_B" (715) and "B_C" (491), indicating potentially popular product categories.
- Product mixes "A_C" and "C_A" have the lowest counts of items, suggesting they might represent niche or specialized products.

2. **Sales Performance**:
   - Product mix "A_A" has the highest total sales (13913), indicating it is a significant revenue driver.
   - "C_A" and "C_B" have equal total sales (42 each), which could imply they are less popular or newly introduced products.
3. **Item Count Analysis**:
   - Product mix "C_C" has the highest count of items (750) but relatively lower total sales (730), indicating a potential surplus in inventory or slower sales velocity.
   - Conversely, product mix "A_A" has a high count of items (961) with correspondingly high total sales (13913), suggesting efficient inventory turnover.

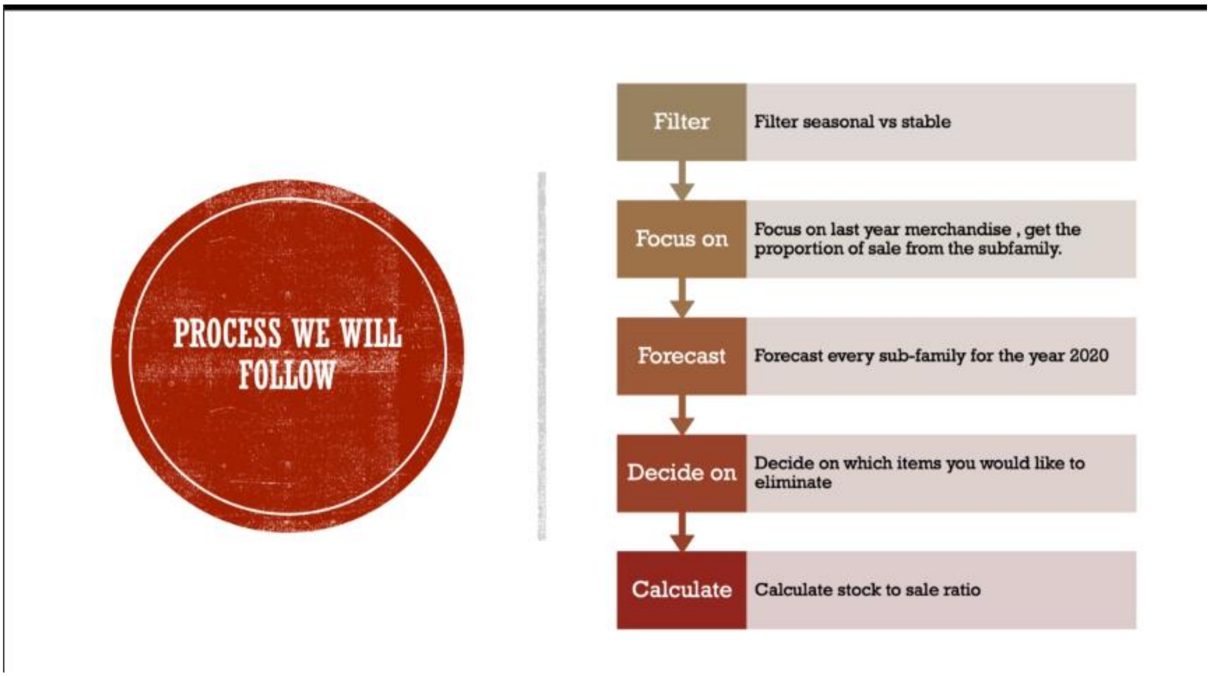
We have also written a code to filter the items that belong to certain product mix category.



```
A_A items:
501     item  3902 PefumesStandard Standard
487     item  3892 PefumesStandard Standard
2060    item  5210 PefumesStandard Standard
2278    item  5352 PefumesStandard Standard
2665    item  5505 PefumesStandard Standard
                           ...
2637            item  5490 topsXlarge White
833                 item  4329 tops14 Black
735               item  4224 shorts17 Black
2143    item  5281 PefumesStandard Standard
3602           item  5942 shortsMedium Blue
Name: skus, Length: 961, dtype: object
```

Task-5: Forecasting:

A sales forecast is an estimate of expected sales revenue within a specific time frame, such as quarterly, monthly, or yearly. It expresses how much a company plans to sell. Forecasters analyse economic conditions, consumer trends, past purchases, and competitors to make accurate predictions. Forecasting is performed to plan ahead how many items the company should buy and to buy the real selling items and to eliminate some items.



PROCESS WE WILL FOLLOW

Filter — Filter seasonal vs stable

Focus on — Focus on last year merchandise , get the proportion of sale from the subfamily.

Forecast — Forecast every sub-family for the year 2020

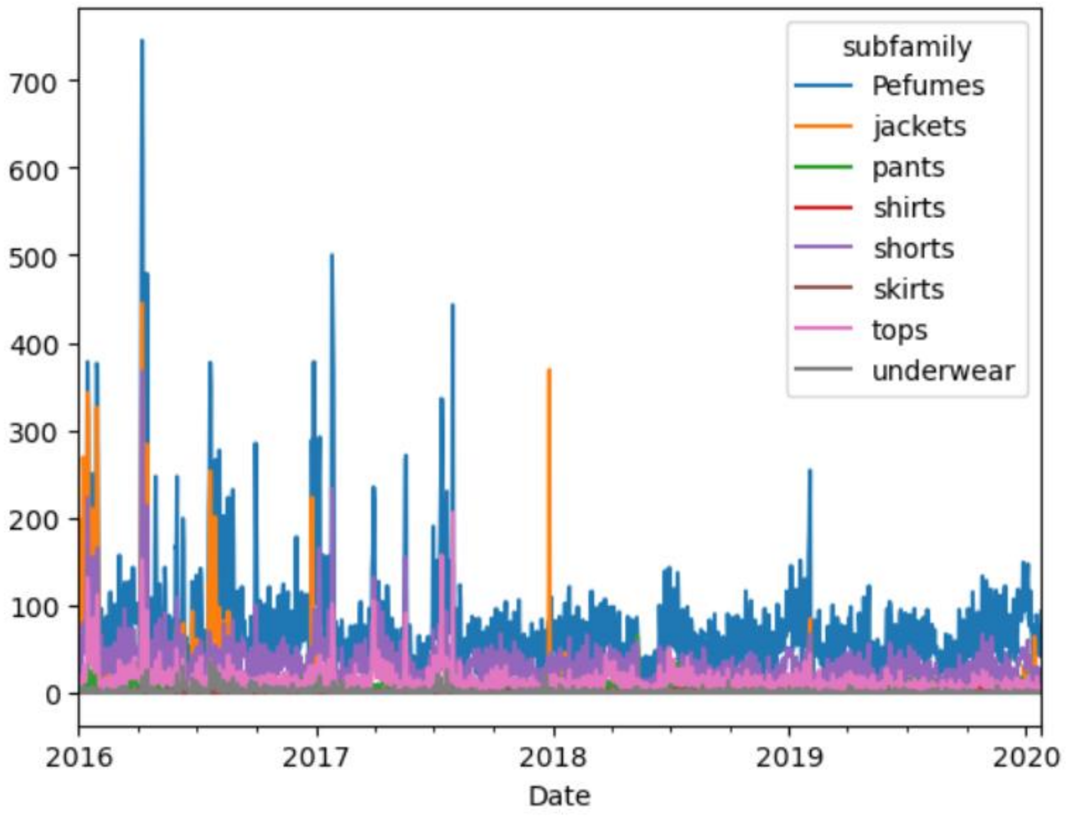
Decide on — Decide on which items you would like to eliminate

Calculate — Calculate stock to sale ratio

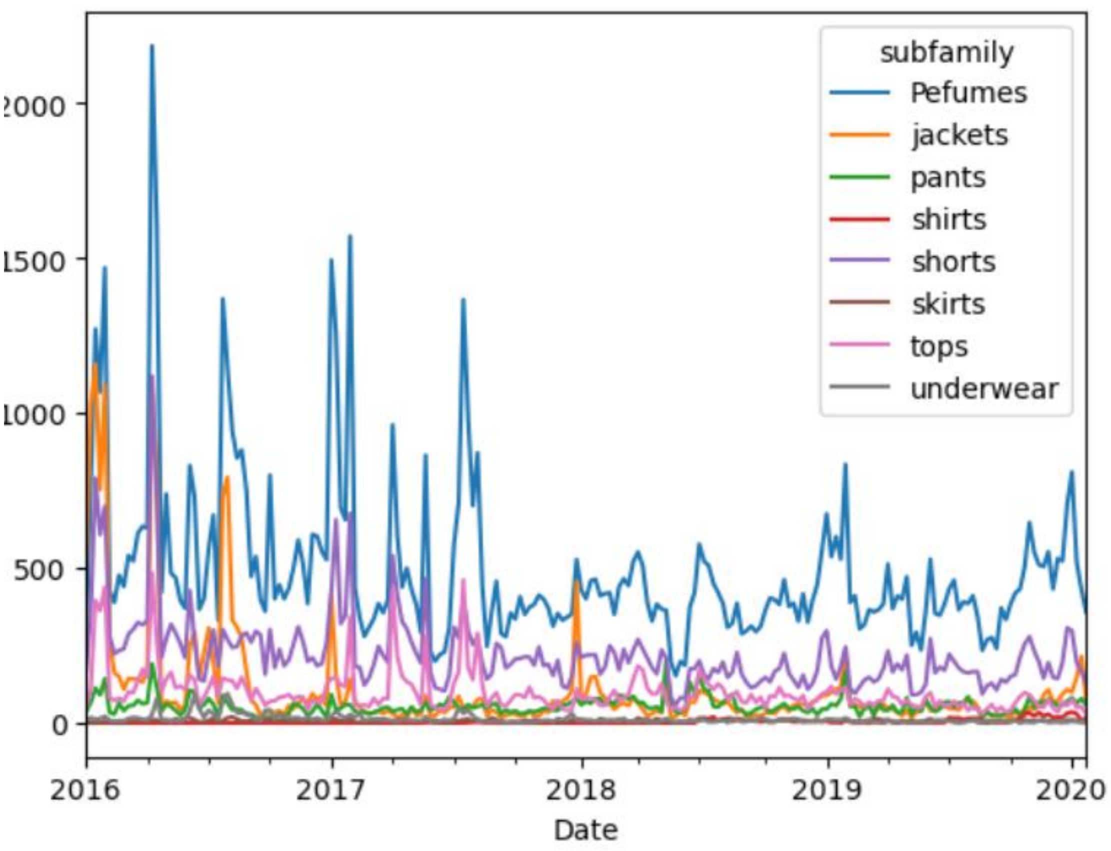
Revenue was calculated by multiplying the unit price paid for each item by the quantity purchased. Profit was then computed by subtracting the total cost of the items from the revenue generated. The 'Date' column was converted into a standardized datetime format to facilitate analysis and visualization. A 'week' column was created to categorize transactions based on the week of the year in which they occurred. This allows for tracking weekly performance trends. Similarly, a 'month' column was added to categorize transactions by the month in which they took place, aiding in monthly performance analysis and comparison.

Then we grouped the data by the 'Date' and 'subfamily' columns, and then sums up the 'Qty' column within each group. This essentially calculates the total quantity of items for each subfamily on each date. created an array containing unique subfamily values from the 'subfamily' column. Pivoted the grouped data (sub_family_grouped) to create a pivot table where dates are the index, subfamilies are columns, and the values are the total quantities ('Qty'). This essentially spreads out the data into a more organized format for visualization or further analysis. Then converted the daily time series to weekly time series using resample method.

Daily time series plot:

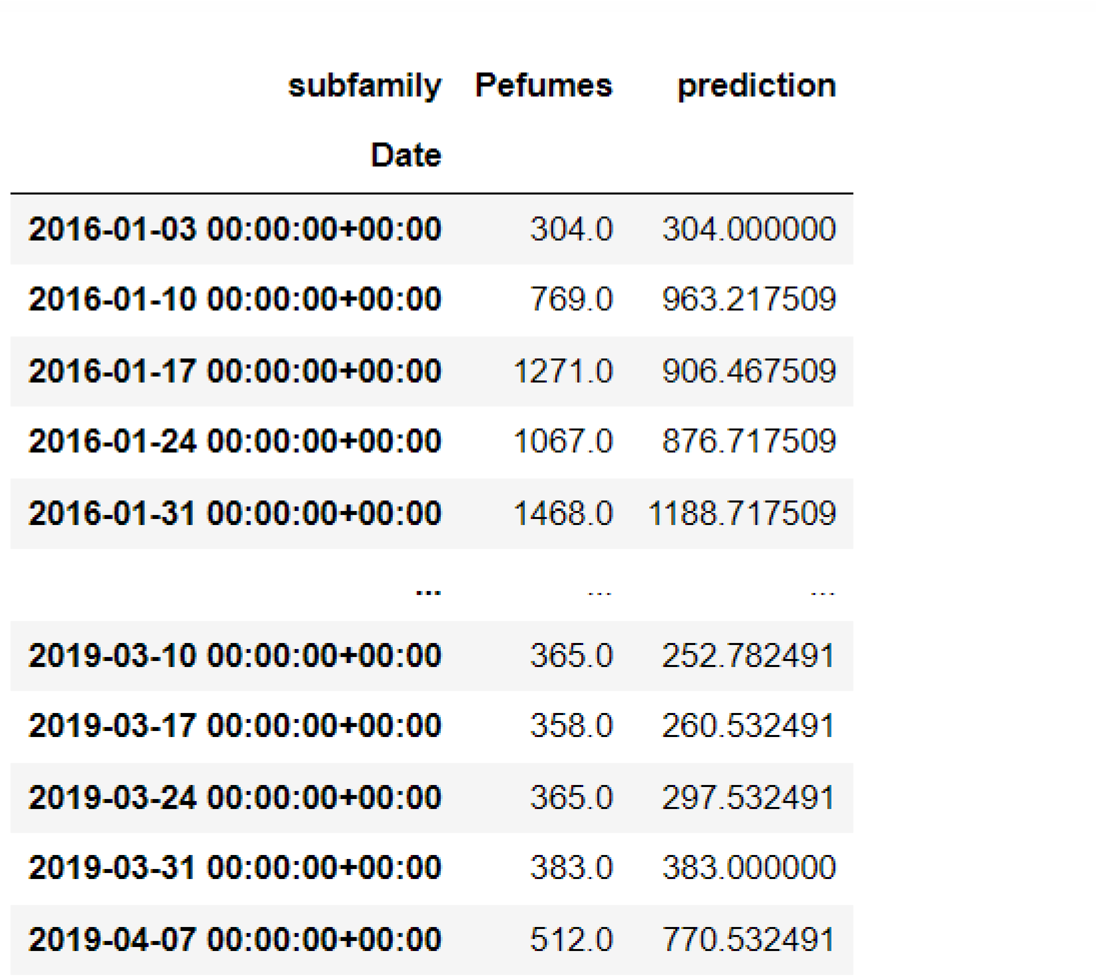


Weekly time series plot:

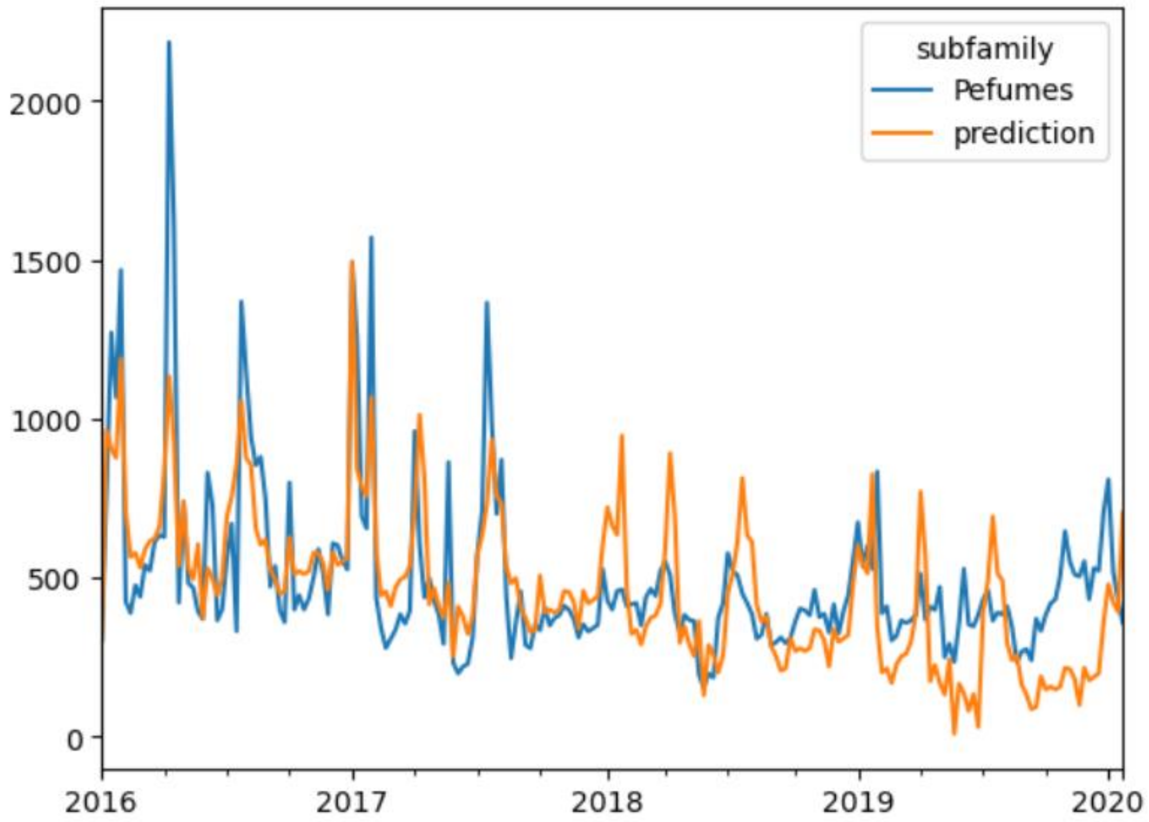


There is some noise and normal ups and downs. Also, the trend is declining. Now we will have 7 different model for 7 different subfamily. Next we will be capturing the trend by adding a trend column which contains a sequence of numbers from 0 to the number of rows in the DataFrame. Two new columns, 'month' and 'week', are created in the 'spreaded_weekly' DataFrame by extracting the month and week information from the 'Date' column. The 'month' and 'week' columns are converted to categorical data types to represent them more efficiently. One-hot encoding is performed on the DataFrame to convert categorical variables into a numerical format suitable for machine learning algorithms.

The independent variables (features) and dependent variable (performance) are separated into training and testing sets. The training set includes data from the beginning up to the 171st row, and the testing set includes data from the 172nd row onwards. A Linear Regression model is instantiated and trained using the training data (features and corresponding performance values). The model's performance on the training data is evaluated using the 'score()' function, which computes the coefficient of determination (R^2).The trained model is used to make predictions on the testing data. Mean Absolute Error (MAE) is calculated to quantify the difference between the actual and predicted performance values The performance of perfumes over time is plotted. The x-axis represents dates, and the y-axis represents the actual and predicted performance values. The training data predictions are plotted first, followed by the predictions on the testing data.

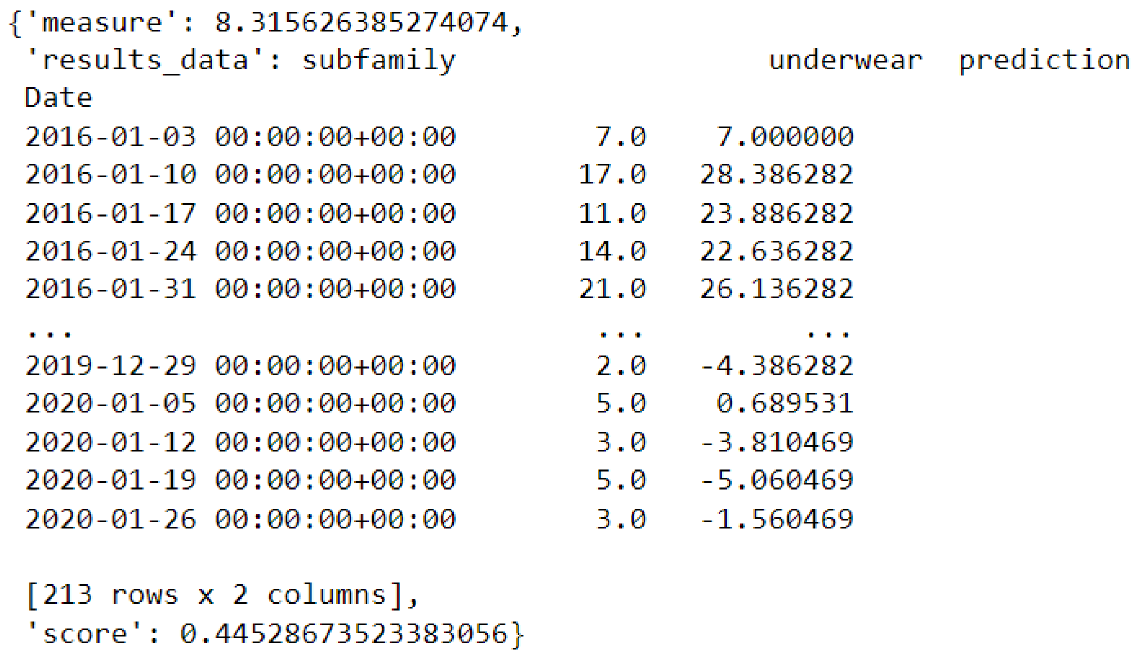


subfamily	Pefumes	prediction
Date		
2016-01-03 00:00:00+00:00	304.0	304.000000
2016-01-10 00:00:00+00:00	769.0	963.217509
2016-01-17 00:00:00+00:00	1271.0	906.467509
2016-01-24 00:00:00+00:00	1067.0	876.717509
2016-01-31 00:00:00+00:00	1468.0	1188.717509
...	...	...
2019-03-10 00:00:00+00:00	365.0	252.782491
2019-03-17 00:00:00+00:00	358.0	260.532491
2019-03-24 00:00:00+00:00	365.0	297.532491
2019-03-31 00:00:00+00:00	383.0	383.000000
2019-04-07 00:00:00+00:00	512.0	770.532491

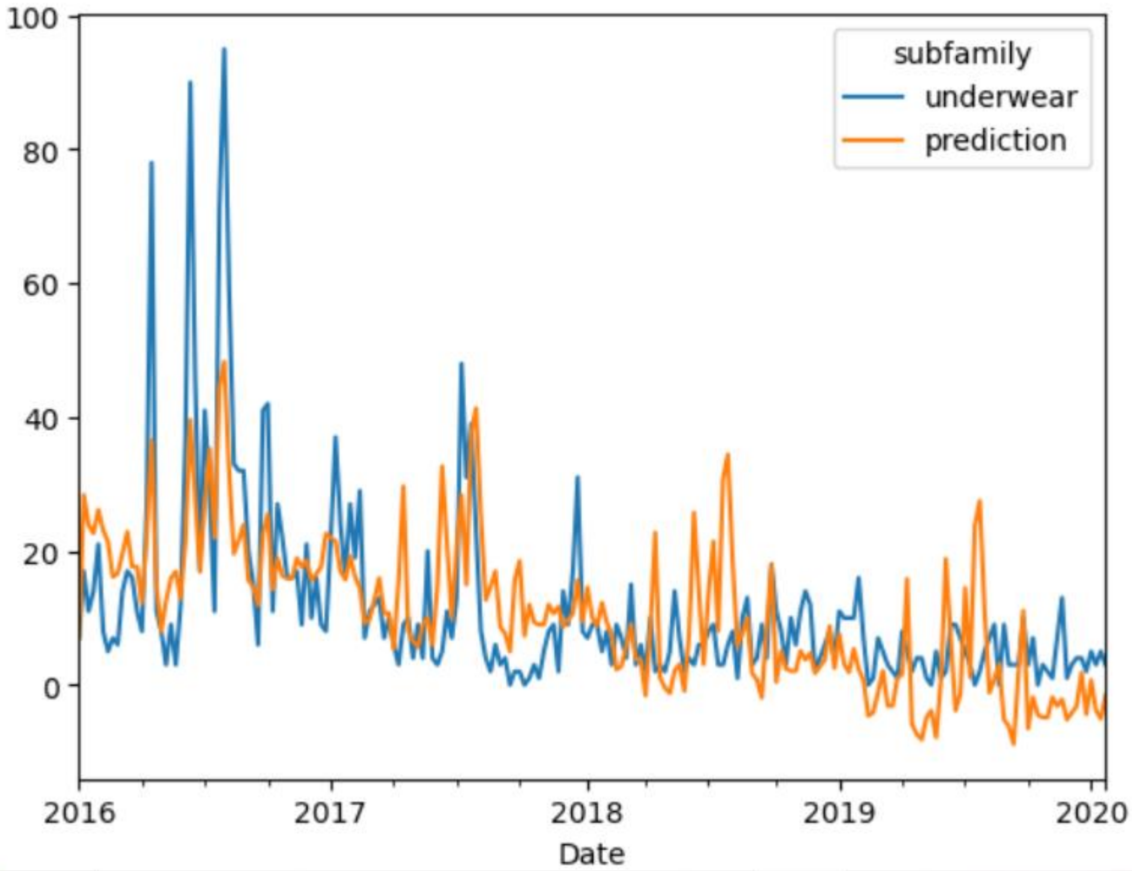


Now we define a forecast function that takes subfamily name as argument and fit the model.

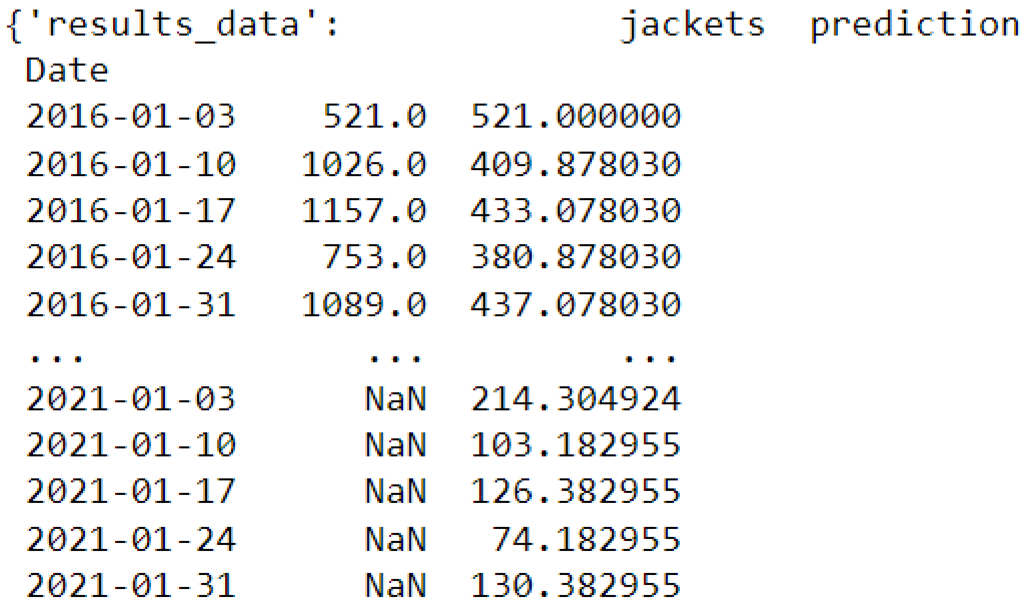
Just checked with forecasting of a item underwear. The results are the following.



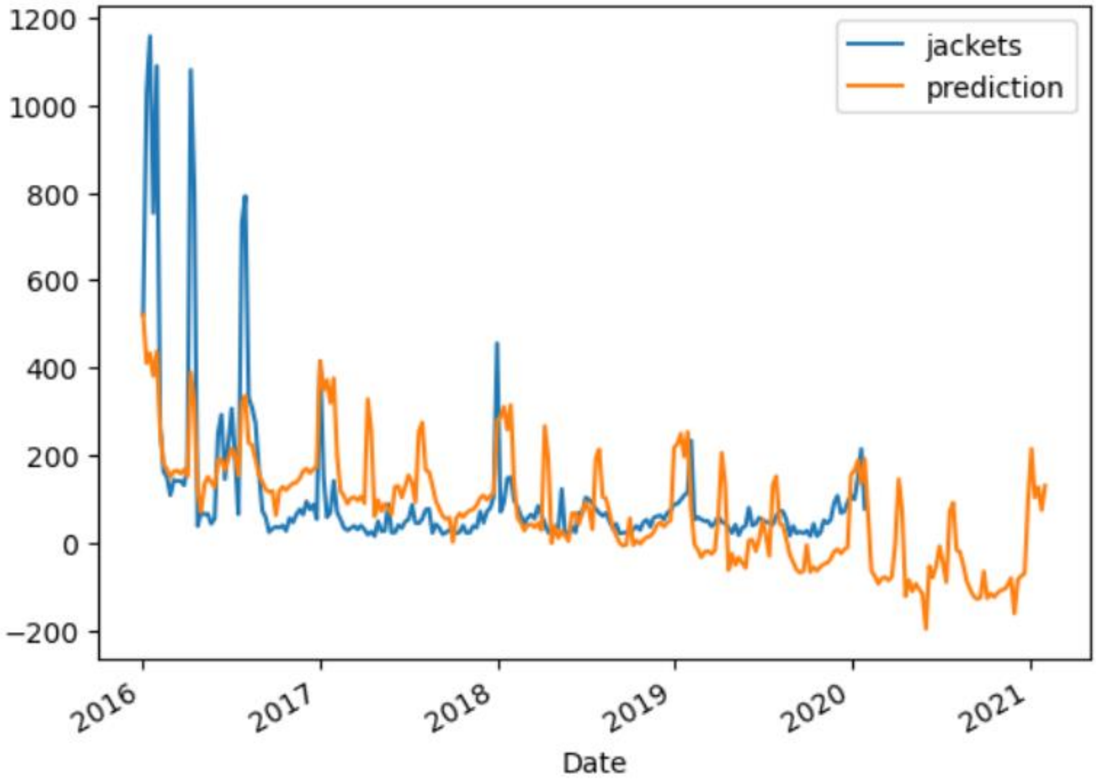
```
{'measure': 8.315626385274074,
 'results_data': subfamily                  underwear  prediction
 Date
 2016-01-03 00:00:00+00:00        7.0    7.000000
 2016-01-10 00:00:00+00:00       17.0   28.386282
 2016-01-17 00:00:00+00:00       11.0   23.886282
 2016-01-24 00:00:00+00:00       14.0   22.636282
 2016-01-31 00:00:00+00:00       21.0   26.136282
 ...                              ...         ...
 2019-12-29 00:00:00+00:00        2.0   -4.386282
 2020-01-05 00:00:00+00:00        5.0    0.689531
 2020-01-12 00:00:00+00:00        3.0   -3.810469
 2020-01-19 00:00:00+00:00        5.0   -5.060469
 2020-01-26 00:00:00+00:00        3.0   -1.560469

 [213 rows x 2 columns],
 'score': 0.44528673523383056}
```

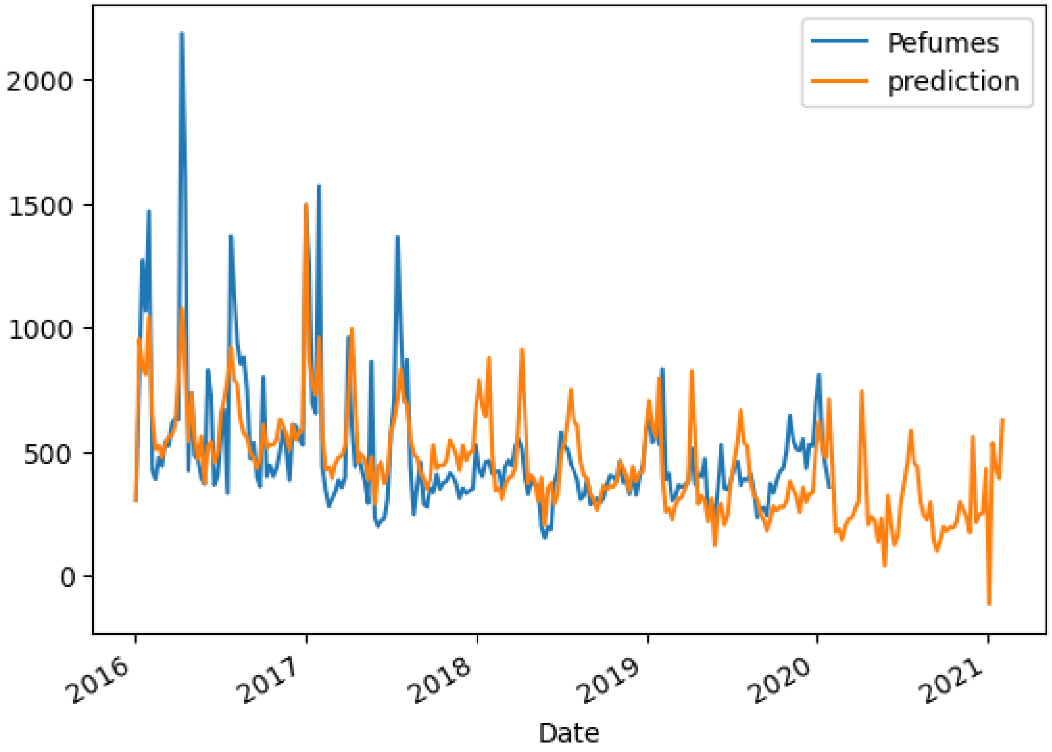


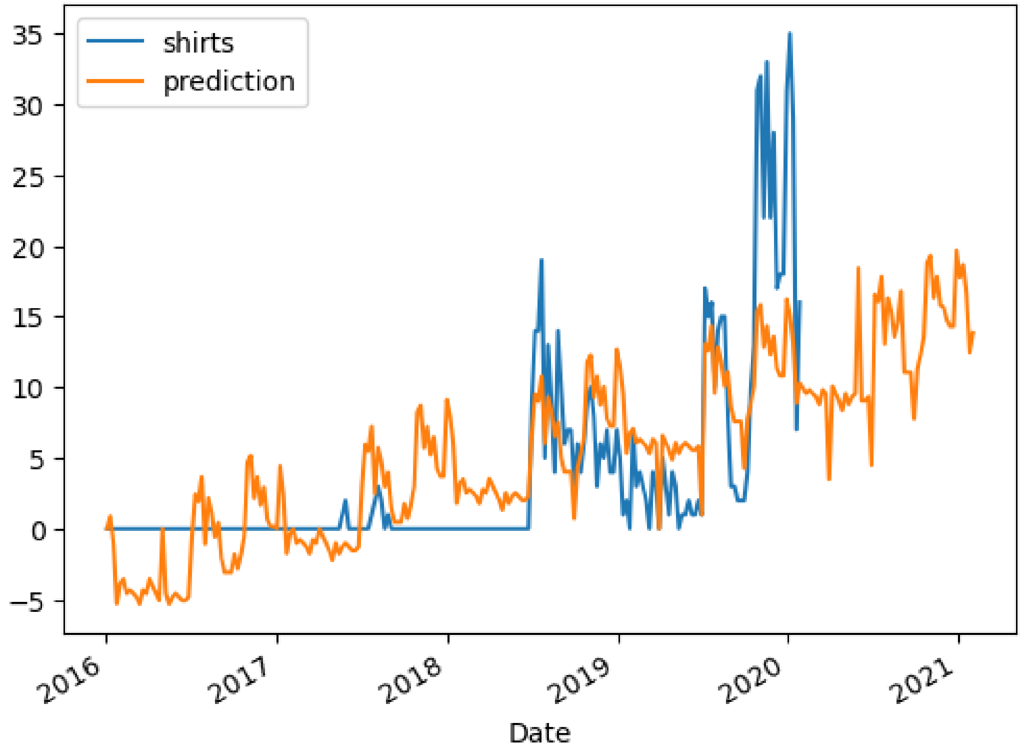
Till now we trained for last 4 years and tested the validity of the model. Now we forecast for the next year 2020 by creating some additional date ranges. The following is the prediction of jacket till next year.

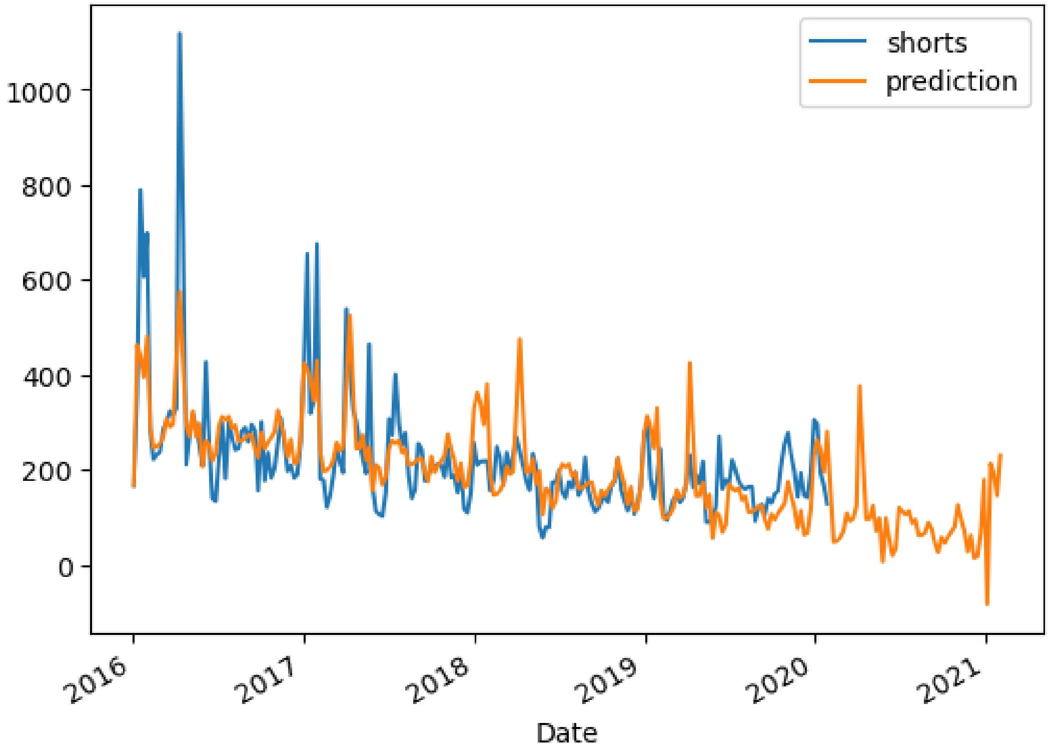


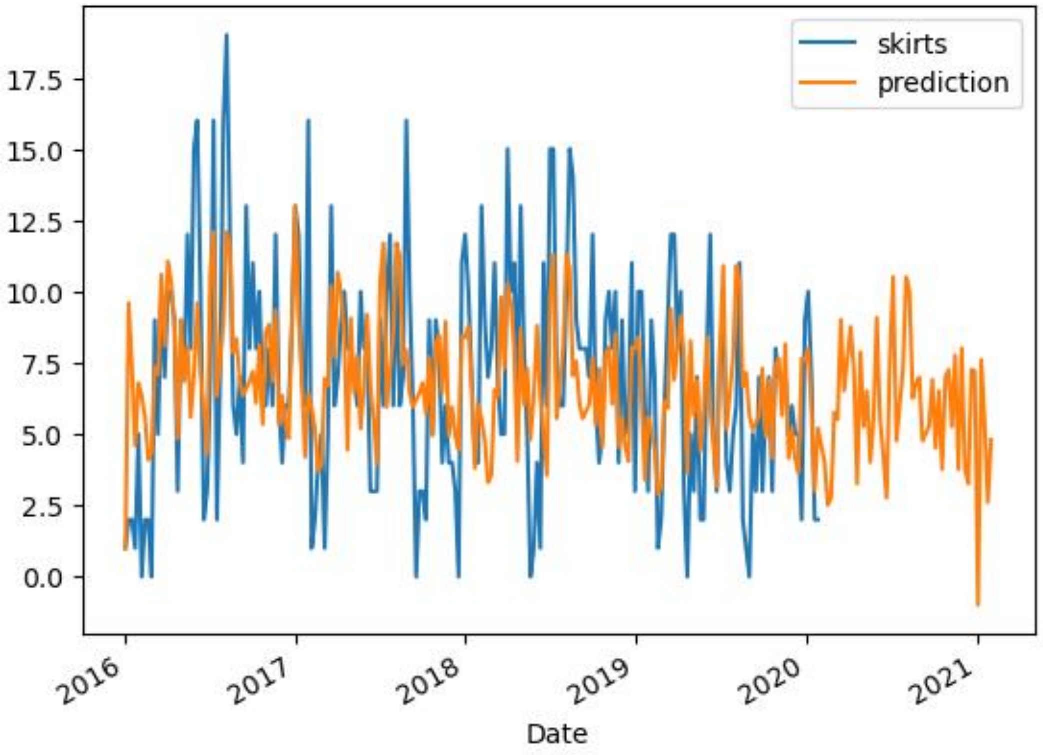
```
{'results_data':              jackets   prediction
 Date
 2016-01-03     521.0   521.000000
 2016-01-10    1026.0   409.878030
 2016-01-17    1157.0   433.078030
 2016-01-24     753.0   380.878030
 2016-01-31    1089.0   437.078030
 ...              ...          ...
 2021-01-03       NaN   214.304924
 2021-01-10       NaN   103.182955
 2021-01-17       NaN   126.382955
 2021-01-24       NaN    74.182955
 2021-01-31       NaN   130.382955
```

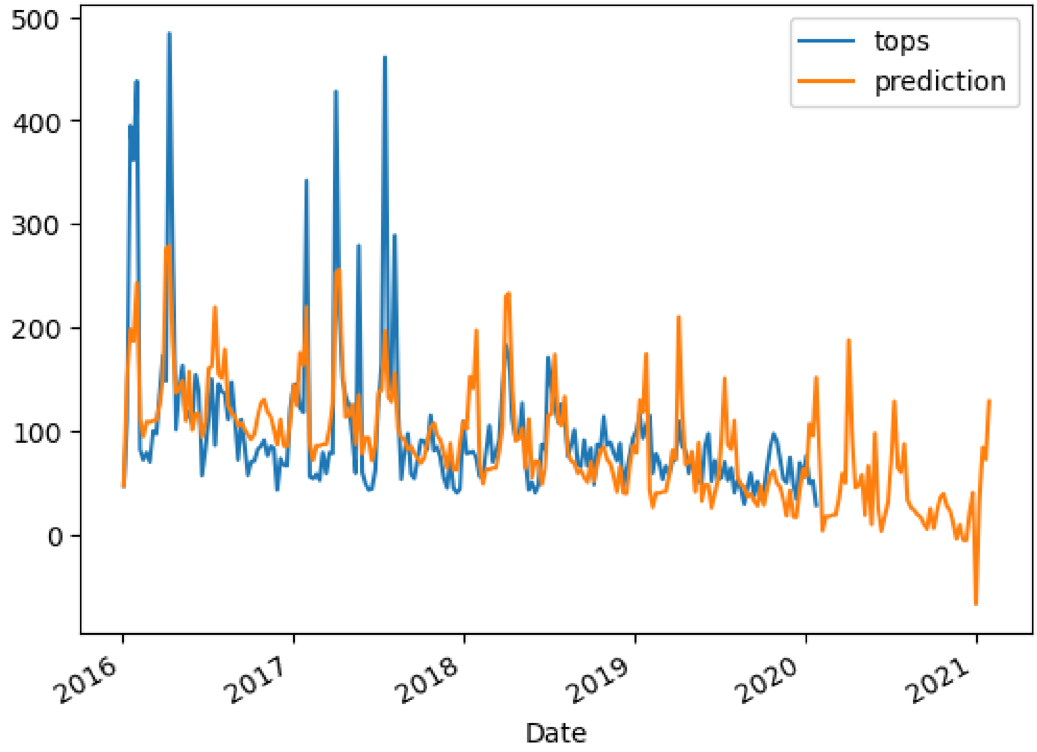

1200
1000
800
600
400
200
0
−200
jackets
prediction
2016
2017
2018
2019
2020
2021
Date

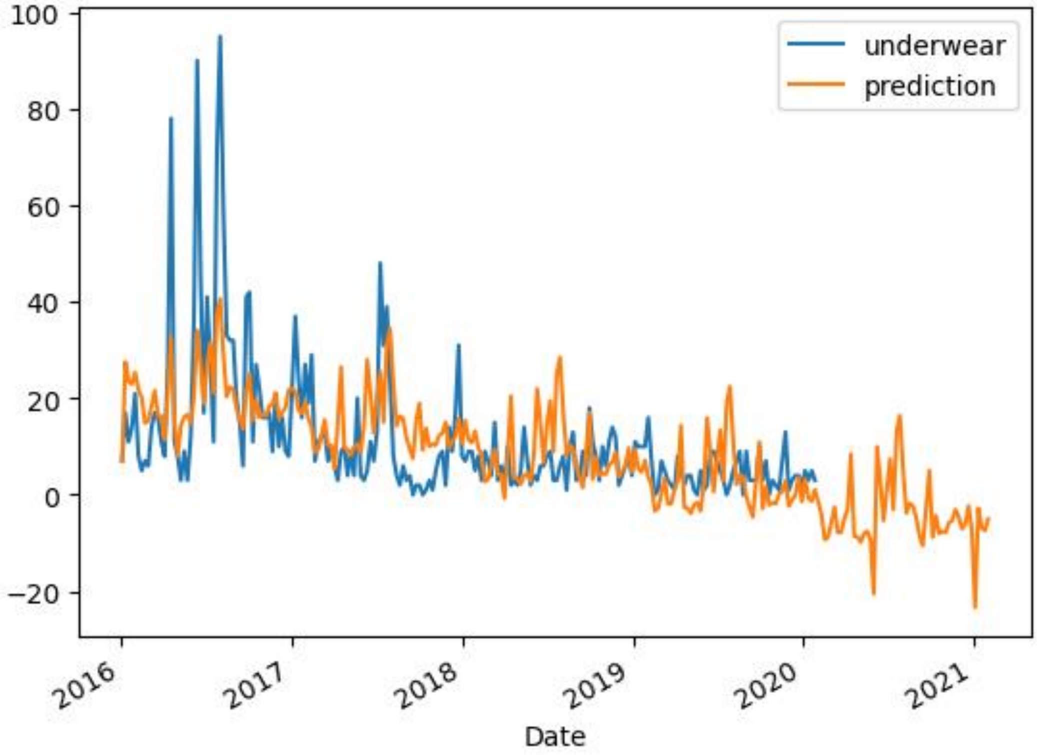

Pefumes
prediction
2000
1500
1000
500
0
2016
2017
2018
2019
2020
2021
Date


shirts
prediction
35
30
25
20
15
10
5
0
−5
2016
2017
2018
2019
2020
2021
Date

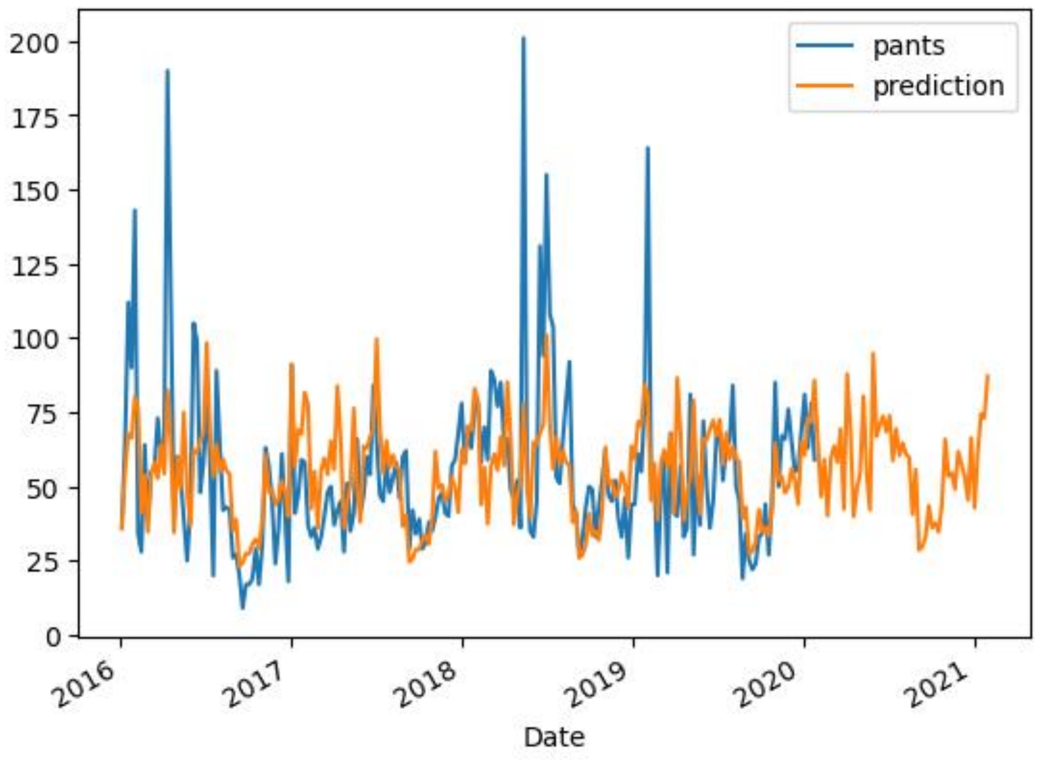

shorts
prediction
1000
800
600
400
200
0
2016
2017
2018
2019
2020
2021
Date


skirts
prediction
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2016
2017
2018
2019
2020
2021
Date

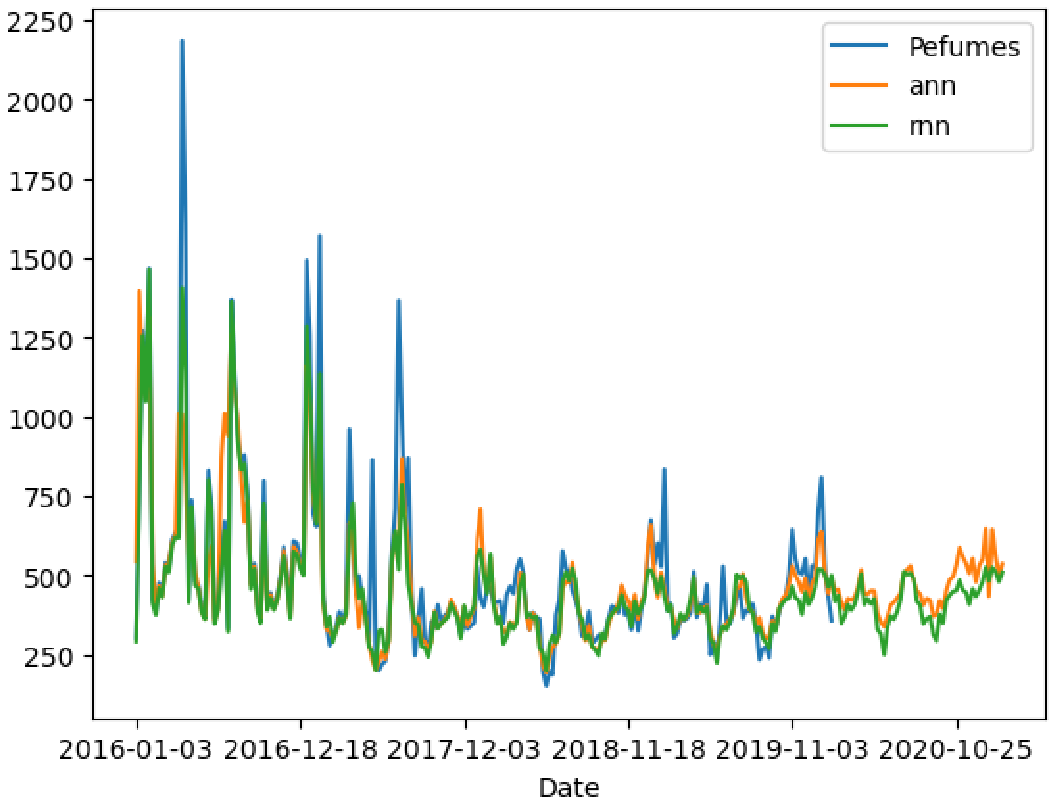


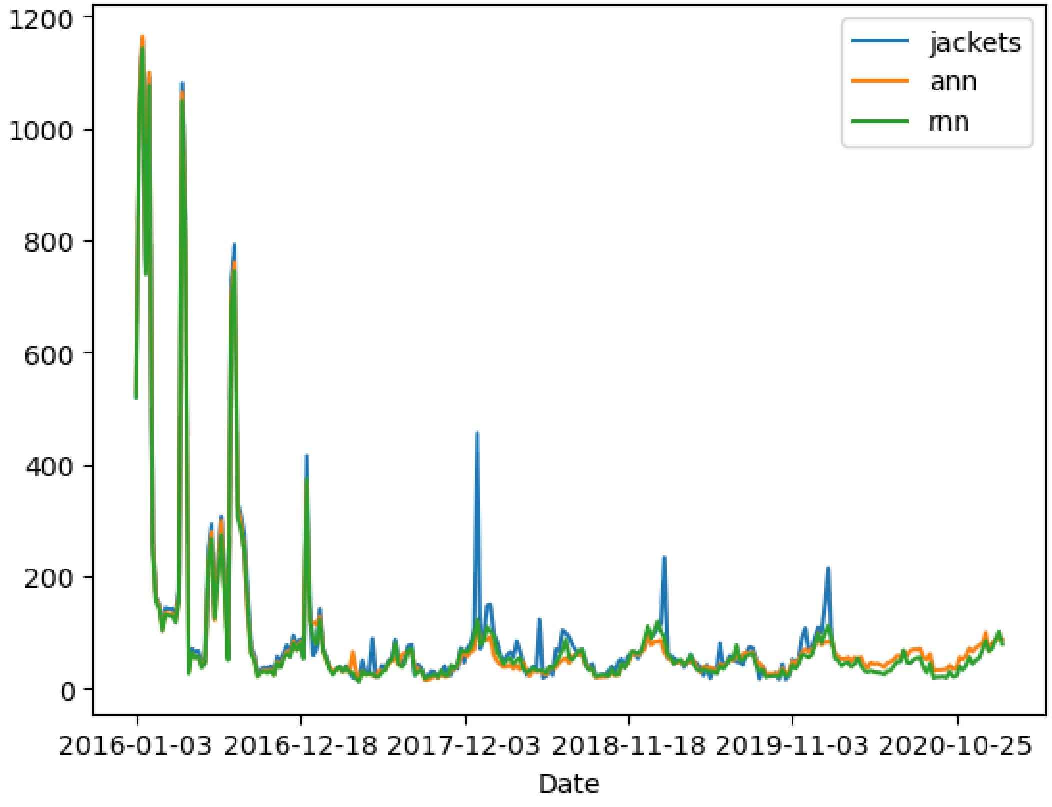


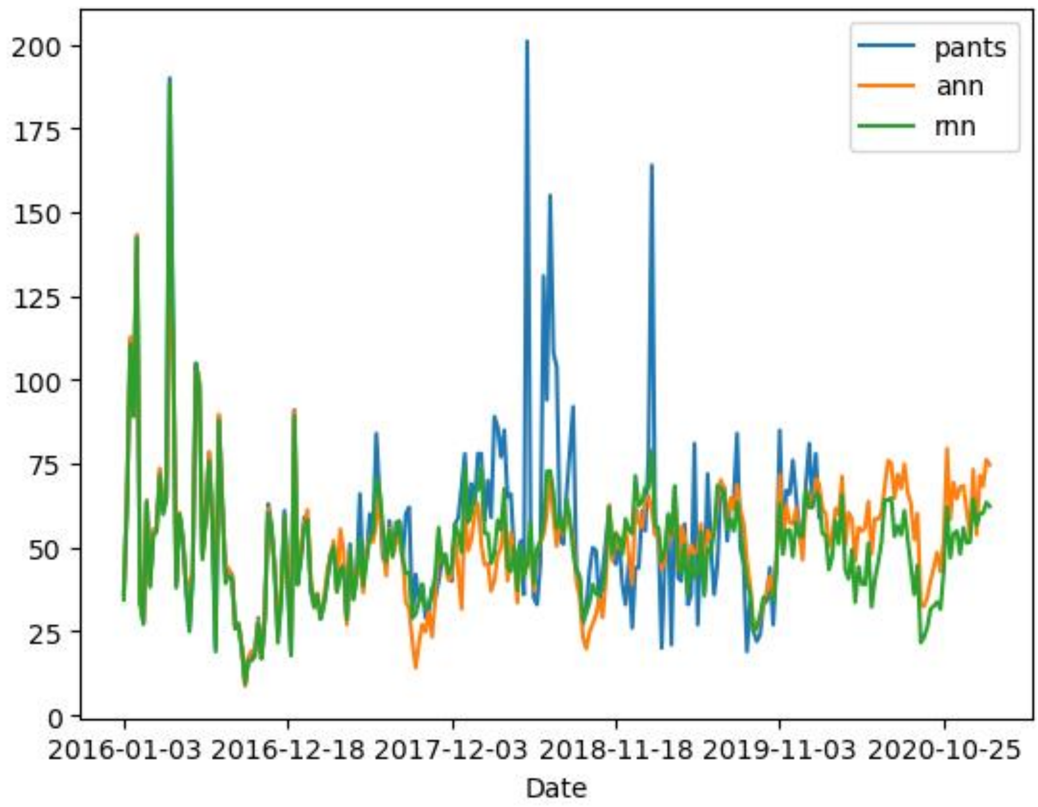
The only one that has an upward trend in future is shirts. Next task is we try to forecast using deep learning models.

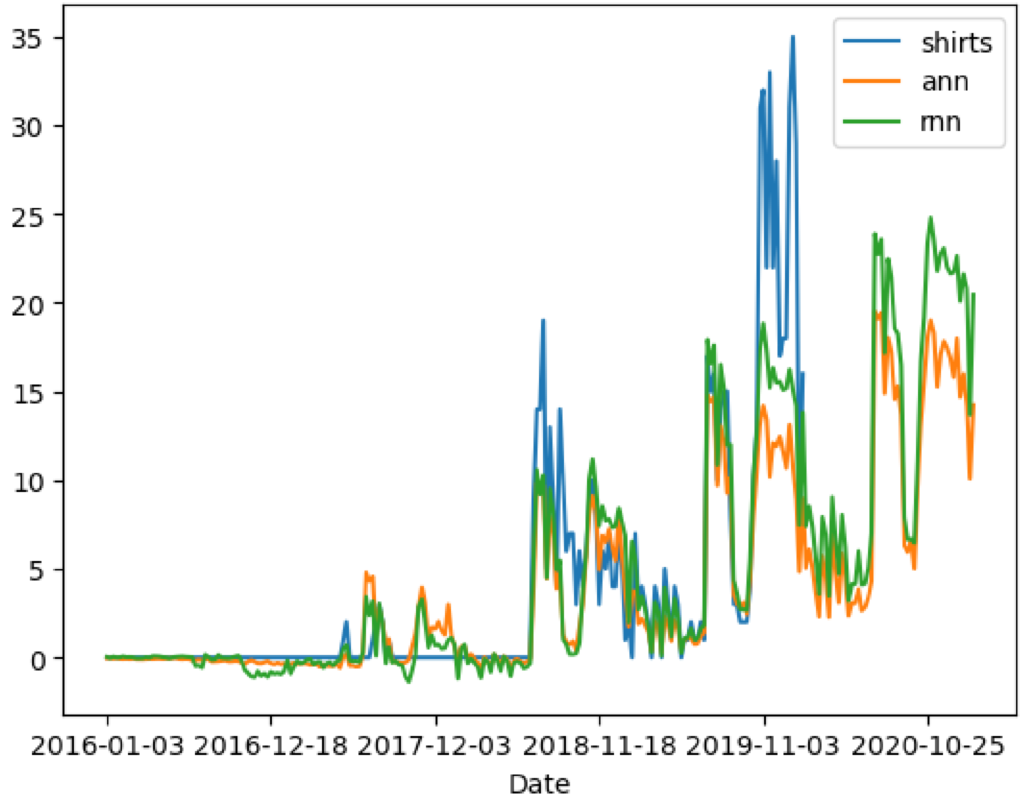


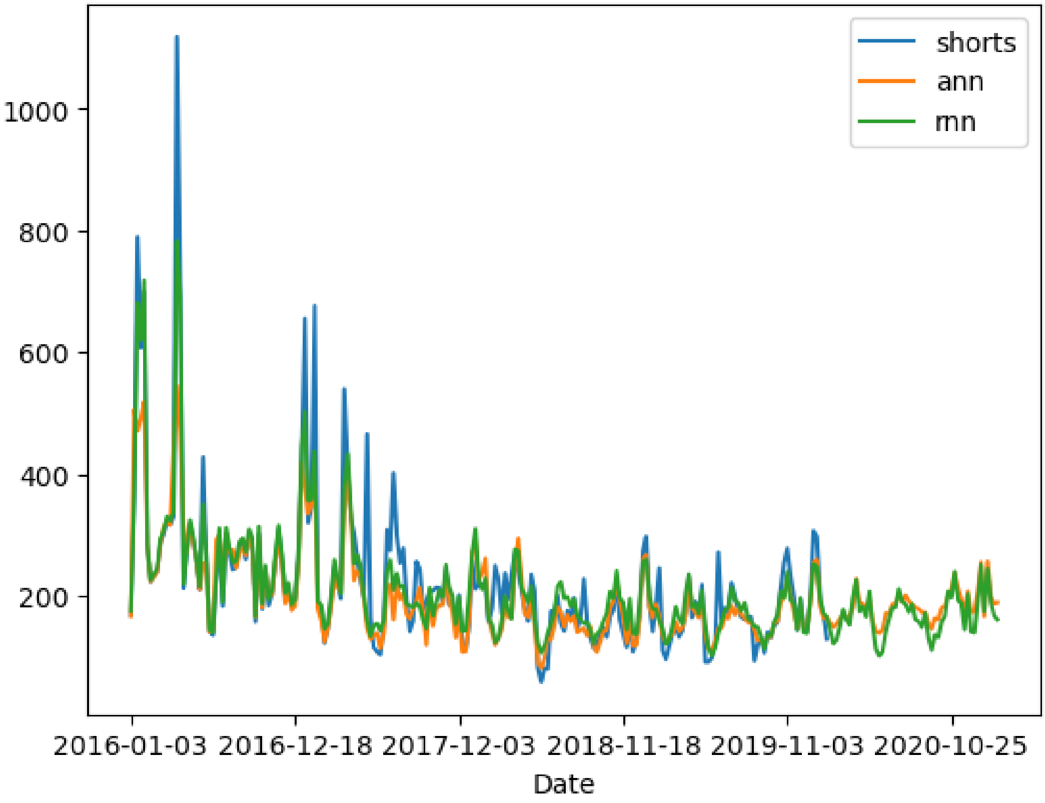
Forecasting using Deep learning models:

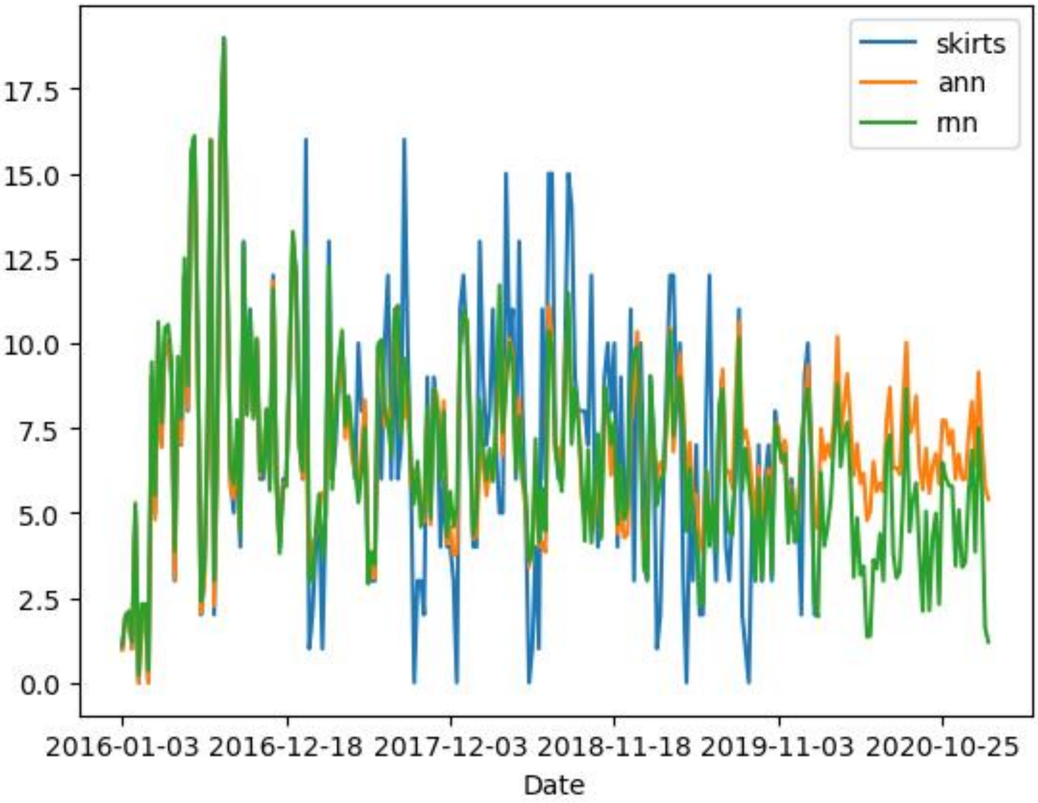

2250
2000
1750
1500
1250
1000
750
500
250
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date
Pefumes
ann
rnn

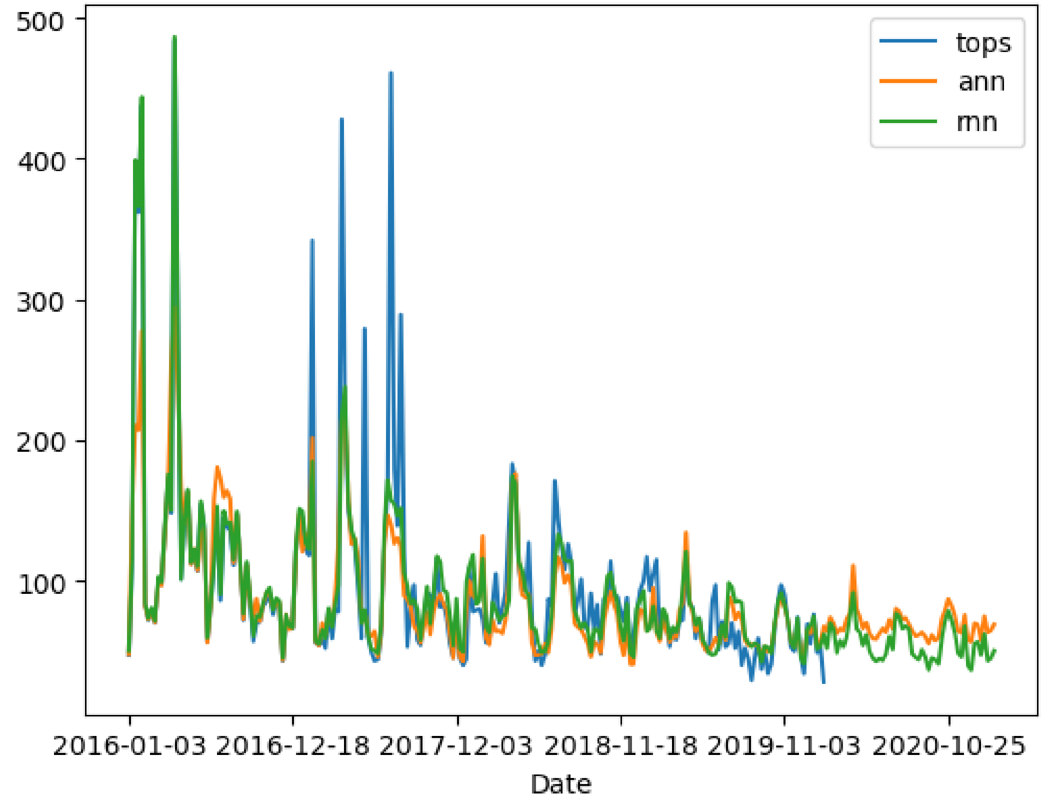

1200
1000
800
600
400
200
0
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date
jackets
ann
rnn

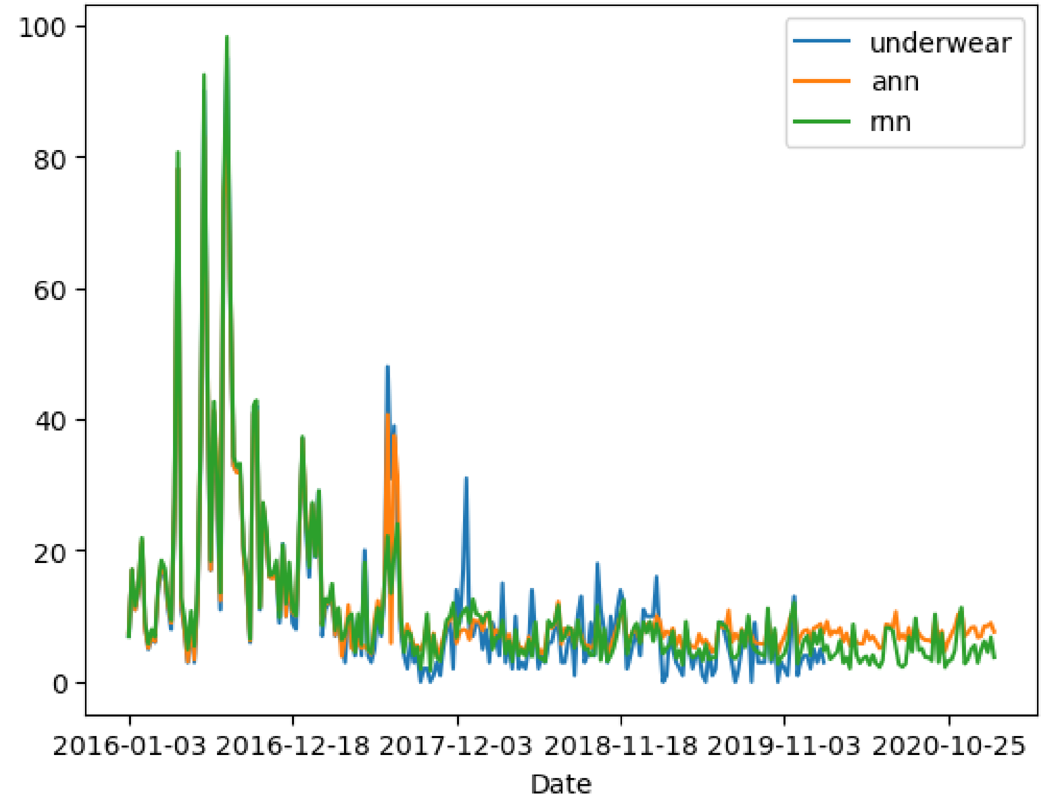

pants
ann
mn
200
175
150
125
100
75
50
25
0
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date


shirts
ann
mn
35
30
25
20
15
10
5
0
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date


shorts
ann
mn
1000
800
600
400
200
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date


skirts
ann
mn
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2016-01-03
2016-12-18
2017-12-03
2018-11-18
2019-11-03
2020-10-25
Date





So for the shirts, the artificial neural network is giving better performance and for all the other items the recurrent neural network gives better results.

Forecasting at the SKU level:

We have already done forecasting for the subfamily. So we use middle out approach now. The middle-out approach combines bottom-up and top-down approaches. First, a “middle level” is chosen and forecasts are generated for all the series at this level. For the series above the middle level, coherent forecasts are generated using the bottom-up approach by aggregating the “middle-level” forecasts upwards. For the series below the “middle level”, coherent forecasts are generated using a top-down approach by disaggregating the “middle level” forecasts downwards. So we look at the contribution of sales of each SKU in previous year and we multiply this contribution with sales of subfamilies to get sales forecast value of SKU. We go bottom up from subfamily to get the sales of family and top down from subfamily to get the sales of SKU.

We need to make a forecast for every SKU. Here SKU is of every different item, different size and different color. So itemno, size and color make a unique key for every SKU. So we created a key first and group by date and key to get the total sales. Our target is to know the contribution of every SKU from total sales of last year. We resampled the data to get weekly time series data points.