Company Sales Data Analytics

A Summer Internship Project Report

Domain: Data Science

Prepared By:

Priyanshu Mohanty (VT20221063)

Guided By:

Mr. Sourav Ghosh

Introduction

Data science is the discipline and study of scientific processes, algorithms, and systems that primarily deal with utilitarian uses of data and information. The modern era has often been dubbed as "**Information Age**" by several entrepreneurs and computer scientists, as nowadays, there is access to more information than ever before, with increasing usage of digital devices all of which leave behind a certain amount of "digital footprint" or associated user data, in more technical parlance.

The wealth of information provides many businesses and organizations with the ability to generate and derive knowledge about trends, patterns, and behaviour of users on the platform. If judiciously analysed and then used, these knowledge systems can prove to be instrumental to learning deeply about how users think and what they might need in the form of products and services, and accordingly suggest improvements in the frontend or backend, and take business decisions.

In this regard, there is a sub-domain known as "**business analytics**." Again, it is the set of processes, tools and technologies used to derive insights that can help in making business decisions and planning. One of the key elements in business analytics is analysing the sales data. The sales can be in any form – offline or online, though with the transition into the digital age, e-commerce has taken over prominently.

In this regard, **Tata Steel**, a reputed Indian iron and steel company, has its version of an ed-tech website (**Tata Digie-Shala**) for facilitating the **skill and capability development** of several employees and professionals, likewise. This portal has several educational and training resources and courses which are sold to customers at nominal prices. The **main motive** and objective of this project is to analyze and **derive useful knowledge** obtained **from the sales data of this**

website to understand trends and patterns and report such findings to the business team, which can better help them in decision-making by having quantifiable parameters to base their decisions on.

Source & Format of Dataset:

The dataset was originally provided in the form of a **Microsoft Excel Open XML Spreadsheet** file with an extension of. xlxs. The data is masked to protect the identity of customers and is in line with the confidentiality clause of the company. But it is generically indicative of the **products** that were purchased on the website by different **persons** using certain **email IDs** at particular **timestamps** between 1st March 2021 – 7th March 2021, a week, which is popularly known as "Founder's Week", which commemorates the founder of the company.

For simplicity's sake, this is first converted into a **CSV (comma-separated values)** file, which provides us with the **flexibility to perform various data analytics operations**, as data is structured neatly in a tabulated format.

Software Requirements

Since I've been tasked with data analytics, I choose a robust programming language that has a lot of dev and community support with built-in tools and is known for its versatility in the data science domain – **Python**. The other modules and requirements are as listed below:

- **Python version 3** or higher should be used which has modern functionalities and features.

- **Jupyter Notebook** in **Anaconda**: JP Notebook is an online web-based interface which provides efficient computing and compilation power for data science applications and Anaconda is a tool for simplifying python package management and is also very useful for complex projects with many dependencies. Having both of these in setup saves time and effort involved for a lot of prerequisites.

- **NumPy:** It's essentially a library for Python, which lends support to large and multi-dimensional arrays and matrices, along with associated high-level mathematical functions operable on them.

- **Pandas:** A software library typically used for data manipulation and analysis. This becomes important as most of the time tasks will be performed using tools from this library – the key benefit being that it offers numerical tables, data structures, and time series.

- **Matplotlib** and **Seaborn**: Data visualization tools that aid in plotting graphs, bar charts, pie charts, scatter plots, histograms, etc., and aid in a visual representation of data which helps normal users to understand data trends better.

- Any modern-day multithreading and multiprocessing supported OS like **Windows, Linux & iOS** are efficient.

Apart from the above-mentioned modules, others that have been made use of occasionally include – **statistics, WordCloud, apyori, random**, etc.

Hardware Requirements

Processor:

1. An ***updated generation of the processor*** is compatible with better performance, efficiency, hardware compatibility, thermal management, and power efficiency. In this regard, ***Intel's 11th Generation processor and AMD's 5th Generation processor*** come in handy.

2. ***Number of cores and threads*** form another important factor. It denotes simply the number of independent CPUs in a single chip. Threads are the instructions that are processed by a single CPU core. A ***minimum of 4 cores and 8 threads*** are recommended for best optimization.

3. ***Cache memory*** acts as a buffer between CPU and RAM. So, a minimum of ***8 MB*** of cache memory is typically recommended so that there is enabling faster retrieval of data and instructions from memory.

4. ***Clock speed*** should be more so that the processor churns out faster in nature.

RAM:

RAM is important especially because it allows for multi-tasking. It is recommended to go for ***8GB or more***. Going for 4 GB is very risky and strictly counterproductive as almost about 60-70% of it will be almost always used by the OS.

Secondary Storage:

1. ***Storage type*** matters a lot in this context. HDD (Hard Disk Drive) are typically very slow, even if the laptop comes in with i7. HDDs take much time to open and load a program because they have mechanical parts which delay the processing of information and reduce durability. A better option/alternative is to go for ***SSDs (Solid State Drives)*** which are more powerful due to no moving parts and also impart greater durability.

2. ***Storage size*** also has a great impact, and as such, it is recommended to go anywhere between ***512 GB – 2 TB*** depending on affordability.

GPU (Graphical Processing Unit):

1. Recommended size is **4GB or greater**.

2. Reputable brands should be considered, for example, **NVIDIA, AMD**, etc. A separate GPU might have more than 100 cores which can greatly enhance processing power.

Proposed Workflow & Implementation

Since I've been imbued with the task of studying the dataset and extricating or extracting meaningful information from it which should then be produced to the business team for its appraisal, I've to do all the steps that are customarily required in any data science lifecycle typically. They are as enlisted below:

Studying Dataset:

Perhaps, the most important task of all is to properly comprehend the data that is presented to us. In this case, there're primarily four types of data: ***Customer***

Name, ***Customer's Email***, ***Products Purchased*** & the ***Timestamp*** at which it was purchased.

Based on these data, we can form a preliminary chain of thought as to what all insights can be derived. Some of the more obvious ones would be → most recurring customer, top-selling product, etc. All other insights are discussed in another section.

Data Cleaning & Feature Engineering:

Data cleaning is defined as the complete set of steps or processes which seeks to fix and remove incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. This is important as we need correctly labelled and formatted data for our purpose of deriving insight using pre-existing tools.

Feature engineering, although, not austerely essential, is a very helpful step in the lifecycle. It is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In deriving the insights that have been enlisted below, we didn't make use of any ML method as such (apart from data mining in market basket analysis), so this method is typically not used.

Exploratory Data Analysis:

Exploratory data analysis (EDA) is an approach to analysing datasets preliminarily by studying statistical data such as mean, mode, median, and quartiles, to name a few. The following EDA approaches were used for the dataset in question {the data frame for the dataset has already been created as '***Order_Details'***}:

```
In [3]: Order_Details.head()
```

Out[3]:

	Name	Email	Product	Transaction Date
0	PERSON_1	PERSON_1@gmail.com	PRODUCT_75	01/03/2021 00:47:26
1	PERSON_2	PERSON_2@tataprojects.com	PRODUCT_75	01/03/2021 02:04:07
2	PERSON_3	PERSON_3@gmail.com	PRODUCT_63	01/03/2021 09:10:43
3	PERSON_4	PERSON_4@gmail.com	PRODUCT_63	01/03/2021 09:49:48
4	PERSON_5	PERSON_5@gmail.com	PRODUCT_34,PRODUCT_86,PRODUCT_57,PRODUCT_89	01/03/2021 10:56:46

Firstly, the first few sample data were viewed.

```
In [4]: Order_Details.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581 entries, 0 to 580
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Name              581 non-null    object
 1   Email             581 non-null    object
 2   Product           581 non-null    object
 3   Transaction Date  581 non-null    object
dtypes: object(4)
memory usage: 18.3+ KB
```

Secondly, certain information about the type of data present was obtained → such as whether is it an object or not and how many occurrences were there and what the total memory occupied.

```
In [5]: Order_Details.describe()
```

Out[5]:

	Name	Email	Product	Transaction Date
count	581	581	581	581
unique	525	525	252	581
top	PERSON_470	PERSON_470@gmail.com	PRODUCT_75	01/03/2021 00:47:26
freq	5	5	74	1

Thirdly, we find information like the number of distinct data values, frequency of such occurrence, etc., details.

```
In [6]: Order_Details.shape
Out[6]: (581, 4)

In [7]: Order_Details.isnull().sum() #no null value, so this part doesn't need to be cleaned
Out[7]: Name              0
        Email             0
        Product           0
        Transaction Date  0
        dtype: int64
```

Finally, we get info on the dimensions (rows, columns) and whether there is any null value which might have to be restructured or redefined.

There's one other reformatting we need to perform before we move on to insights delivery: - the data in the product column has been clubbed together in a single row for a customer purchasing multiple products at the same time. This might make our analysis of individual products difficult, so we need to segregate them into different columns keeping all the other details intact:

```
In [11]: reshaped = \
         (Order_Details.set_index(Order_Details.columns.drop('Product',1).tolist())
            .Product.str.split(',', expand=True)
            .stack()
            .reset_index()
            .rename(columns={0:'Product'})
            .loc[:, Order_Details.columns]
         )

In [12]: print(reshaped)
             Name                    Email     Product     Transaction Date
0        PERSON_1        PERSON_1@gmail.com  PRODUCT_75  01/03/2021 00:47:26
1        PERSON_2  PERSON_2@tataprojects.com  PRODUCT_75  01/03/2021 02:04:07
2        PERSON_3        PERSON_3@gmail.com  PRODUCT_63  01/03/2021 09:10:43
3        PERSON_4        PERSON_4@gmail.com  PRODUCT_63  01/03/2021 09:49:48
4        PERSON_5        PERSON_5@gmail.com  PRODUCT_34  01/03/2021 10:56:46
...           ...                       ...         ...                  ...
1344   PERSON_524      PERSON_524@gmail.com  PRODUCT_86  07/03/2021 23:59:26
1345   PERSON_524      PERSON_524@gmail.com  PRODUCT_63  07/03/2021 23:59:26
1346   PERSON_524      PERSON_524@gmail.com  PRODUCT_54  07/03/2021 23:59:26
1347   PERSON_525      PERSON_525@gmail.com  PRODUCT_66  07/03/2021 23:59:19
1348   PERSON_525      PERSON_525@gmail.com  PRODUCT_34  07/03/2021 23:59:19

[1349 rows x 4 columns]
```

Now, the valuable insights derived will be discussed in the upcoming section.

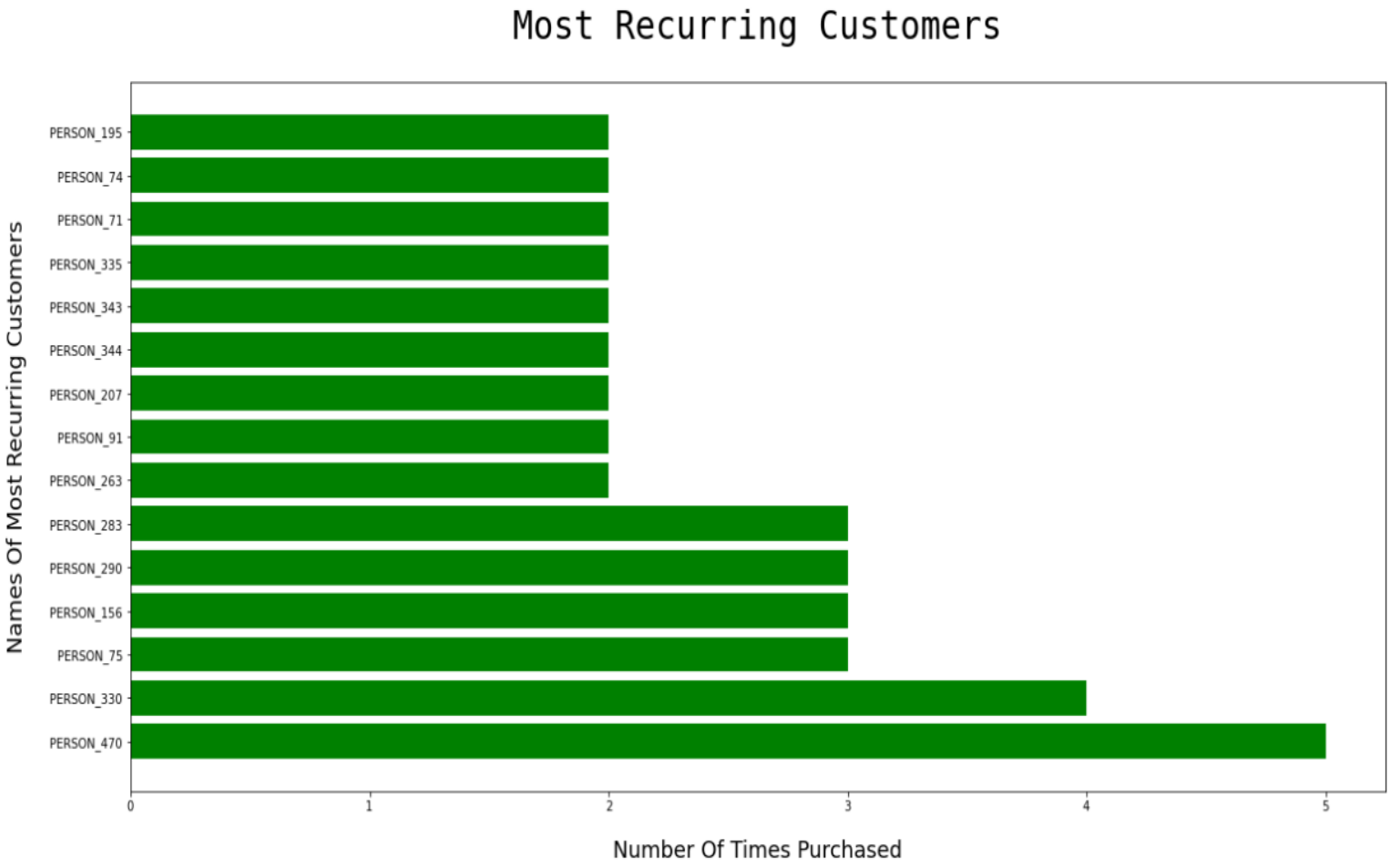
Results & Discussion

#1: Most Recurring Customers

A recurring customer is someone who makes multiple purchases of the same or different products. Finding the insight of '***most recurring customers***' will help us understand which customers are more likely to be retained for the long term, and on the availability of more profiles, we can contact him on offers.

A simple Pandas function of ***value_counts*** alone was sufficient to yield the result which was sorted in descending order and then plotted on a graph (top 10):

Customer Name	Number of Times Purchased
PERSON_470	5
PERSON_330	4
PERSON_75	3
PERSON_156	3
PERSON_290	3
PERSON_283	3
PERSON_263	2
PERSON_91	2
PERSON_207	2
PERSON_344	2
PERSON_343	2
PERSON_335	2
PERSON_71	2
PERSON_74	2
PERSON_195	2

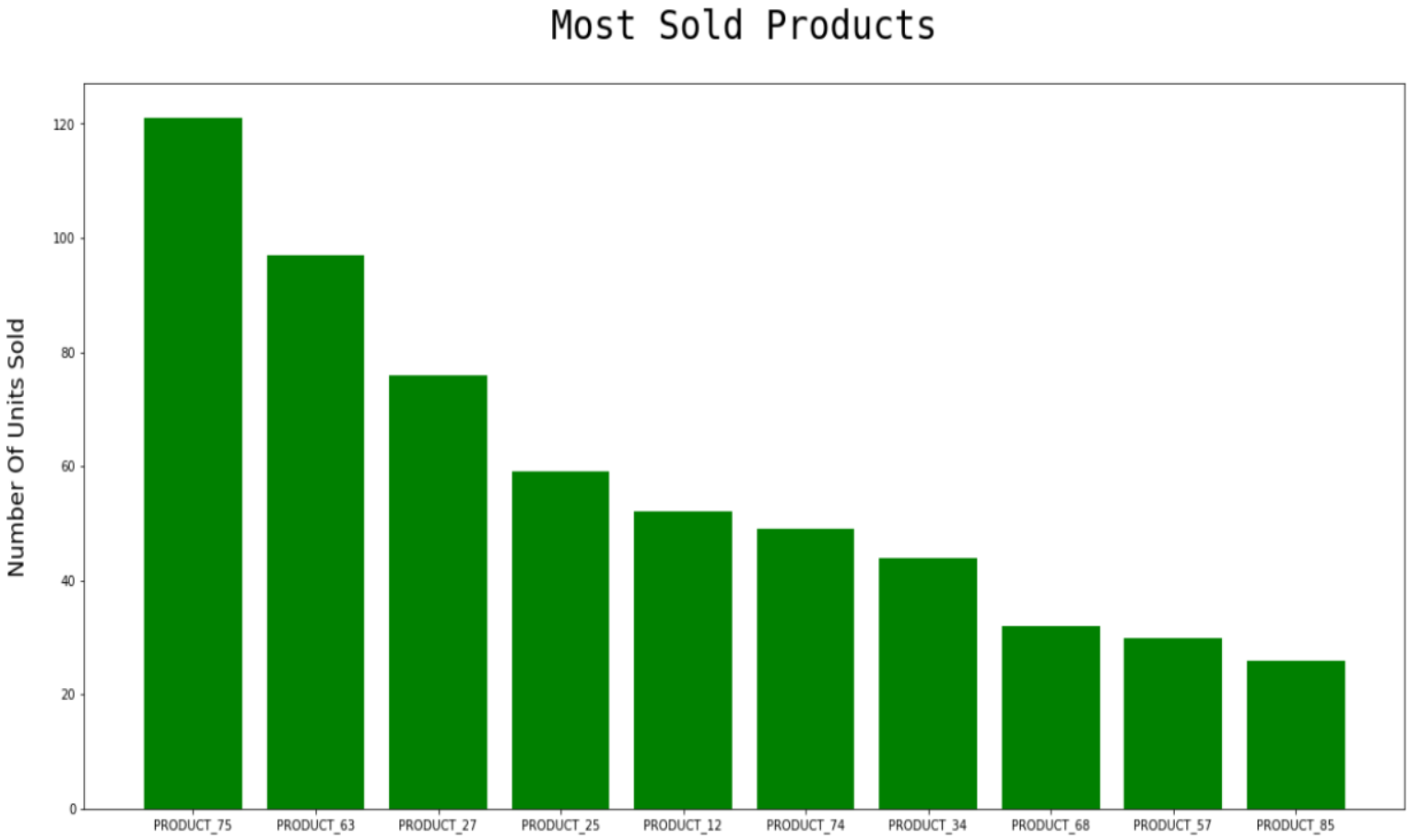


#2: Most Sold Products

The label itself is suggestive that we want to find the ***top-selling products*** which are very popular and in demand, and accordingly, the business team can come up with offers to rake in more profit by spiking prices, offering deals, etc. This insight is also useful as it is also a measure of the quality of the product, as more people are likely to buy a product which has better value.

So, just like the previous insight, we use the Pandas function of ***value_counts***. But here, we do it with the reshaped structure and finally plot it (for the top 10):

Product Name	Units Sold (Number/Quantity)
PRODUCT_75	121
PRODUCT_63	97
PRODUCT_27	76
PRODUCT_25	59
PRODUCT_12	52
PRODUCT_74	49
PRODUCT_34	44
PRODUCT_68	32
PRODUCT_57	30

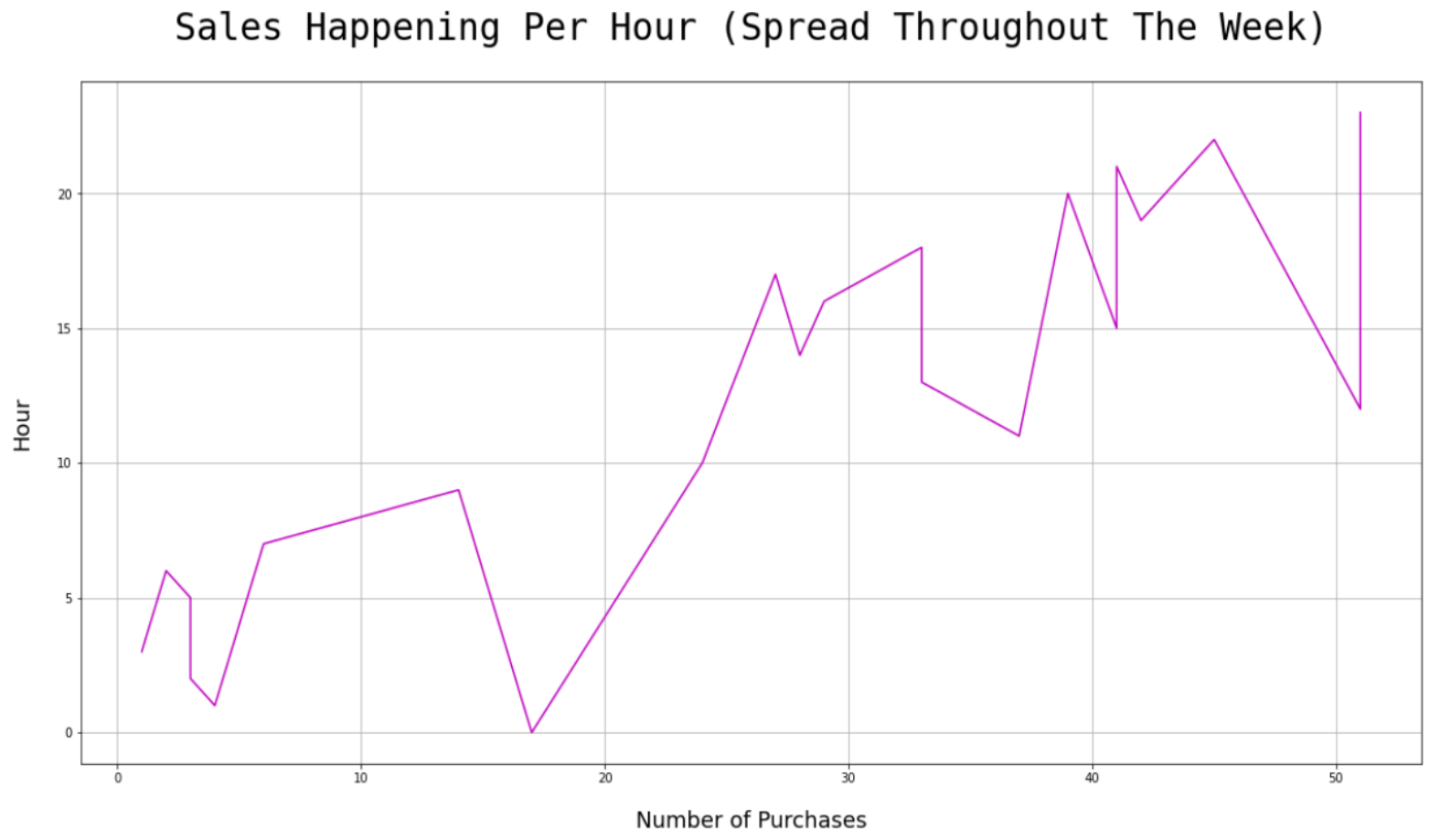


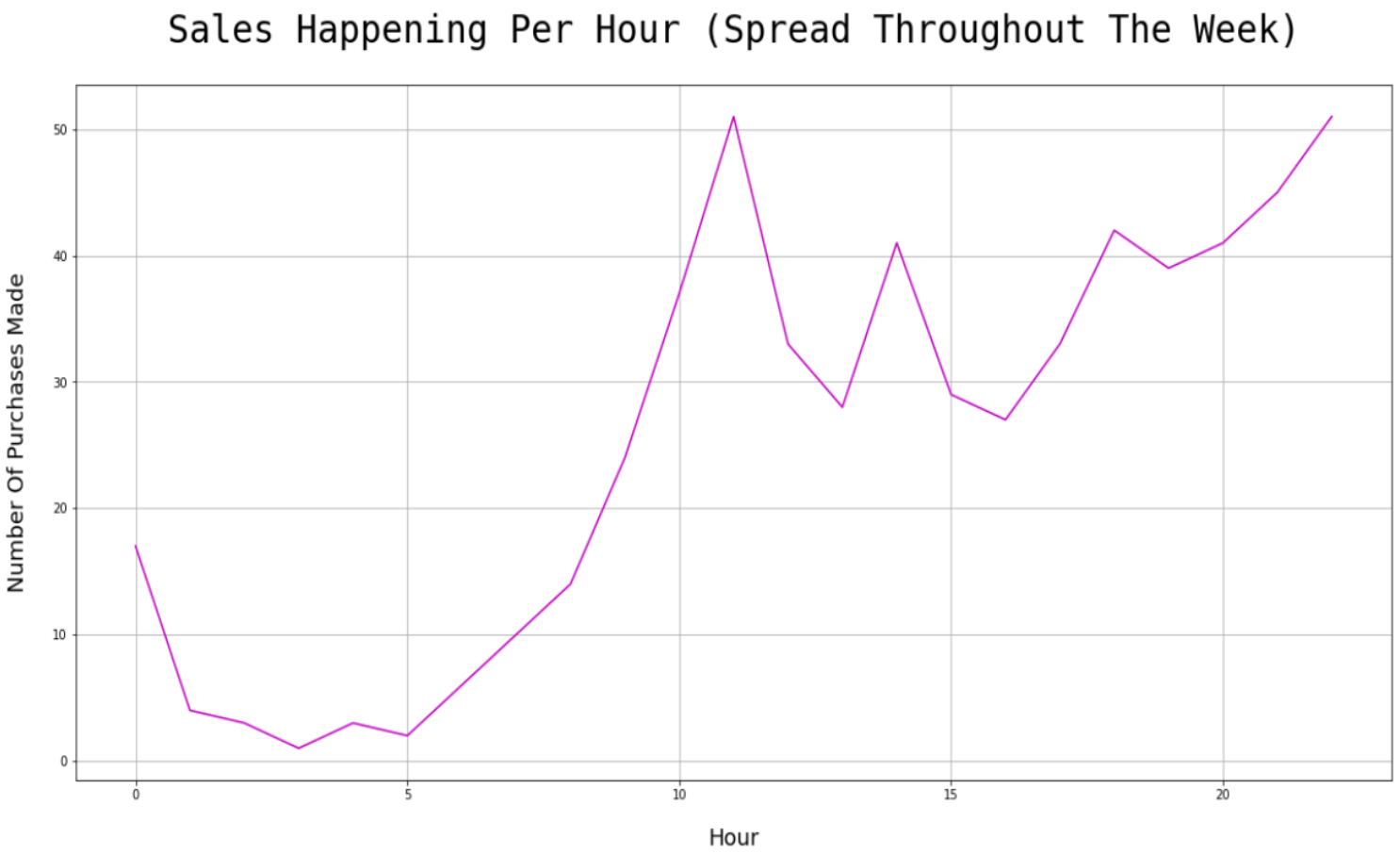
#3: Peak Times by Engagement

Peak times of engagement are a very important indicator to gauge how many effectual purchases and transactions are being completed successfully at a given point in time. It also denotes which time of the day gets the most transactional operations. If given more background information on the customers such as demographics, and their job sector, it would assist in curating particular product lists for them at a specific time of day, so that it can attract attention and may lead to potential purchases.

In this insight, we've analyzed on an hour-on-hour basis, so this gives some interesting knowledge:

Hour Of Day	Cumulative Number of Purchases
23	51
12	51
22	45
19	42
21	41
15	41
20	39
11	37
13	33
18	33
16	29
14	28
17	27
10	24
0	17
9	14
8	10
7	6
1	4
2	3
5	3
6	2
3	1





It can be seen that early hours have lesser sales and as the time approaches midnight, the number of sales (or engagement) peaks after progressively increasing for some time. This reveals to us that night-times are most voluminous in terms of engagements and it is these hours that we should target to increase sales.

For this, I'd to first convert the time to a date timestamp format, which allowed me the flexibility to make use of an hour attribute of a built-in function to extract details of hours and place it in another column as a new attribute, via which we can simply obtain the frequency count. And finally, plot it to observe the trend and pattern.

#4: Email Analysis

It is an important observation to make that most of the email IDs that have been used are of Gmail, which is suggestive that most people buy using personal mail

accounts. But, in case people make use of other email IDs other than their ones then we can analyse them to comprehend more about the organizations from which they're from. This is an important factor to understand which organizations have a majority of customers, and accordingly, keeping a segmented target for them specifically.

The organizations from which the customers are from can be divined by looking at the domain name of the email ID from which they've made the transaction. So, firstly, we need to segregate the username and the domain names. Then from the domain name we truncate all the suffixes after the dot, such as .co.in, .com, etc. These ***domain names are indicative of the organizations/company*** and can be further displayed in the form of a WordCloud.

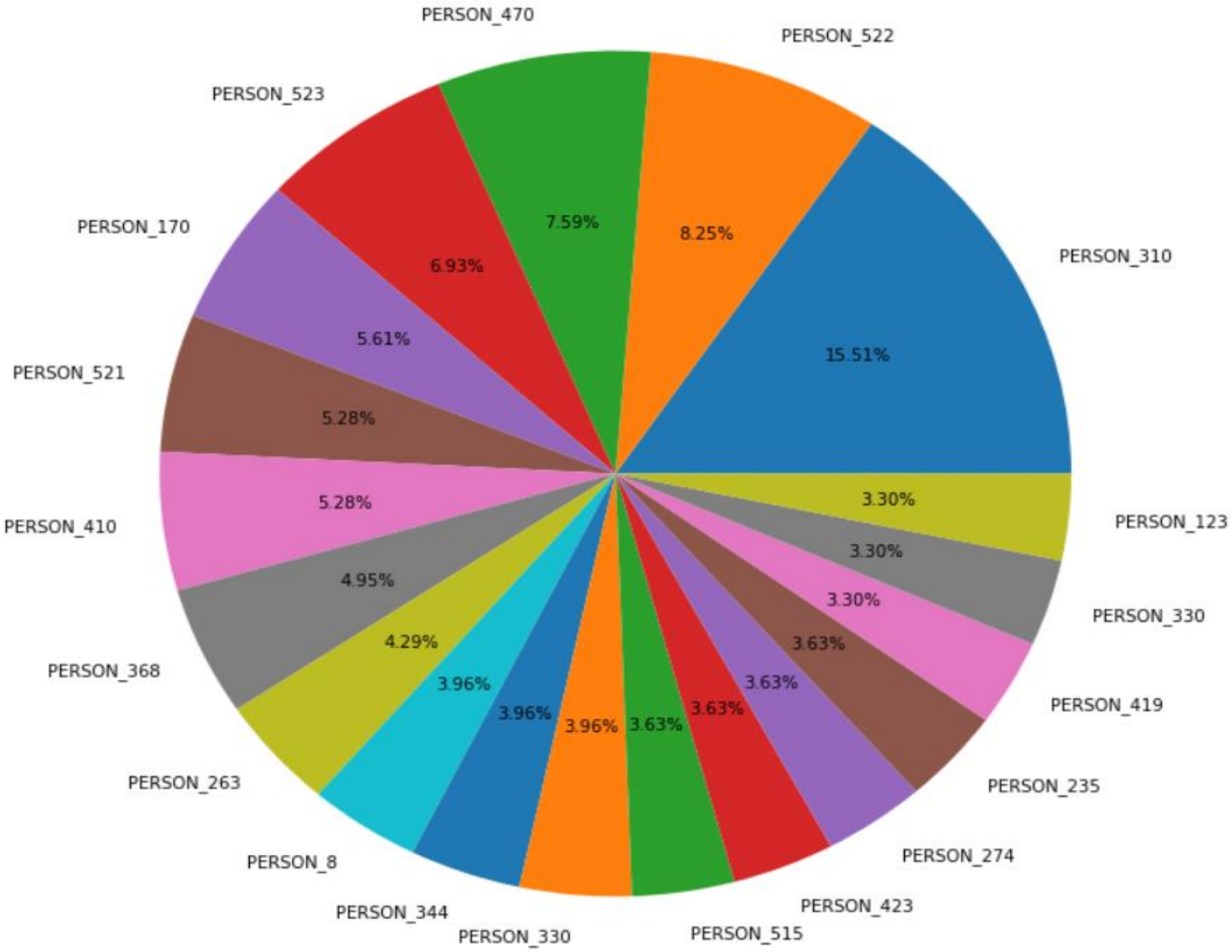
35	PERSON_35	PERSON_35@ntpc.co.in	PRODUCT_57,PRODUCT_90,PRODUCT_66,PRODUCT_58,PR...	01/03/2021 22:50:54	2021-01-03 22:50:54	22	ntpc	co.in
53	PERSON_52	PERSON_52@moirasariya.com	PRODUCT_83	02/03/2021 13:59:58	2021-02-03 13:59:58	13	moirasariya	com
60	PERSON_59	PERSON_59@labomed.in	PRODUCT_27	02/03/2021 17:32:45	2021-02-03 17:32:45	17	labomed	in
61	PERSON_60	PERSON_60@mcdermott.com	PRODUCT_86,PRODUCT_72,PRODUCT_78,PRODUCT_57,PR...	02/03/2021 17:41:33	2021-02-03 17:41:33	17	mcdermott	com
65	PERSON_64	PERSON_64@sophos.com	PRODUCT_63,PRODUCT_75	02/03/2021 19:39:35	2021-02-03 19:39:35	19	sophos	com
83	PERSON_82	PERSON_82@tatachemicals.com	PRODUCT_24,PRODUCT_57	03/03/2021 10:46:55	2021-03-03	10	tatachemicals	com

#5: Bulk Customers

Bulk customers are those that typically purchase any number of similar/dissimilar items/products in bulk in a single transaction, which means that they purchase a good deal of products at once. They constitute great profitability in a short period, so if we seek to have short-term customer retention who buy in bulk, this insight does come in very handy, when we want to dispense with several different products at the same time.

In this case, for counting the number of products purchased, within the same original data frame, I counted the number of commas in the column of the products and added 1 to it. And this count was then transferred to another new column in the data frame, and finally, it was sorted in descending order and plotted as a pie percentage chart (for the top 10), with bulk considered if several purchases exceed 10 in a transaction:

Name Of Customer	Number Of Purchases In Single Transaction
PERSON_310	47
PERSON_522	25
PERSON_470	23
PERSON_523	21
PERSON_170	17
PERSON_521	16
PERSON_410	16
PERSON_368	15
PERSON_263	13
PERSON_8	12
PERSON_344	12
PERSON_330	12
PERSON_515	11
PERSON_423	11
PERSON_274	11
PERSON_235	11
PERSON_419	10
PERSON_330	10
PERSON_123	10



#6: Recurring Customers Who Are Also Bulk Customers

It is a very valuable piece of information to understand which ***recurring customers are also bulk customers***. From what we know historically, bulk customers are those that are good for short-term gains as they buy more in a less period and recurring customers are those that buy repeatedly but may buy less at a single instance of time. A cross of both of them reveals that these customers are genuinely interested very much in our products, and we may cater to special treatment for an exalted '***Customer Lifetime Value***.'

For this dataset, we get the following recurring customers who are also bulk customers:

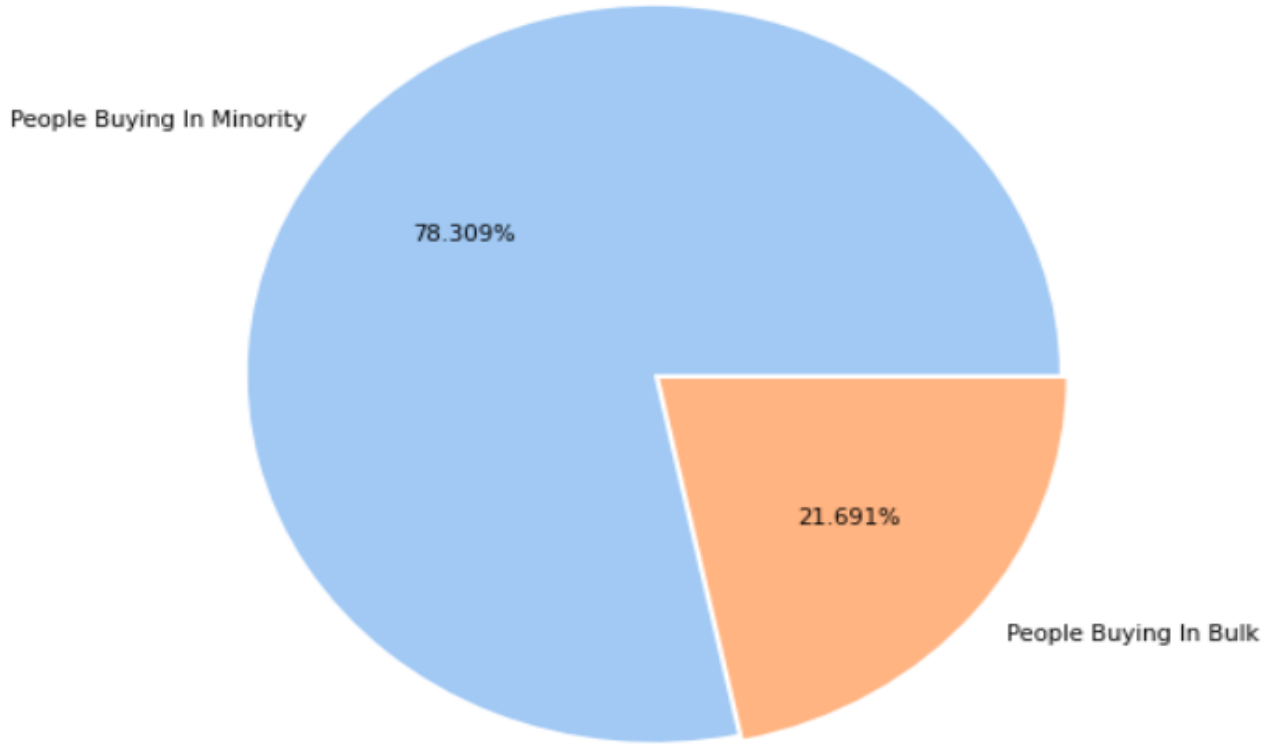
Name	Email	Product	Transaction Date	Time	Hour	0	1	Max No. Of Products In A Single Transaction	Frequency Of Purchases
RSON_470	PERSON_470@gmail.com	PRODUCT_3,PRODUCT_6,PRODUCT_47,PRODUCT_52,PROD...	07/03/2021 20:10:07	2021-07-03 20:10:07	20	gmail	com	23	5
RSON_330	PERSON_330@gmail.com	PRODUCT_57,PRODUCT_79,PRODUCT_24,PRODUCT_83,PR...	06/03/2021 19:37:56	2021-06-03 19:37:56	19	gmail	com	12	4
RSON_330	PERSON_330@gmail.com	PRODUCT_34,PRODUCT_30,PRODUCT_76,PRODUCT_26,PR...	06/03/2021 12:02:28	2021-06-03 12:02:28	12	gmail	com	10	4

For attaining this, a novel methodology was adopted. Firstly, the top 10 recurring customers' list was obtained along with their counts. Then, we merged or joined the results with that of bulk customers that were obtained in the previous insight. As a result of these two conditions, together we obtained recurring customers who also purchased in bulk.

#7: People Purchasing in Lesser Quantities (In A Single Transaction)

Whilst, it is in our interest to focus more on the heavy spenders, sometimes it is also an equally essential stat to know how many people are purchasing in lesser quantities, as it might help the company to set a future target metric to achieve in terms of increasing sales volume by sprucing up offers on bulk purchases on items for users, which they might end up eventually doing on the lucrativeness of discounts.

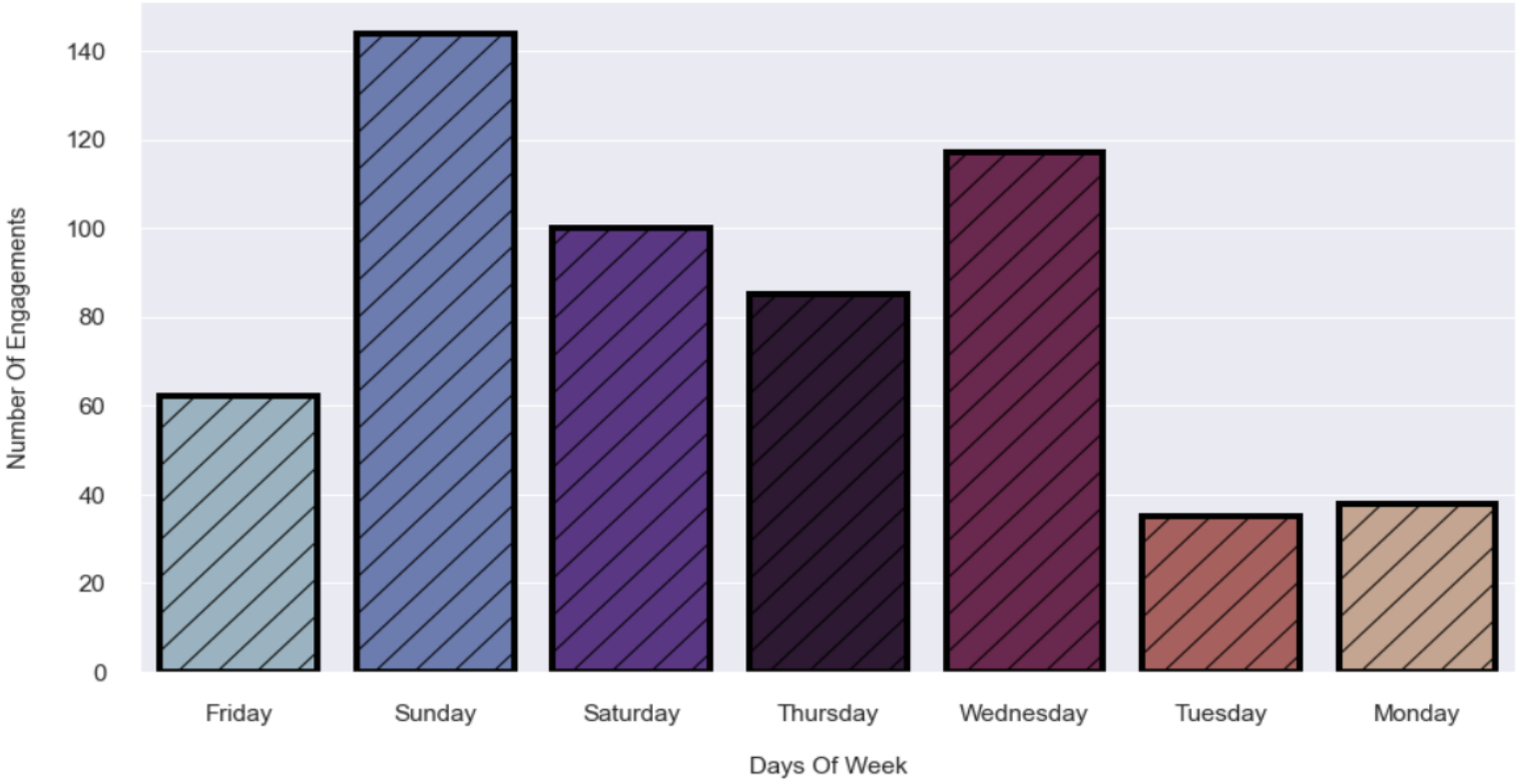
```
No. Of Distinct People Making Purchases In Lesser Quantities In Single Transaction: 426

Percentage Of People Making Minority Purchases (Single Transaction):  78.309%
```



#8: Engagement by Days of Week

When we say "***engagement by days of the week***", it implies that we're referring to the number of hits/sales that transpire on any given day in a week. Since the given dataset comprises only seven days, the trend can't be conclusively determined for a longer period. But even in this shorter period, there are some interesting nuggets of information that are imparted through thorough analysis. And if we get to know the days on which sales peak, especially, if it's any kind of holiday or special occasion, this will confirm the rational spending behaviour of customers to buy during discounts in such seasons.

	Day Of Week	No. Of Engagements
0	Sunday	144
1	Wednesday	117
2	Saturday	100
3	Thursday	85
4	Friday	62
5	Monday	38
6	Tuesday	35



Before finding out the count of purchases by the day of the week, it is important to realize that we've not been presented with the time as we want it to. So first we need to convert into a datetime format in YYYY- MM- DD HH:MiMi: SS. Only when it is formatted properly, can we apply the built-in datetime function in pandas to find the day of the week from the timestamp and maintain a copy in a separate column in the same data frame. And after that, it's just a matter of simply counting the number of purchases made.

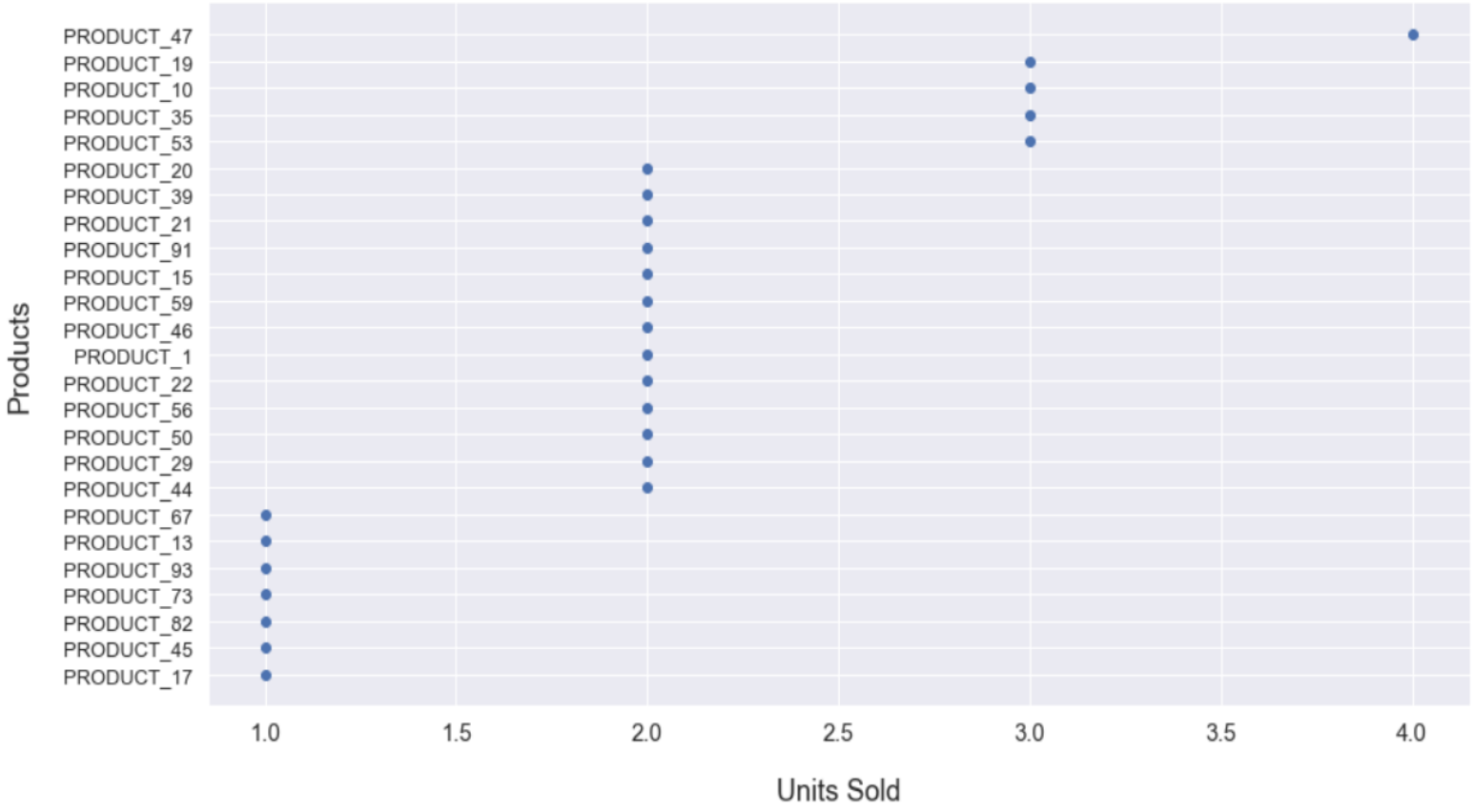
#9: Least Sold Products

One of the most important analyses in sales has to be to gather information about which products didn't perform up to the mark in terms of the number of sales accrued. This is an important indicator as this tells companies which products are not receiving much attention. It might either be due to poor quality, negative feedback, improper publicity, or any number of such allied reasons. Hence, this insight is specially meant to upgrade these products or make them

more visible/marketable in order to boost their sales. Here, we've taken those products which have sold for less than 5 times.

Product Name	Units Sold (Number/Quantity)
PRODUCT_17	1
PRODUCT_45	1
PRODUCT_82	1
PRODUCT_73	1
PRODUCT_93	1
PRODUCT_13	1
PRODUCT_67	1
PRODUCT_44	2
PRODUCT_29	2
PRODUCT_50	2
PRODUCT_56	2
PRODUCT_22	2
PRODUCT_1	2
PRODUCT_46	2
PRODUCT_59	2
PRODUCT_15	2
PRODUCT_91	2
PRODUCT_21	2
PRODUCT_39	2
PRODUCT_20	2
PRODUCT_53	3
PRODUCT_35	3
PRODUCT_10	3
PRODUCT_19	3
PRODUCT_47	4

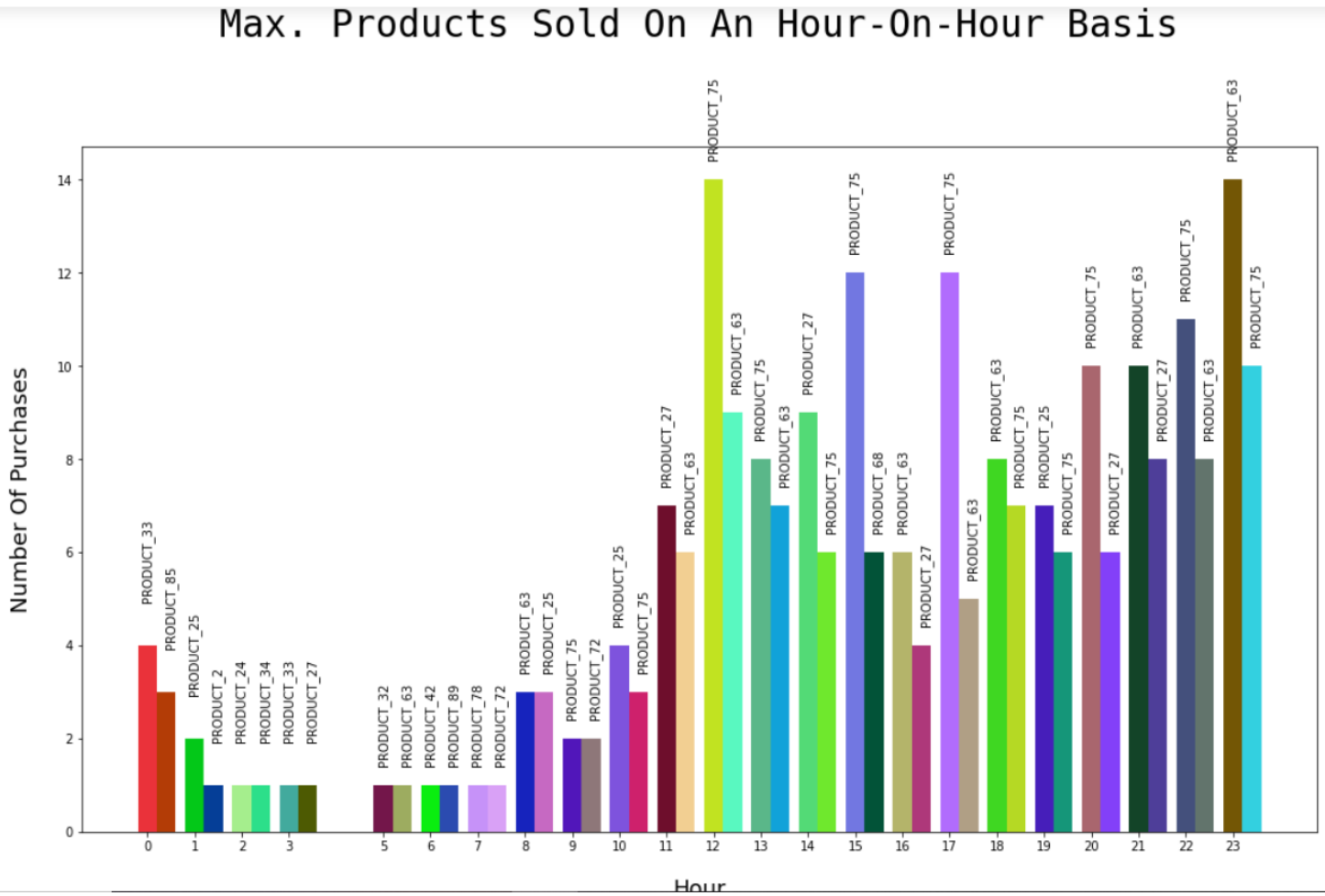
Percentage Of Products Sold Less Than 5 Times: 33.684%



#10: Top ‘N’ Selling Products on An Hour-On-Hour Basis

This insight, as is very suggestive, rates the highest selling products on an hourly basis. This way, we can estimate which products are popular at a specific time of the day, and keep encouraging users to buy it (through push notifications) or other strategies.

	Product	Hour	Count of Hourly Purchases
892	PRODUCT_33	0	4
914	PRODUCT_85	0	3
68	PRODUCT_25	1	2
937	PRODUCT_2	1	1
124	PRODUCT_24	2	1
69	PRODUCT_34	2	1
125	PRODUCT_33	3	1
126	PRODUCT_27	3	1
481	PRODUCT_32	5	1
482	PRODUCT_63	5	1
128	PRODUCT_42	6	1
315	PRODUCT_89	6	1
483	PRODUCT_78	7	1
484	PRODUCT_72	7	1
949	PRODUCT_63	8	3
673	PRODUCT_25	8	3
131	PRODUCT_75	9	2
133	PRODUCT_72	9	2
495	PRODUCT_25	10	4
685	PRODUCT_75	10	3
337	PRODUCT_27	11	7
142	PRODUCT_63	11	6
356	PRODUCT_75	12	14
39	PRODUCT_75	18	7
447	PRODUCT_25	19	7
250	PRODUCT_75	19	6
549	PRODUCT_75	20	10
1161	PRODUCT_27	20	6
1212	PRODUCT_63	21	10
555	PRODUCT_27	21	8
874	PRODUCT_75	22	11
470	PRODUCT_63	22	8
300	PRODUCT_63	23	14
880	PRODUCT_75	23	10
18	PRODUCT_63	12	9
370	PRODUCT_75	13	8
368	PRODUCT_63	13	7
23	PRODUCT_27	14	9
393	PRODUCT_75	14	6
407	PRODUCT_75	15	12
93	PRODUCT_68	15	6
785	PRODUCT_63	16	6
424	PRODUCT_27	16	4
1079	PRODUCT_75	17	12
1081	PRODUCT_63	17	5
542	PRODUCT_63	18	8



#11: Top 'N' Selling Products on Weekday Basis

This insight is same as that of the previous one except here we check for active weekdays in which products were sold.

	Product	Week Day	Count of Weekday Purchases
44	PRODUCT_63	Monday	13
1	PRODUCT_75	Monday	10
80	PRODUCT_75	Tuesday	10
121	PRODUCT_27	Tuesday	7
301	PRODUCT_63	Wednesday	29
207	PRODUCT_75	Wednesday	29
468	PRODUCT_75	Thursday	24
325	PRODUCT_63	Thursday	14
634	PRODUCT_75	Friday	10
530	PRODUCT_34	Friday	8
759	PRODUCT_12	Saturday	15
752	PRODUCT_75	Saturday	15
988	PRODUCT_75	Sunday	23
1198	PRODUCT_12	Sunday	21

#12: Top ‘N’ Customers Based on Weekday Basis

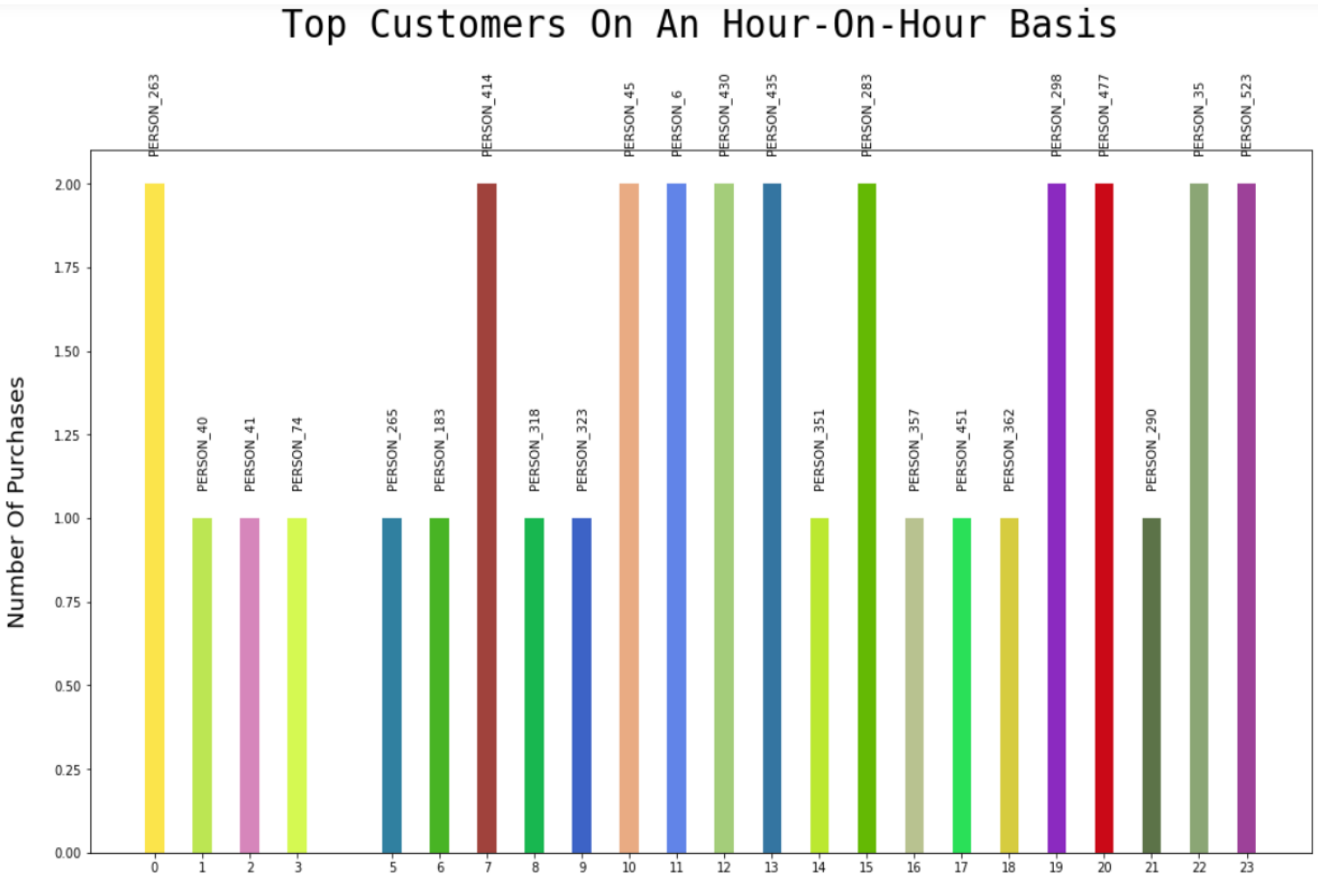
This insight seeks to find customers who were overly active on which days of the week and accordingly we can prepare a curated list to ping them with personalized offers on those days of the week, as this insight is steeped in the philosophy of customer retention:

	Name	Week Day	Count of Weekday Purchases (For Customers)
5	PERSON_6	Monday	4
7	PERSON_8	Monday	4
56	PERSON_55	Tuesday	7
54	PERSON_53	Tuesday	7
106	PERSON_103	Wednesday	6
109	PERSON_106	Wednesday	6
216	PERSON_207	Thursday	10
209	PERSON_200	Thursday	10
312	PERSON_297	Friday	5
313	PERSON_298	Friday	5
414	PERSON_383	Saturday	11
415	PERSON_384	Saturday	11
560	PERSON_508	Sunday	12
565	PERSON_512	Sunday	12

#13: Top ‘N’ Customers Based on Hour-On-Hour Basis

This is similar to the previous insight, with the key difference being that here, we see on an hourly basis:

	Name	Hour	Count of Hourly Purchases (For Customers)
438	PERSON_263	0	2
40	PERSON_40	1	1
41	PERSON_41	2	1
75	PERSON_74	3	1
278	PERSON_265	5	1
191	PERSON_183	6	1
449	PERSON_414	7	2
341	PERSON_318	8	1
346	PERSON_323	9	1
45	PERSON_45	10	2
5	PERSON_6	11	2
465	PERSON_430	12	2
472	PERSON_435	13	2
378	PERSON_351	14	1
383	PERSON_283	15	2
386	PERSON_357	16	1
494	PERSON_451	17	1
391	PERSON_362	18	1
313	PERSON_298	19	2
527	PERSON_477	20	2
326	PERSON_290	21	1



#14: Dominant/Supplement Product Pair:

Dominant/supplement product pair refers to pair of products that have dominance/meekness over other products. What this roughly means is that if let's say we buy a product 'A' along with product 'X' but not necessarily vice versa, and the other customer does the same thing, then it implies that there's a great probability that when any customer buys product 'X', then he/she will also buy product 'A'. Here, product 'X' becomes the dominant and 'A' becomes the supplement product.

So, in data mining, we've four concepts, namely, ***support, confidence, lift, conviction***, etc., as measures for product affinity.

Now, each row in the dataset is a ***transaction*** and each entry in the product column of it becomes an ***itemset***.

Support simply emphasizes ***how popular an item set is***. It is commonly used to determine the strength of association between items. ***Confidence*** denotes the ***likelihood of certain items*** being ***purchased together***. They are typically expressed as:

$$Support = \frac{freq(i_1, i_2)}{N} \qquad P(butter \mid bread) = \frac{Support(Bread, Butter)}{P(Bread)}$$

Bread & butter are sample data

Lift and Conviction are allied metrics based on confidence and support and can be expressed in the following way:

$$Lift(Bread \Rightarrow Butter) = \frac{Support(Bread, Butter)}{Support(Bread) * Support(Butter)} = \frac{P(Bread, Butter)}{P(Bread) * P(Butter)}$$

Definition of Lift

$$Conviction(Bread \Rightarrow Butter) = \frac{1 - Support(Butter)}{1 - Confidence(Bread \Rightarrow Butter)}$$

So, I've construed a simple algorithm that finds the relationship:

In [35]:
```
def frequency_items (x,y):
    fx_=sum([x in i for i in arr3])
    fy_=sum([y in i for i in arr3])

    fxy_=sum([all(z in i for z in [x,y]) for i in arr3])

    support=fxy_/len(arr3)
    confidence = support/(fx_/len(arr3))
    lift =confidence /(fy_/len(arr3))
    if confidence ==1:
        conviction = 0
    else:
        conviction=(1-(fy_/len(arr3)))/(1-confidence)

    print("Support = {}".format(round(support,2)))
    print("Confidence = {}".format(round(confidence,2)))
    print("Lift= {}".format(round(lift,2)))
    print("Conviction={}".format(round(conviction,2)))
```

In [36]:
```
frequency_items('PRODUCT_27','PRODUCT_63')
```

```
Support = 0.05
Confidence = 0.37
Lift= 2.21
Conviction=1.32
```

The above result means that only 5% of the transactions have both products in them. **PRODUCT_63** appears in 37% of the transactions where **PRODUCT_27** occurs. The value of the lift is much greater than 1, this means that these two products are very much likely to be purchased together.

Accordingly, we can take custom input for other product pairs and find the relationship.

#15: Market Basket Analysis

Market basket analysis is just an extension of the previously discussed insight, wherein we explore a wide plethora of correlations between different market products in a basket (hence, the name). Making use of specialized ML libraries

like ***mlxtend's frequent_patterns*** and ***preprocessing*** help us ease the process for a large dataset and formalize results properly.

Firstly, we apply a transformation to the dataset and then the apriori algorithm for itemsets. Then, we can view metrics such as confidence, lift, support and conviction, for all the pairs of products.

Out[38]:

	PRODUCT_1	PRODUCT_10	PRODUCT_11	PRODUCT_12	PRODUCT_13	PRODUCT_14	PRODUCT_15	PRODUCT_16	PRODUCT_17	PRODUCT_18	...	PRC
0	False	False	False	False	False	False	False	False	False	False	...	
1	False	False	False	False	False	False	False	False	False	False	...	
2	False	False	False	False	False	False	False	False	False	False	...	
3	False	False	False	False	False	False	False	False	False	False	...	
4	False	False	False	False	False	False	False	False	False	False	...	
...	...	...	...	...	...	...	...	...	...	...	...	
576	False	False	True	False	False	False	False	False	False	False	...	
577	False	False	False	False	False	False	False	False	False	False	...	
578	False	False	False	True	False	True	False	False	False	False	...	
579	False	False	False	False	False	False	False	False	False	False	...	
580	False	False	False	False	False	False	False	False	False	False	...	

581 rows × 95 columns

Encoded data after transformation (above)

Out[39]:

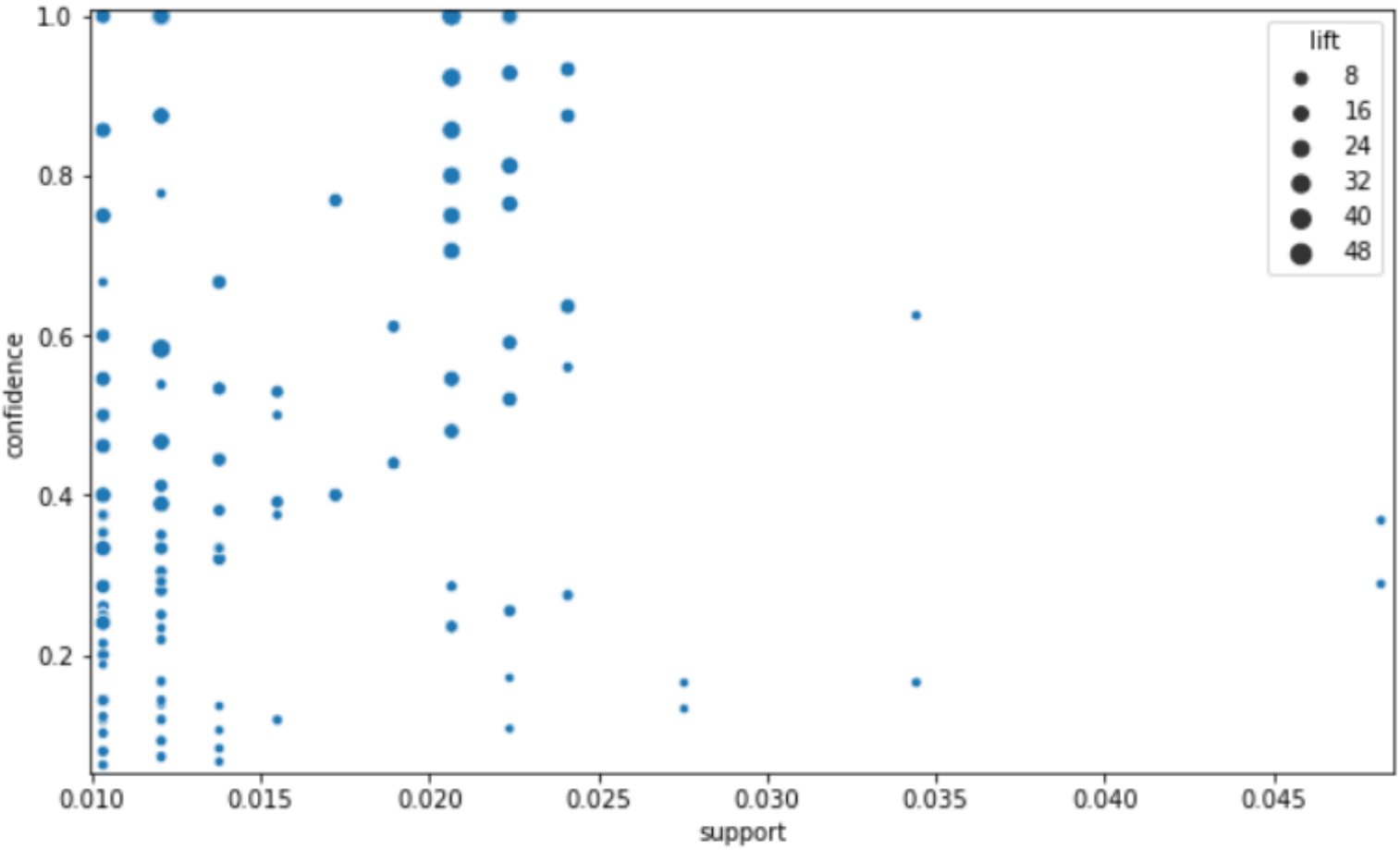
	support	itemsets
0	0.013769	(PRODUCT_11)
1	0.087780	(PRODUCT_12)
2	0.010327	(PRODUCT_14)
3	0.020654	(PRODUCT_2)
4	0.039587	(PRODUCT_24)
5	0.101549	(PRODUCT_25)
6	0.029260	(PRODUCT_26)
7	0.130809	(PRODUCT_27)
8	0.025818	(PRODUCT_28)
9	0.012048	(PRODUCT_3)
10	0.025818	(PRODUCT_30)
11	0.034423	(PRODUCT_32)
12	0.034423	(PRODUCT_33)
13	0.072289	(PRODUCT_34)
14	0.017212	(PRODUCT_37)
15	0.010327	(PRODUCT_38)
16	0.025818	(PRODUCT_4)
17	0.010327	(PRODUCT_40)
18	0.020654	(PRODUCT_42)

Out[40]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PRODUCT_28)	(PRODUCT_12)	0.025818	0.087780	0.020654	0.800000	9.113725	0.018388	4.561102
1	(PRODUCT_12)	(PRODUCT_28)	0.087780	0.025818	0.020654	0.235294	9.113725	0.018388	1.273931
2	(PRODUCT_43)	(PRODUCT_12)	0.024096	0.087780	0.020654	0.857143	9.764706	0.018539	6.385542
3	(PRODUCT_12)	(PRODUCT_43)	0.087780	0.024096	0.020654	0.235294	9.764706	0.018539	1.276182
4	(PRODUCT_49)	(PRODUCT_12)	0.024096	0.087780	0.020654	0.857143	9.764706	0.018539	6.385542
5	(PRODUCT_12)	(PRODUCT_49)	0.087780	0.024096	0.020654	0.235294	9.764706	0.018539	1.276182
6	(PRODUCT_12)	(PRODUCT_55)	0.087780	0.027539	0.024096	0.274510	9.968137	0.021679	1.340420
7	(PRODUCT_55)	(PRODUCT_12)	0.027539	0.087780	0.024096	0.875000	9.968137	0.021679	7.297762
8	(PRODUCT_12)	(PRODUCT_61)	0.087780	0.037866	0.022375	0.254902	6.731729	0.019051	1.291285
9	(PRODUCT_61)	(PRODUCT_12)	0.037866	0.087780	0.022375	0.590909	6.731729	0.019051	2.229872
10	(PRODUCT_62)	(PRODUCT_12)	0.029260	0.087780	0.022375	0.764706	8.711649	0.019807	3.876936
11	(PRODUCT_12)	(PRODUCT_62)	0.087780	0.029260	0.022375	0.254902	8.711649	0.019807	1.302835
12	(PRODUCT_12)	(PRODUCT_63)	0.087780	0.166954	0.012048	0.137255	0.822114	-0.002607	0.965577
13	(PRODUCT_63)	(PRODUCT_12)	0.166954	0.087780	0.012048	0.072165	0.822114	-0.002607	0.983171
14	(PRODUCT_74)	(PRODUCT_12)	0.084337	0.087780	0.010327	0.122449	1.394958	0.002924	1.039507
15	(PRODUCT_12)	(PRODUCT_74)	0.087780	0.084337	0.010327	0.117647	1.394958	0.002924	1.037751
16	(PRODUCT_85)	(PRODUCT_12)	0.043029	0.087780	0.024096	0.560000	6.379608	0.020319	2.073228
17	(PRODUCT_12)	(PRODUCT_85)	0.087780	0.043029	0.024096	0.274510	6.379608	0.020319	1.319068
18	(PRODUCT_24)	(PRODUCT_26)	0.039587	0.029260	0.015491	0.391304	13.373402	0.014332	1.594787
19	(PRODUCT_26)	(PRODUCT_24)	0.029260	0.039587	0.015491	0.529412	13.373402	0.014332	2.040878

Out[41]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PRODUCT_28)	(PRODUCT_12)	0.025818	0.087780	0.020654	0.800000	9.113725	0.018388	4.561102
1	(PRODUCT_12)	(PRODUCT_28)	0.087780	0.025818	0.020654	0.235294	9.113725	0.018388	1.273931
2	(PRODUCT_43)	(PRODUCT_12)	0.024096	0.087780	0.020654	0.857143	9.764706	0.018539	6.385542
3	(PRODUCT_12)	(PRODUCT_43)	0.087780	0.024096	0.020654	0.235294	9.764706	0.018539	1.276182
4	(PRODUCT_49)	(PRODUCT_12)	0.024096	0.087780	0.020654	0.857143	9.764706	0.018539	6.385542
5	(PRODUCT_12)	(PRODUCT_49)	0.087780	0.024096	0.020654	0.235294	9.764706	0.018539	1.276182
6	(PRODUCT_12)	(PRODUCT_55)	0.087780	0.027539	0.024096	0.274510	9.968137	0.021679	1.340420
7	(PRODUCT_55)	(PRODUCT_12)	0.027539	0.087780	0.024096	0.875000	9.968137	0.021679	7.297762
8	(PRODUCT_12)	(PRODUCT_61)	0.087780	0.037866	0.022375	0.254902	6.731729	0.019051	1.291285
9	(PRODUCT_61)	(PRODUCT_12)	0.037866	0.087780	0.022375	0.590909	6.731729	0.019051	2.229872
10	(PRODUCT_62)	(PRODUCT_12)	0.029260	0.087780	0.022375	0.764706	8.711649	0.019807	3.876936
11	(PRODUCT_12)	(PRODUCT_62)	0.087780	0.029260	0.022375	0.254902	8.711649	0.019807	1.302835
12	(PRODUCT_12)	(PRODUCT_63)	0.087780	0.166954	0.012048	0.137255	0.822114	-0.002607	0.965577
13	(PRODUCT_63)	(PRODUCT_12)	0.166954	0.087780	0.012048	0.072165	0.822114	-0.002607	0.983171
14	(PRODUCT_74)	(PRODUCT_12)	0.084337	0.087780	0.010327	0.122449	1.394958	0.002924	1.039507
15	(PRODUCT_12)	(PRODUCT_74)	0.087780	0.084337	0.010327	0.117647	1.394958	0.002924	1.037751
16	(PRODUCT_85)	(PRODUCT_12)	0.043029	0.087780	0.024096	0.560000	6.379608	0.020319	2.073228
17	(PRODUCT_12)	(PRODUCT_85)	0.087780	0.043029	0.024096	0.274510	6.379608	0.020319	1.319068
18	(PRODUCT_24)	(PRODUCT_26)	0.039587	0.029260	0.015491	0.391304	13.373402	0.014332	1.594787
19	(PRODUCT_26)	(PRODUCT_24)	0.029260	0.039587	0.015491	0.529412	13.373402	0.014332	2.040878



From the results obtained above, we can view the product pairs that are viable to be sold together in a combined basket form. For example, if we take **PRODUCT_49** & **PRODUCT_12**, they have a confidence of 85%, which denotes strong bonding betwixt them. Likewise, we can analyze the top 'N' such products, and form offers and branding strategies to boost their sales.

Conclusion

Hence, through this project's medium, I was able to successfully derive and conclude a certain number of insights from the data presented by the company. As a result of constant nitty-gritty researching and a hands-on approach that I undertook, I learnt a lot about the domain that I'd previously not known before. The practicality of this work has impressed upon me the real-time and real-life work scenario and the potential applications and use case of technology to reap the benefits of the knowledge base.

Appendix (Code Link):

https://github.com/pragmatic-philosopher09/Tata-Steel-Summer-Internship--Data-Science-