Spend Analytics

Capstone Project [DSP-55; team -2]

[Gaurav Singh, Stuti Thakkar, Dolly Dubey, Tanmay Pendse, Yash Upadhyay]

3/15/22

Data Science Pro Degree

Introduction

Spend analysis is the process of reviewing current and historical spending. The goal of the exercise is to reduce cost, improve strategic sourcing, and increase the efficiency of spend management. An analysis requires spend data processed into KPIs and metrics and then visualized to show patterns. The dataset provided to us was the spend procurement of a poultry-based company that needs to be analysed to save the cost for future purchases.

Spend analysis has three main parts:

1. **Spend Visibility**– Having clean spend data as well as KPIs and other metrics as a way to see spending from many points of view.
2. **Spend Analysis**– Asking questions about corporate spending and procurement, finding the answers in the metrics, and creating ways to reduce costs and improve results.
3. **Procurement Process Improvement**– Taking the results of the analysis and implementing changes to improve future performance meeting corporate goals.

Data Gathering and Processing for Spend Analytics

1. **Identify all procurement and sourcing-related data sources:**
   Data sources can include general ledgers, ERP systems, e-procurement software, expense systems, P-cards, etc. Collect spend data from everywhere-all departments, business units, and manufacturing plants. Remember, it's valuable to analyse both direct and indirect spending.

2. **Gather the data into one, main location:**
   Compiling data can prove difficult since the data commonly is in different formats, different currencies, or different languages. Specifically designed extract, transform, load (ETL) procedures exist to overcome these issues. A Spend Analytics solution handles the variances and ETL automatically. When using spreadsheets, procurement analysts need to process the differences themselves or use a data management tool.

3. **Cleanse data for more accurate processing:**
   Besides language and currency, product and supplier fields, such as names and descriptions, are compared and normalized to be the same. For example, three different business units may buy laptops from Dell each using a different supplier name—DELL, Dell Technologies, Dell, Inc. Standardizing sourcing data makes it easier for companies (and machines and algorithms) to interpret the data.

4. **Enrich data for complete entries and additional metrics:**

Data coming from varied sources will have different fields potentially causing issues when they are brought together. Common problems include missing specific fields, abbreviations, and misspellings. Smartly combining the data generates more complete entries for each item. Include outside data sources to enrich data for more ways to analyse, for example, industry codes, supplier diversity status, and ISO certifications.

5. **Categorize items and materials into logical hierarchies:**
   Having all spending data in a unifying taxonomy allows procurement professionals to understand and track where the money is being spent. There are existing categorization standards, such as UNSPSC (United Nations Standard Products and Services Code), NAICS (North American Industry Classification System), or eClass. Regardless if a company uses its own classification system or variants of existing ones, all spend must be accurately categorized including marketing, travel, office supplies, and legal services. The deeper the categorization, the more granular and informative the spend analytics can be. Classification can be a tedious, detailed, months-long task. Using Spend Analytics software with an AI-powered classification engine can speed that process to days and categorize 60-70% of the first pass data automatically. After human review and correction, the system learns and improves, classifying a higher percentage each time. The ROI on analysis solutions like these can be nearly immediate – the savings in time and resources alone are tremendous.

The Power of Spend Analysis

As spend analysis moves from a daunting months-long project to a task that takes a week, or a few days, the benefits to organizations increase tremendously. The following are reasons to use spend analysis and some new value gained when making it a regular procurement activity.

1. **Improve Data Quality:**
   The first part of a spend analysis involves gathering, cleaning, normalizing, and enriching the ALL the purchasing data. If you only use a subset of the data, you limit your review and ability to get useful results. Using data that is not scrubbed means the analysis will have duplicate items and suppliers, preventing paths for consolidation of suppliers. Remember, Garbage In = Garbage Out.

2. **Increase Opportunities to Save:**
   When you have all the data clean, you have a firm base to find trends, measure KPIs, and benchmark performance. The data needs to be classified, the deeper, the better to highlight these trends and actionable insights.

Documentation

1. We will start by importing the libraries we will require for performing the Data analysis and drawing conclusions, trends and patterns for the same. These include NumPy, Pandas, Matplotlib and Seaborn.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

2. Loading the data set into python using pandas. Pandas is a very useful and powerful tool for data analysis that provides various features to analyze the data in a very simple and in detail manner.

```
df = pd.read_excel("D:/Imarticus lecture Files/Capstone Project Folder/KPMG Data_Spend Analytics 2.xlsx")
df
```

3. Calculating the number of null values in the data set so that we can impute those values to avoid any kind of errors in the analysis of the data.

```
#To display the na in all the columns of the data frame
pd.set_option('display.max_rows',65)
```

```
df.isna().sum()
```

4. Dropping the columns that are not required for the Analysis purpose so that they don't create discrepancy in the analysis. Checking the shape of the data set to see if the task is performed as required. These following columns have been removed from the dataset as it has more than 95% of values being null which cannot be computed for further analysis.

```
df.drop(columns=["Conv.","GRT","DCI","Agr. Cum. Qty","TOZ","Quantity",
                 "Cat","Net value","Object no.","Time of Transmission",
                 "Next Transmission Number","Itm","Itm.1",
                 "Requirement Urgency","CRM  Item No",
                 "Down Payment Amount","Item.2"], inplace= True, axis =1)
```

```
df
```

5. These are the further columns that are eliminated from the dataset to get an added advantage into data analysis and finding trends and patterns.

These columns have been eliminated because all of these columns have only 1 unique level which does not make any difference while model building.

```
df.drop(columns=["Conv.","GRT","DCI","Agr. Cum. Qty","TOZ","Quantity",
                 "Cat","Net value","Object no.","Time of Transmission",
                 "Next Transmission Number","Itm","Itm.1","Requirement Urgency",
                 "CRM  Item No","Down Payment Amount","Item.2"], inplace= True, axis =1)
```

```
df
```

6. Finding the summary of the data set by using describe function to get the clear idea about the distribution of the dataset. Using the “Object” function will give out the summary of the categorical data too.

```
#To find the summary for entire dataframe.
# include = object gives summary for categorical data s
df.describe(include='object')
```

7. Removing the rows in these columns which have null values as computing these rows won’t be possible due to their distinct feature. Now we are left with no columns having null values.

```
df.drop(columns = "Un", inplace = True, axis = 1)
```

```
df.isna().sum()
```

```
...
```

```
df.dropna(subset=["MTyp"],inplace= True, axis=0)
```

```
df.isna().sum()
```

```
...
```

```
df.dropna(subset=["Price Date"],inplace= True, axis=0)
```

```
df.isna().sum()
```

8. Creating a heat map of all the columns to check the correlation between these columns to check which columns will affect the model building and accuracy of the model.

```
correlations = df.corr()
correlations
#df.describe()
```

```
#Plotting the
plt.figure(figsize=(12,8))
sns.heatmap(correlations)
plt.show()
```

9. These are the additional columns that have been dropped after a deep understanding of the dataset.
   The reasons for deleting these columns are as follows:
   a. Some information is not of any use to check the spend analytics for a company
   b. Some columns have only two levels out of which one level is a dominant one.
   c. Some columns have approximately 98% of zeros present in them which makes the column irrelevant.
   d. Some columns determine unique codes which cannot be used for analysis

```
# Here we are dropping columns based on few basic understanding.
df.drop(columns = ["SLoc","OUn","Eq. To","Per","BUn","Non-deductible",
                   "Effective value","PTm","Net Weight","Profit Ctr",
                   "Gross Weight","Volume","RShLi",
                   "NCM Code","Priority (Material Required Within)","Ordered By",
                   "Approved By","Indenter ID","Item"],
        inplace = True, axis = 1)
```

```
df.shape
```

10. Saving the cleansed data into a new file to use it for future modelling and analysis. This will save time required for loading unnecessary data.

```
#Saving the file as an excel file and from now onwards this will be used for doing any analysis.
df1 = df.to_excel("Final_data.xlsx", sheet_name = "Data")
```

```
df
```

11. Renaming the columns for an easy understanding using the rename function.

```
df_Cluster.rename(columns={"Short Text":"Items","Net Value":"Price"},inplace=True)
df_Cluster
```

```
df_Cluster_2 = df_Cluster[["Items","Price"]]
```

12. Scaling the columns to be used for clustering. This is also called as Standardization. Standardization is another scaling technique where the values are centred around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. It's done twice once with minimum, maximum scaling and once with mean standard deviation method.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_Scaled = scaler.fit_transform(df_Cluster_2)
df_Scaled
```

```
df_Scaled = pd.DataFrame(df_Scaled)
```

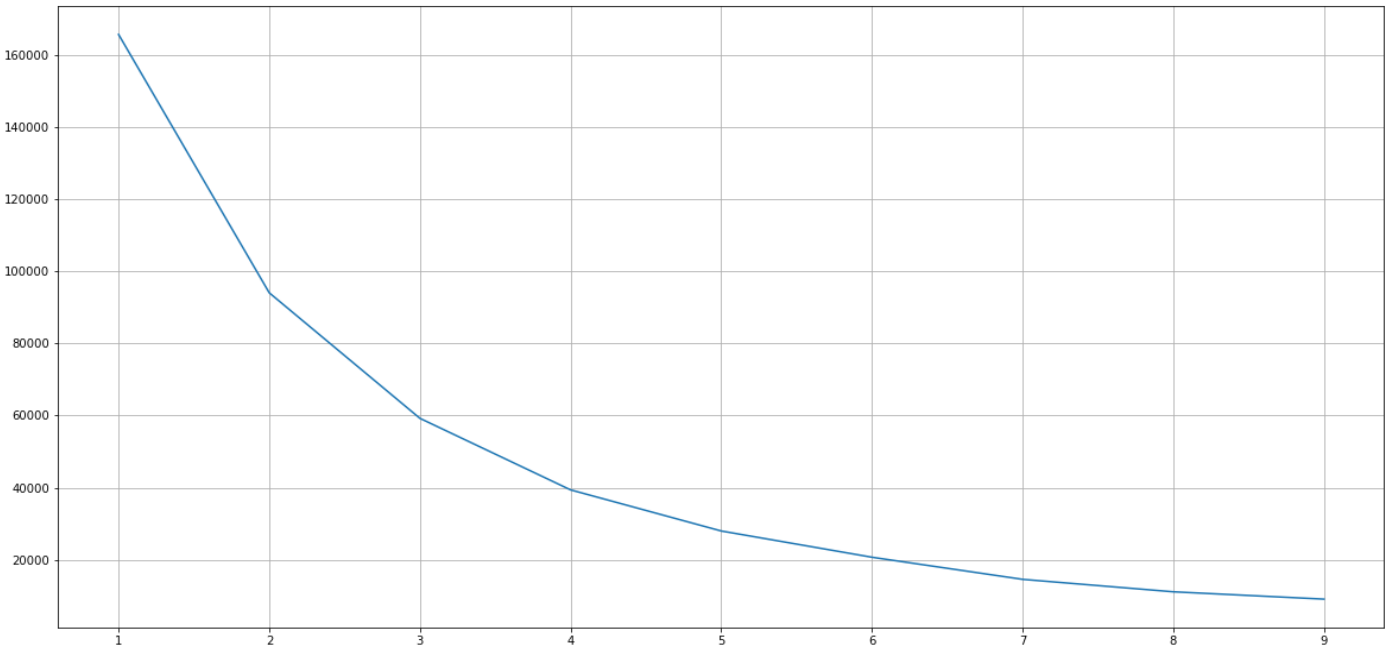
13. Importing the libraries used for K-means clustering and running the clustering algorithm to build a machine learning model. K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit_transform(df_Scaled)
print(kmeans.inertia_)
```

```
inertia = []
for k in range(1,10):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(df_Scaled)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(20,10))
plt.grid()
plt.plot(range(1,10),inertia)
plt.show()
```

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(df_Scaled)
print(kmeans.inertia_)
```



14. Importing the Silhouette Library to find the silhouette score for the K-means clustering mode.
    Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1. 1: Means clusters are well apart from each other and clearly distinguished.

```
from sklearn.metrics import silhouette_score
score = silhouette_score(df_Scaled, kmeans.labels_, metric="euclidean")
print(score)
```

15. Importing the Min-Max-Scaler library to perform the standardization process and running the k-means clustering model again to get a better accuracy score.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
principalComponent = pca.fit_transform(df_Scaled.drop(columns="Labels"))
principalComponent
```

```
...
```

```
df_pca = pd.DataFrame(data=principalComponent,
                      columns=["PCA1","PCA2"])
target = pd.Series(df_Scaled["Labels"],name='target')
result_df = pd.concat([df_pca,target],axis=1)
result_df
```

```
...
```

```
plt.scatter(result_df["PCA1"],result_df["PCA2"])
```

16. Importing PCA to cut down the number of columns to two so that we can plot this on a 2D plot diagram with X axis and Y axis.
Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.

```
# Using MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_Scaled = scaler.fit_transform(df_Cluster_2)
df_Scaled
```

```
...
```

```
df_Scaled = pd.DataFrame(df_Scaled)
df_Scaled
```

```
...
```

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(df_Scaled)
print(kmeans.inertia_)
```

17. Finding out the exact number of clusters that will be required to have the highest accuracy of the model and then plotting that on a 2D scatter plot to have a clear understanding of the data set.

```
df_Cluster["Labels"] = kmeans.labels_
df_Cluster
```

```
df_Cluster[df_Cluster["Labels"]==0].describe()
```

```
df_Cluster[df_Cluster["Labels"]==1].describe()
```

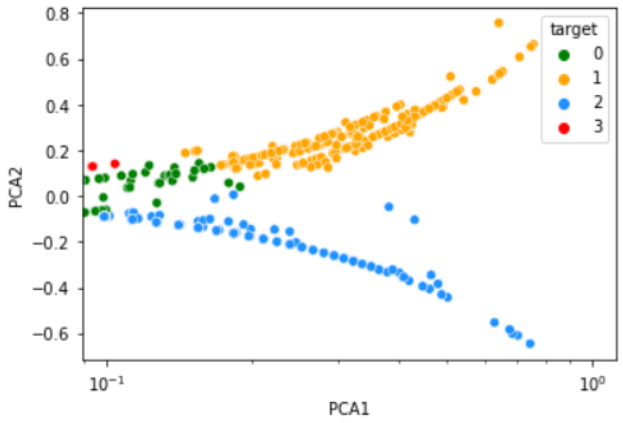
```
df_Cluster[df_Cluster["Labels"]==2].describe()
```

```
df_Cluster[df_Cluster["Labels"]==3].describe()
```

```
plt.figure(figsize=(10,8))
plt.scatter(df_Cluster["Items"],df_Cluster["Price"]);
```

18. PCA Plot of the clusters

```
c = sns.scatterplot( x="PCA1", y="PCA2", hue="target",
                     data=result_df, palette=['Green','orange','dodgerblue','red'], legend='full');
c.set(xscale="log");
```

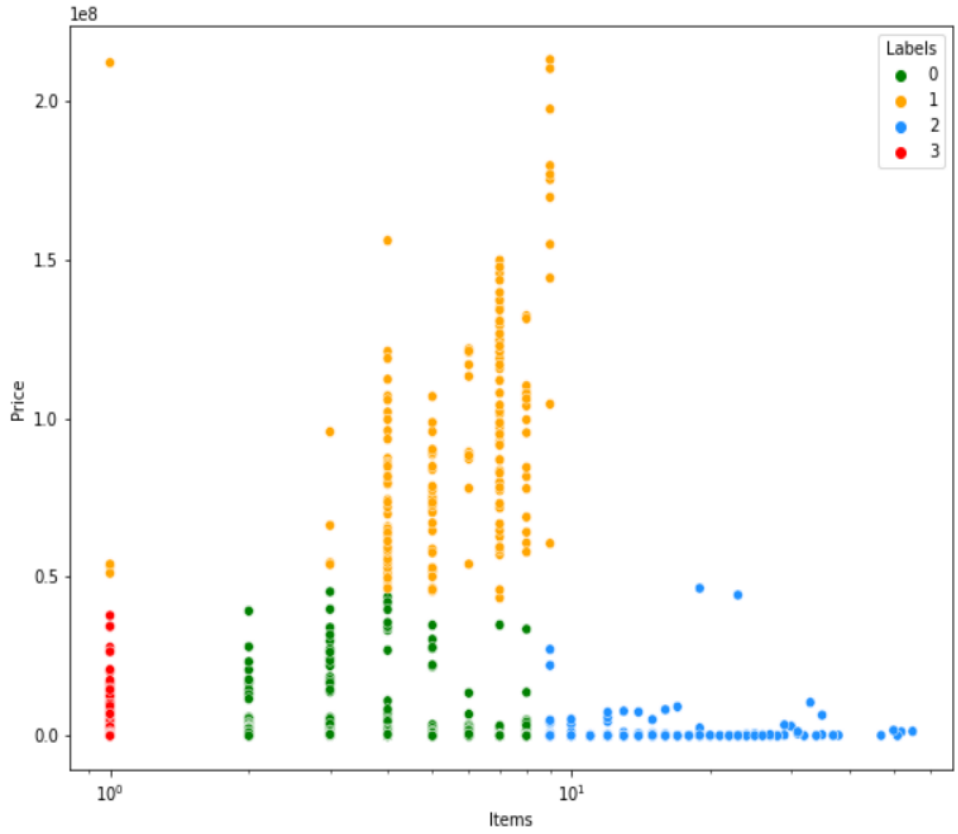


19. Cluster in original data:

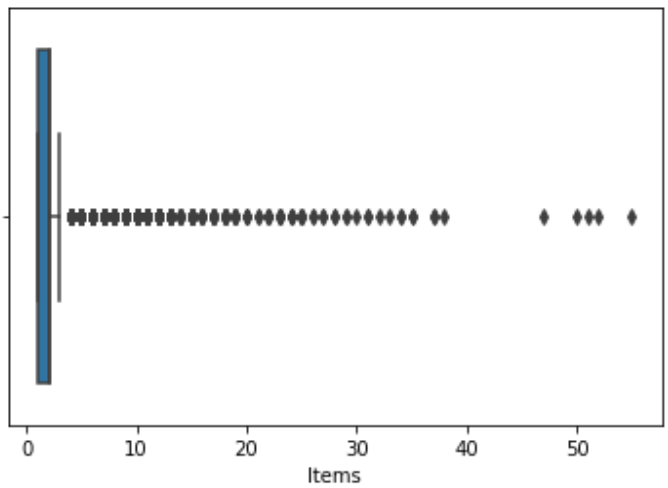
a) Red coloured is less frequent and items of less amount
b) **Green and Yellow ones are with high frequency and high amount items**
c) Blue cluster is of items of very high frequency and of very low amount
d) **We are interested into the green and yellow cluster**

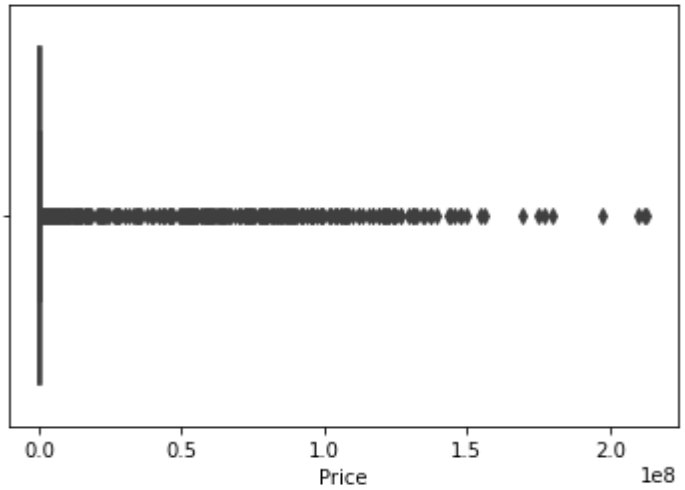
Below scatter plot shows the clusters that are formed in the original data

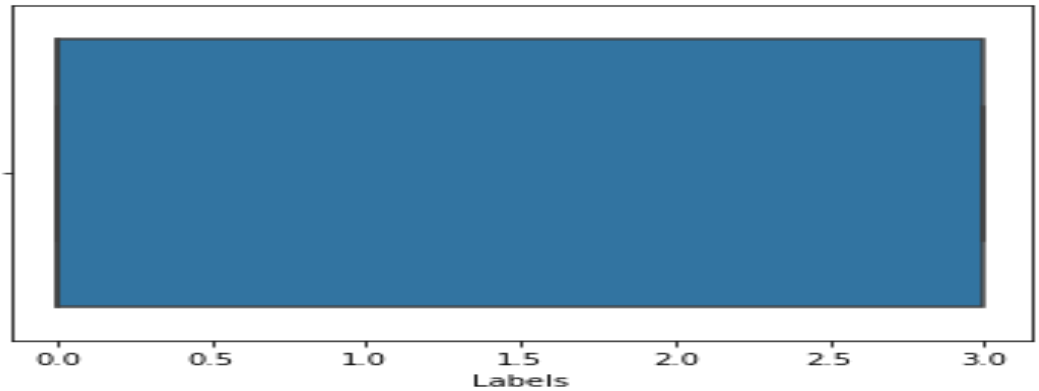
```
plt.figure(figsize=(10,8))
c = sns.scatterplot( x="Items", y="Price", hue="Labels",
                     data=df_Cluster, palette=['Green','orange','dodgerblue','red'], legend='full');
c.set(xscale="log");
```



Box plot of the same original data used for cluster are shown below





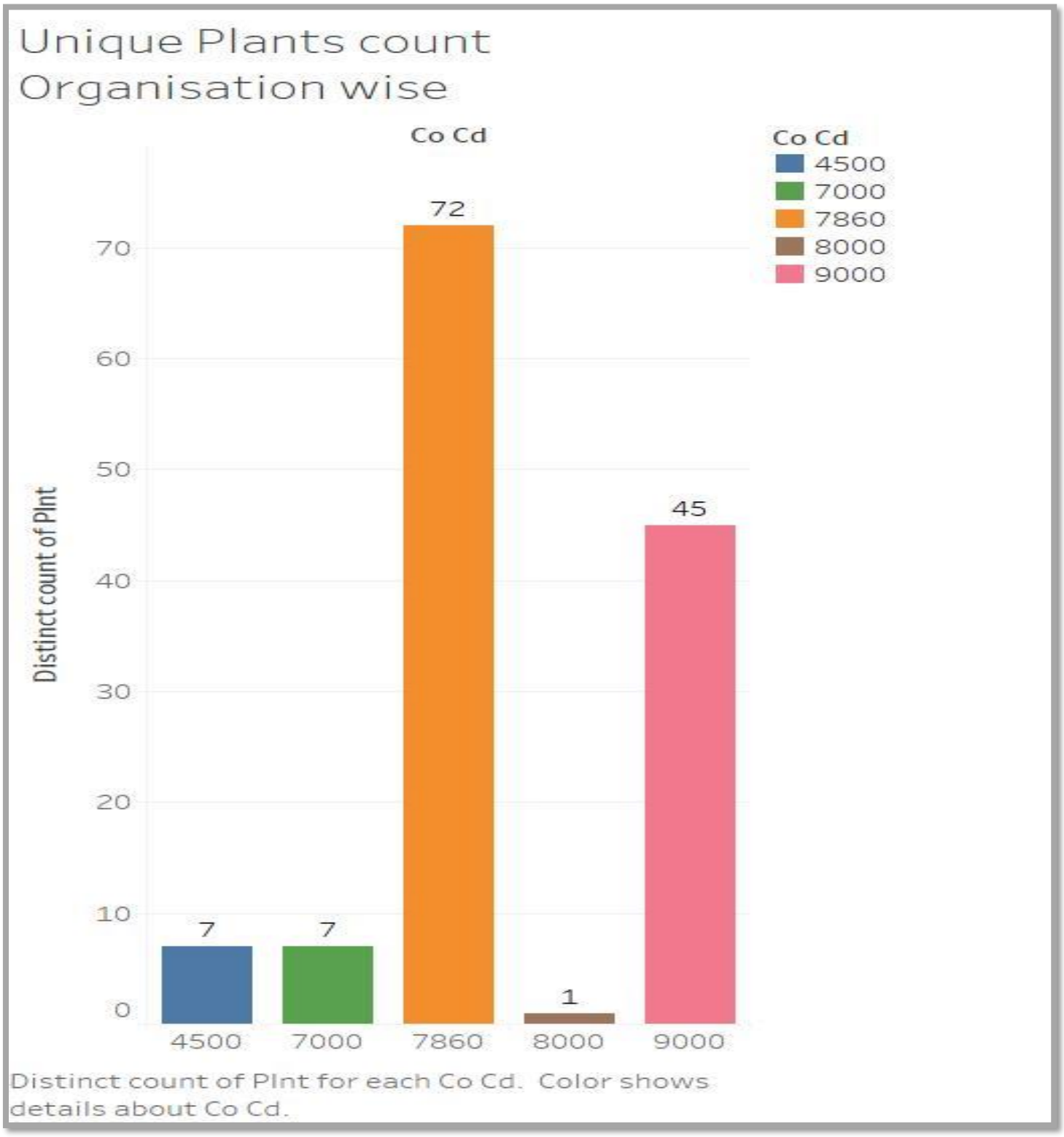


Data Analytics (Some insights that were found in the data)

All the Visualizations are taken from Tableau:

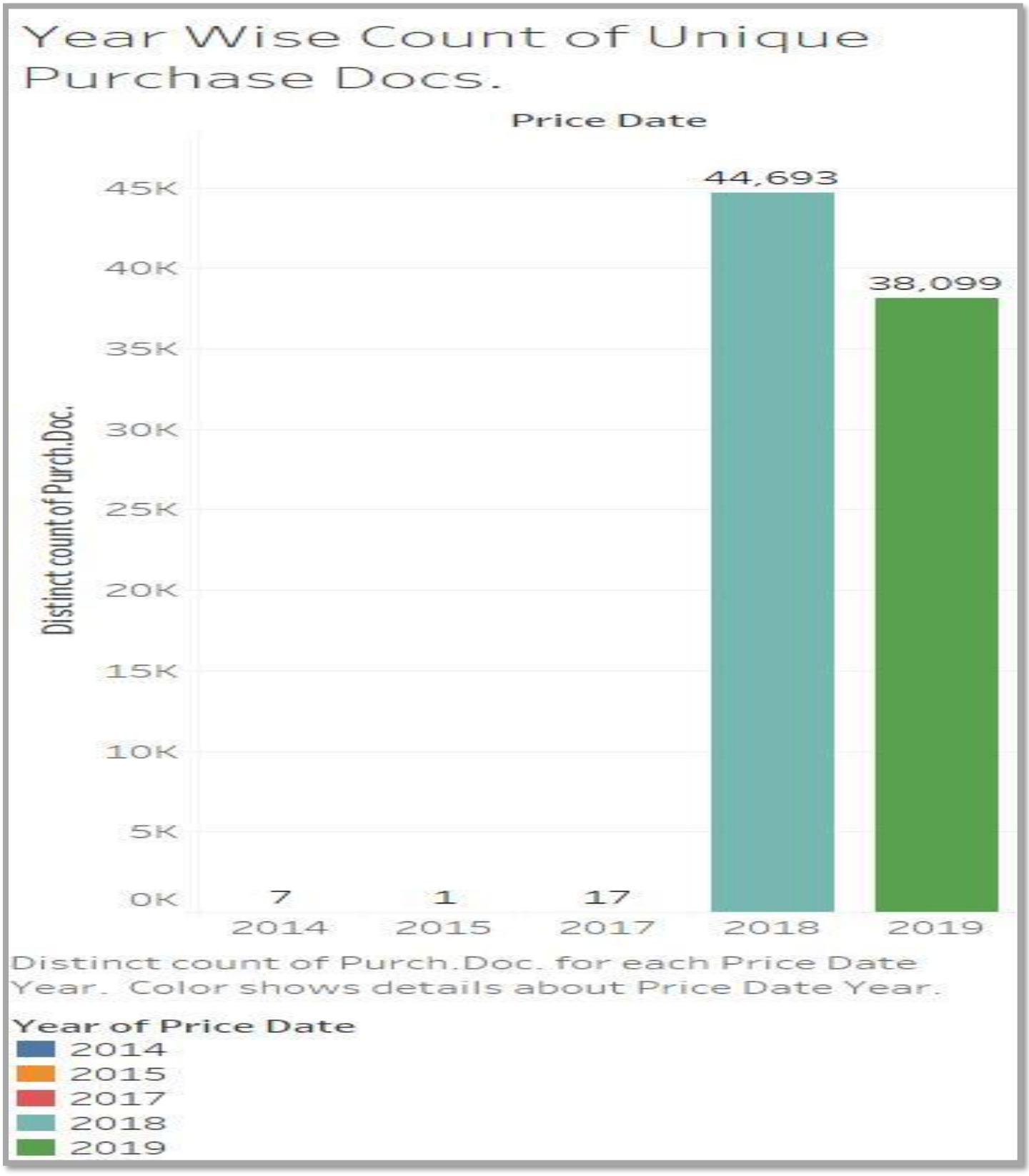
- **Unique Plants count Organisation wise:**

  In the below image we observed that data had 5 organisation codes, within each of them there were multiple plant codes all unique to each Plant. With counts 7; 7; 72; 1; & 45 under Co-cd codes 4500; 7000; 7860; 8000; & 9000 respectively.



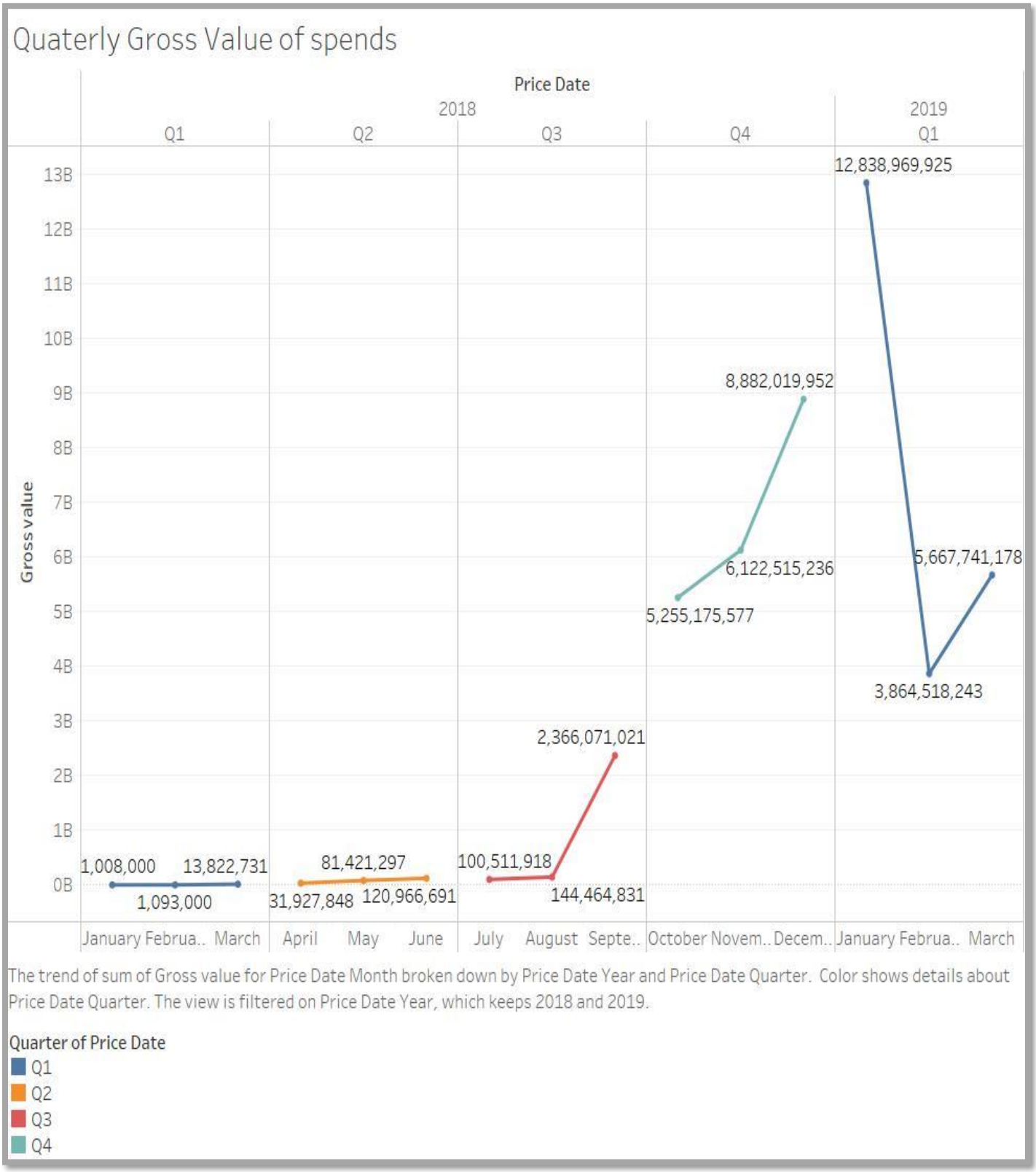
- **Year Wise Count of Unique Purchase Docs.**

  Purchase Docs. are simply Order Bill Numbers; The below graph shows the counts of unique bills (Purchase Docs) generated year on year basis.
  As it is clearly visible that the most the Purchase Docs were generated in the last 2 years i.e., Year 2018 & 2019.



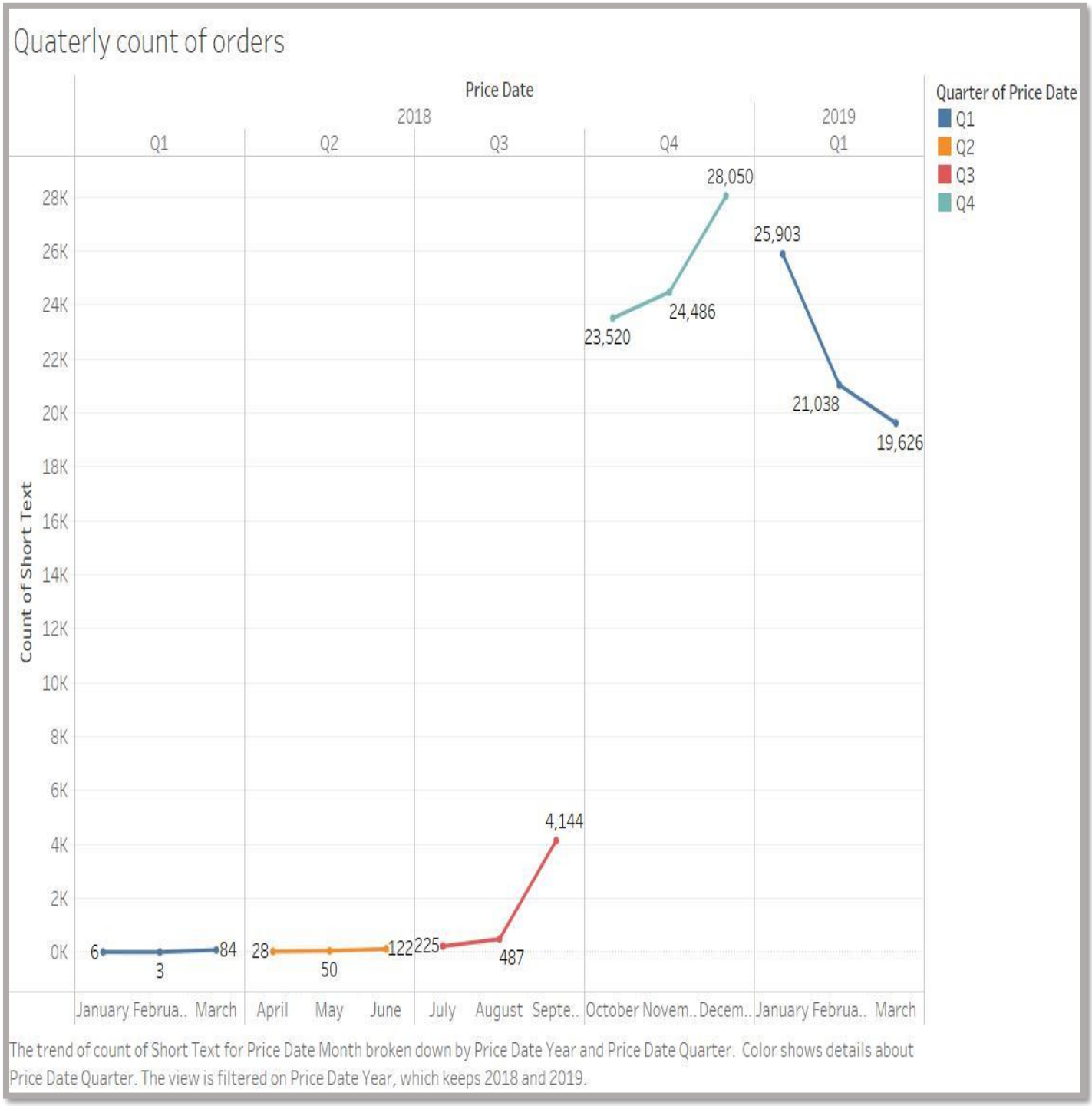
- **Quarterly Gross Value of spends:**

  Below image shows the Quarterly spend in Gross Value term. As we can see there's a sharp increase in the value in first quarter of 2019 as compared to previous quarters the reason for this is that there was increase in the quantity of a specific feeds purchased, and many similar items were purchased in multiple quantity compared to previous quarters.



- **Quarterly count of orders:**

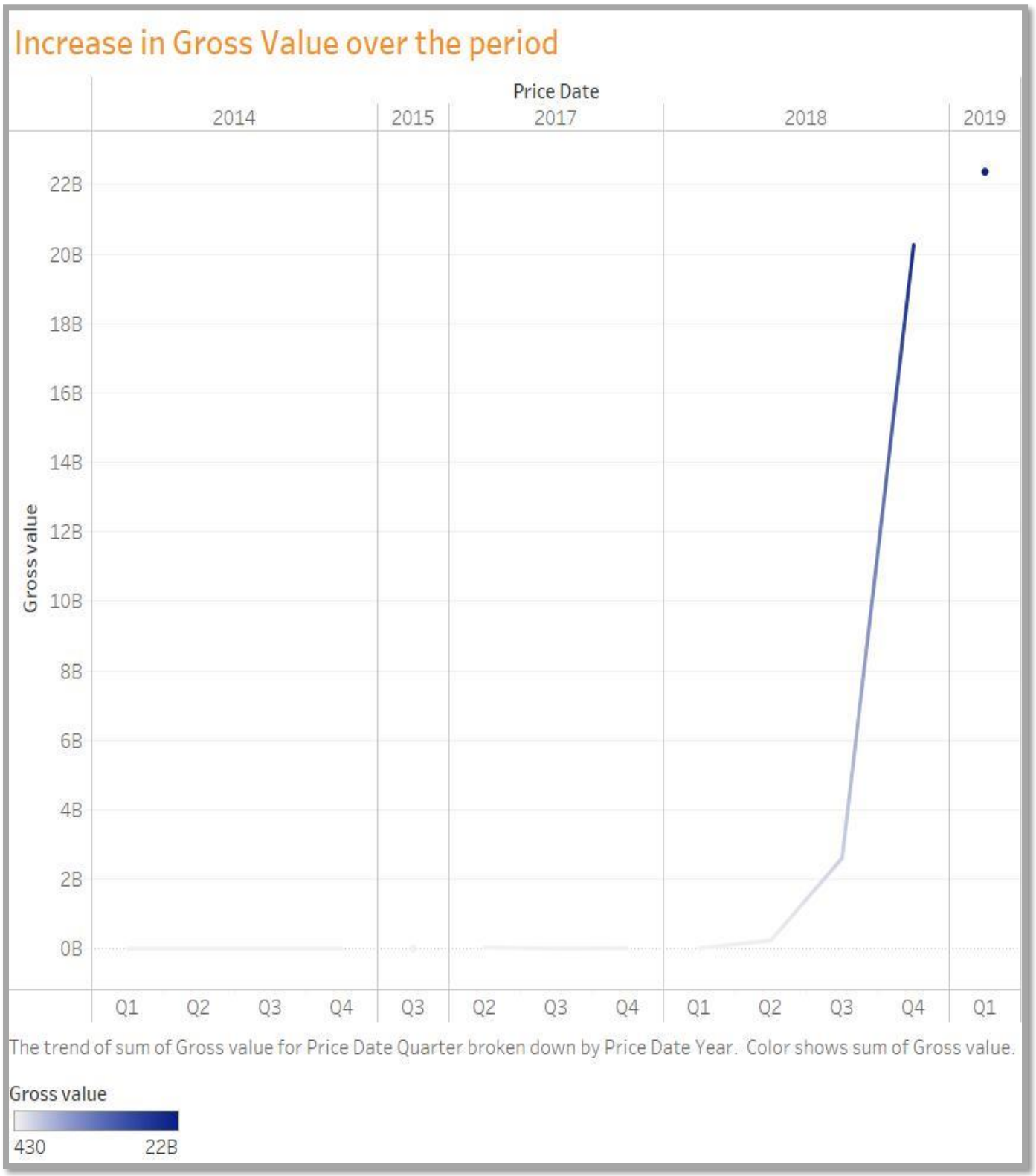
  Below Graph shows the number of orders made in every quarter.
  As seen in the above graph that last quarter of 2018 and first of 2019 accounted for most of the Spend in Gross Value terms it's clearly understood here that it's for the same reason there are higher counts of orders generated here.
  Interesting observation is that the percent Gross value difference and percent gross value difference are not close it's because of the order amount were of higher value even though the count od orders did not increase with the same rate.



- **Increase in Gross Value over the period:**

  It's a simple line graph indicating the spend pattern of the organization.

  Over the given period of time the it can be seen that organisation started spending very high in the late 2018 and 2019 it has been an exponential growth in the gross value.



- **Count of orders Month wise:**

  Below Tabular format of graph is showing the counts of orders month wise. It can be seen which are those items that were ordered the greatest number of times each month. Clearly, it indicated that these top 20 items are of regular use which is needed to run the business.

Count of orders Month wise

	Price Date			
	2018		2019	
	Q1	Q4	Q1	
Short Text	January	Februa.. Decem..	Jan..	Februa..
IB Ross Broiler Finis..		2,607	2,675	2,488
IB Ross Broiler Start..		2,350	2,343	2,228
Gunny-Bags Damag..		3,963	2,129	1,179
Gunny-Bags Damag..		947	1,857	1,909
IB Ross Broiler Pre-S..		1,768	1,750	1,693
Maize		2,910	1,374	109
Rice Bran Boiled		240	1,346	1,134
Rice Bran Raw		469	1,177	665
Soya Bean - (MP)		1,707	1,142	550
Soya Bean - (A)		946	784	772
Gunny-Bags Damag..		1,121	762	462
Soya Bean		1,075	711	1,002
Rice Husk (New)		433	504	386
Khandha		52	256	532
Diesel		155	159	140
Nakki		229	140	196
Beat Husk (New)		146	113	22
Diesel (New)		79	99	87
Egg Shell (Chilka & C..		93	89	85
GREEN		29	80	77
Clon-30 Vacc. (1000..		88	72	65
Clon-30 Vacc. (2500..		77	63	49
Gumboro Inter. (100..		80	56	59
Gumboro Plus (1000..		78	54	55
DOC Broiler (Ross)		21	51	38
Formoline		11	46	17

CNT(Short Text)

1 3,963

- **Yr-18-19 Top Items on the basis of Gross value:**

  The tabular graph below shows the list of top items that contributed to the Gross Value during the given time period. As we can clearly see there's an overlap of multiple items from the above graph indicating that those items who were ordered were also of high amount.

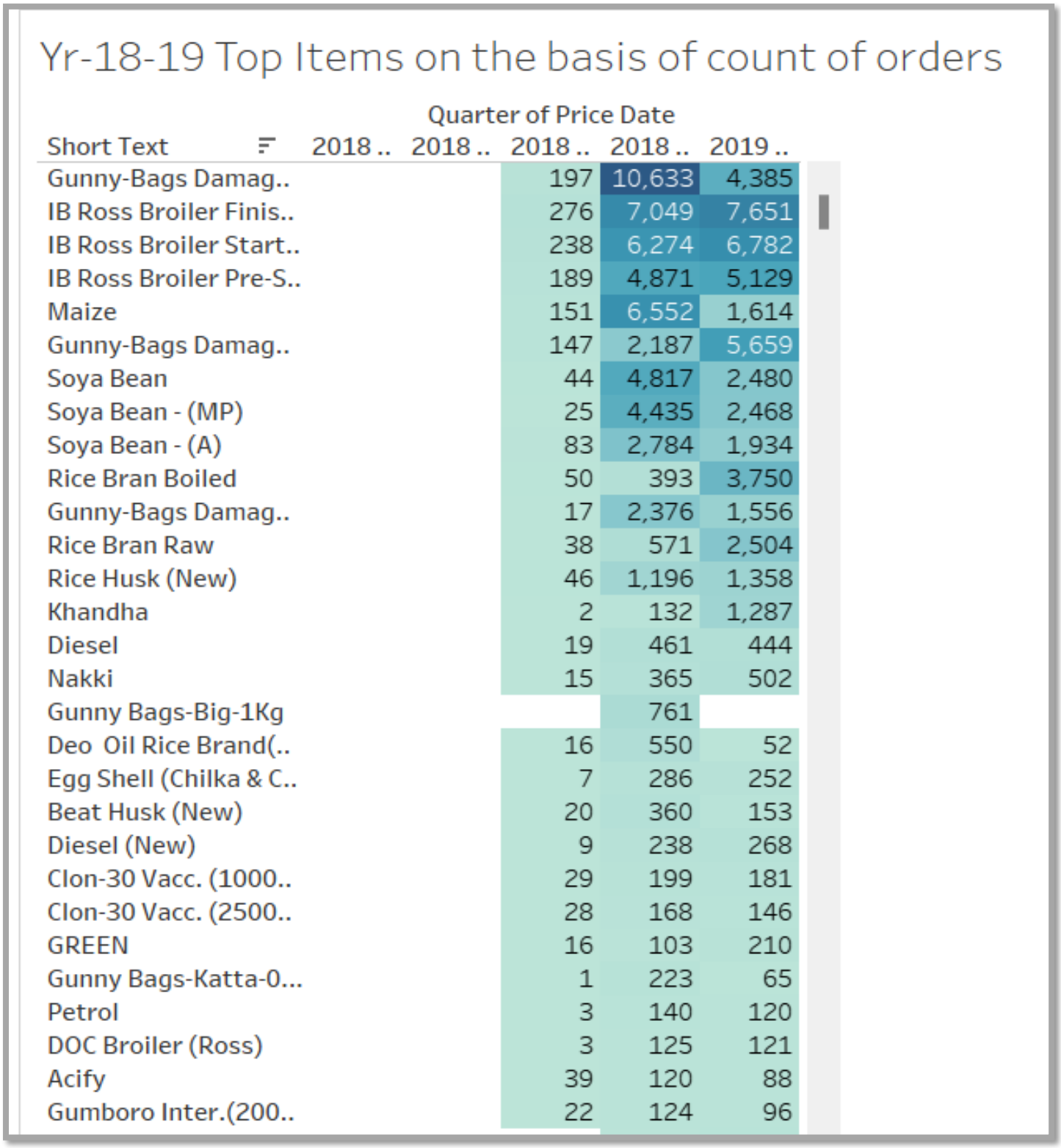
  High Value and High Frequency items

Yr-18-19 Top Items on the basis of Gross value

	Price Date				
	2018				2019
Short Text	Q1	Q2	Q3	Q4	Q1
IB Ross Broiler Finisher Feed			66,253,318	1,805,551,019	2,413,455,290
IB Ross Broiler Starter Feed			47,591,345	1,381,969,842	1,780,001,685
Soya Bean - (MP)			9,957,530	1,776,300,063	1,160,665,556
Maize			75,778,312	2,182,882,543	684,057,289
Soya Bean - (A)			57,361,310	1,221,905,446	948,613,722
B4 IB Ross Feed			134,743,710	696,920,000	1,054,080,000
PL-3			122,984,317	625,540,608	833,413,496
IB Ross Broiler Pre-Starter Feed			27,914,478	690,390,186	854,016,119
PL-4			140,504,519	574,331,226	817,984,426
Soya Bean			7,497,165	996,843,348	508,552,339
Feed P1			115,382,628	515,714,478	788,124,895
PL-5			87,313,431	574,885,354	665,772,945
B1 IB Ross Feed			72,926,854	414,796,560	820,581,512
B2 IB Ross Feed			138,259,813	498,255,663	631,991,205
B3 IB Ross Feed			150,222,038	476,441,650	636,587,727
Grand Parents Layer			86,250,914	356,935,750	531,899,455
GP-5 Feed			43,866,267	330,513,364	548,267,646
Feed PS			71,810,483	342,167,043	486,108,039
C2			106,300,000	368,257,539	369,002,000
Grand Parents Grower			43,656,900	282,254,902	423,651,274
Soya Crude Oil - Purchase			3,489,398	498,638,608	245,276,820
GP-4 Feed			23,377,824	281,123,501	419,359,019
Grand Parents Starter			24,318,507	231,067,760	313,953,034
C1			50,816,252	208,539,948	278,983,762
Khandha			465,925	34,886,298	465,973,763
Grand Parents Starter Male			24,110,679	146,679,729	191,392,621
PL-6			24,260,679	101,391,229	227,009,692
Rice Bran Boiled			6,186,465	32,369,199	242,602,807
IBG Common Vitamin			212,000,000	26,225,000	26,500,000
IMP Degum Soya Oil			15,194,021		191,939,242

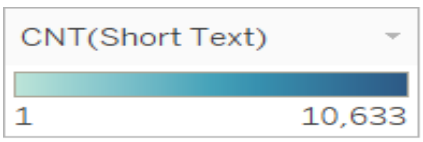
- **Yr-18-19 Top Items on the basis of count of orders:**

  Below clip shows the most ordered item in the year 2018 and 2019. It is clearly visible that the top 3 are fairly high compared to items ranked fourth to tenth, rest of the items are comparatively at significant lower count.



Yr-18-19 Top Items on the basis of count of orders

	Quarter of Price Date				
Short Text	2018 ..	2018 ..	2018 ..	2018 ..	2019 ..
Gunny-Bags Damag..			197	10,633	4,385
IB Ross Broiler Finis..			276	7,049	7,651
IB Ross Broiler Start..			238	6,274	6,782
IB Ross Broiler Pre-S..			189	4,871	5,129
Maize			151	6,552	1,614
Gunny-Bags Damag..			147	2,187	5,659
Soya Bean			44	4,817	2,480
Soya Bean - (MP)			25	4,435	2,468
Soya Bean - (A)			83	2,784	1,934
Rice Bran Boiled			50	393	3,750
Gunny-Bags Damag..			17	2,376	1,556
Rice Bran Raw			38	571	2,504
Rice Husk (New)			46	1,196	1,358
Khandha			2	132	1,287
Diesel			19	461	444
Nakki			15	365	502
Gunny Bags-Big-1Kg				761	
Deo Oil Rice Brand(..			16	550	52
Egg Shell (Chilka & C..			7	286	252
Beat Husk (New)			20	360	153
Diesel (New)			9	238	268
Clon-30 Vacc. (1000..			29	199	181
Clon-30 Vacc. (2500..			28	168	146
GREEN			16	103	210
Gunny Bags-Katta-0...			1	223	65
Petrol			3	140	120
DOC Broiler (Ross)			3	125	121
Acify			39	120	88
Gumboro Inter.(200..			22	124	96



Few Analytical insights that were also observed are mentioned below:

a) The discount that was obtained on overall Gross Value was around “10000000”
b) The Surcharges that were obtained on overall Gross Value was around “60000000”
c) Discounts that were obtained on the items were few but impacting to a great extent
d) Total of 5092 transactions were given some discount under which we have 150 unique items
e) Total of 11983 transactions were surcharged under which we have 3284 unique items
f) Co-cd No. 9000 had twice the transactions compared to Co-cd No. 7860 but Co-cd No. 9000 constituted just half Gross Value as compared to Co-cd No. 7860

Conclusion and Solution on Analysis done:

As It’s understood now that the spend is highly impacted by items of high frequency and high value which is clusters were shown by yellow and green colour.

The organisation should look into procurement of these items by obtaining better discounts on these cluster. Organization is of poultry farm so they should see into priority of these items and also reach to better seller in the market so as save costs on procuring them.

Over all there be a better mechanism into procuring these products. Which will eventually optimise the process and help in savings.

Thank you for reading this…