MINOR PROJECT

Details of Team Members	
Ojass Dhadiwal	B21EE045
Paawan Karwa	B21EE046

Dataset:Retail

>Abstract Idea-

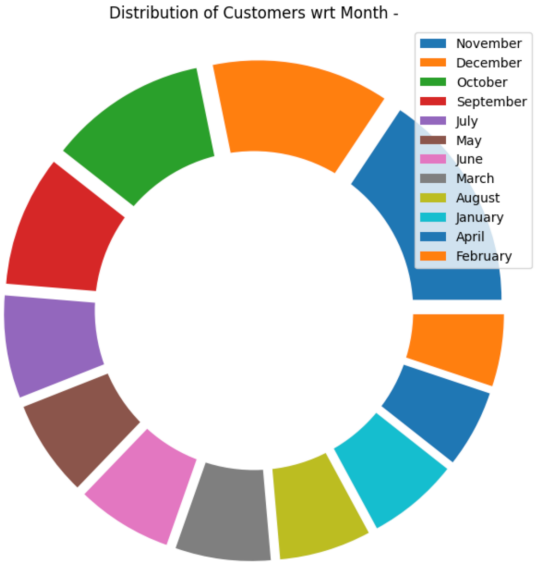
We have a **database of retail** so basically we can use this data to either predict **things which a customer is more probable to purchase** together or we can **classify or group customers into clusters** so that we can predict or analyze them better.

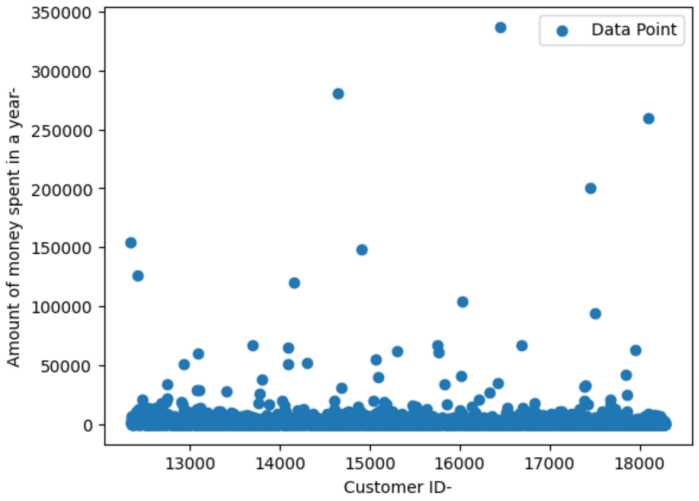
>Dataset Description-

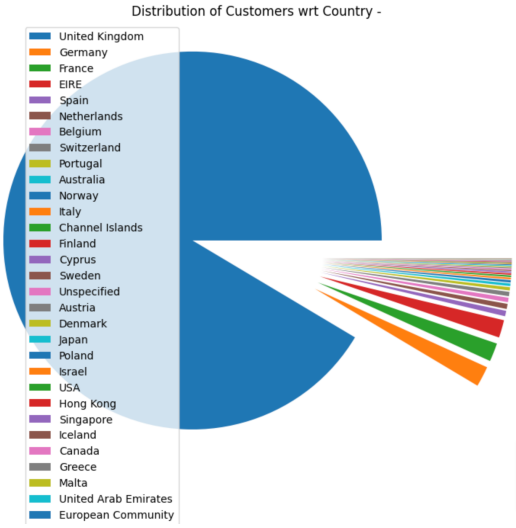
We have a retail dataset of a store consisting of **Invoice No, Stock Code, Description of Product, Quantity, Invoice Date,Unit Price, Customer ID and Country**. Now we can use machine learning to cluster the customers or moreover find which product is being purchased at what amount of time and which type of customer.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

Now I plotted some graphs to better understand the dataset. For example, the Majority of the country the data is from is the UK, moreover we can also see that , the amount of money spent by all customers and how there are very few customers who spent more than 50000.







>Dataset Preprocessing and Data Analysis-

>First of all we saw that there are null values in columns - Description and CustomerID.Now as **no of null Description is 0.00268%** of total rows, so these rows can be dropped. As **null values in CustomerID is approx 24.92%** so we can't drop them. But on the other hand

CustomerId is a very important column for Customer Analysis. Thus we need to drop these columns.

›Now we can see that there is **very noisy data in StockCode**, like there are letters in some Stock Codes so we basically converted them to numbers. And moreover we can see that some values of **Unit Price and Quantity are also -ve**. So we first converted values for all of them.

›Now on further analysis of data **we saw that these defected Descriptions have letters of Descriptions in lower case**.Thus we basically applied this theory and calculated no of corresponding rows and **there came out to be a total 2847 rows**.

```
?? missing 1
wet pallet 3
????missing 1
???missing 2
lost in space 1
wet? 1
lost?? 1
wet 1
wet boxes 1
mixed up 2
lost 1
Total values which can be dropped- 2847
```

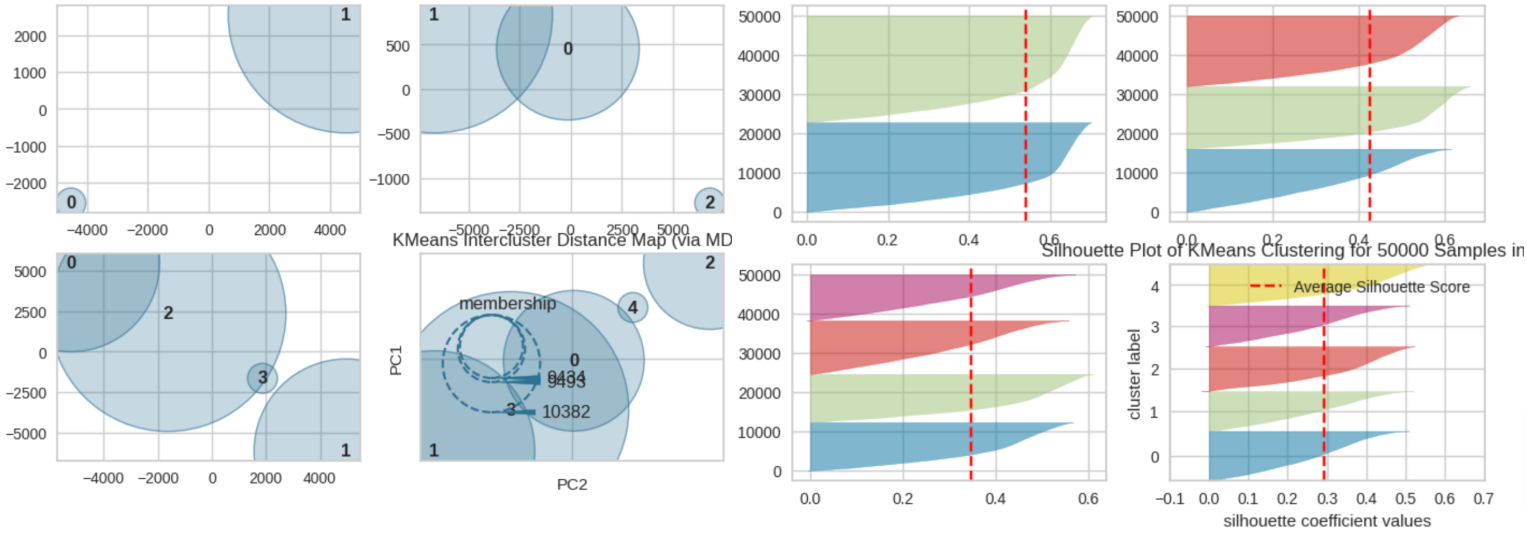
›Now further we convert every **Description value to its equivalent alpha-numeric value** only without any spaces and symbols to remove further noise.

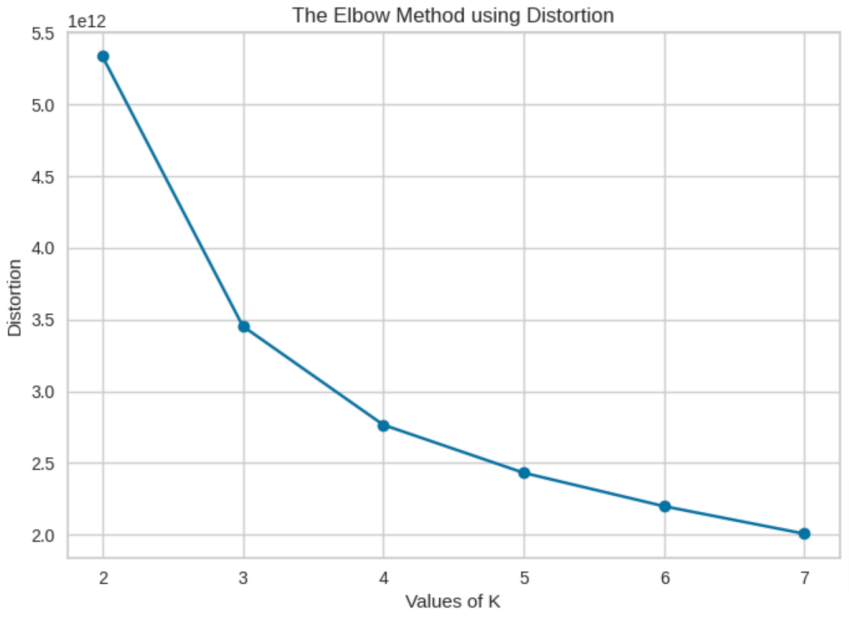
›Further we **extracted month and Time from InvoiceDat**e, and finally made two new columns for them.Year, day, minute and second are not that important. Thus we finally **divided time into 6 bins** namely from 12-4PM , 4-8PM and so on.

›Finally we created a new column for the **amount which is equal to unit price x quantity**. This column will help us to classify customers more clearly.Finally **we dropped Invoice Date , Stock Code and label encoded Country, Invoice No and Description**.

>What are Optimum no of clusters for the data ?

Now as data is **9 dimensional thus we can't see data in 2d plots without any transformation** like PCA and ICA. Thus we would need other techniques for obtaining optimum no of clusters namely, **Inter Cluster Distance Plots, Silhouette Score for various no of clusters and finding clusters from elbow method**. So we used the **yellowbricks** library which has inbuilt functions for each of them. And I received the following plots for them-





>Firstly for InterClusterDistance we can see that when **no_of_clusters = 5 then there are several overlaps** but when there are 4 clusters

then then we can see there are nearly **no overlaps and it also covers most of the data**.
>Secondly from Silhouette score plots we can see that though silhouette score is not very good for all of them but **no_of_clusters should be either 3 or 4**.

>Thirdly we can see that from the elbow method also , **the point is created near no_of_clusters = 4 and after n=4 the graph is approximately linear**.

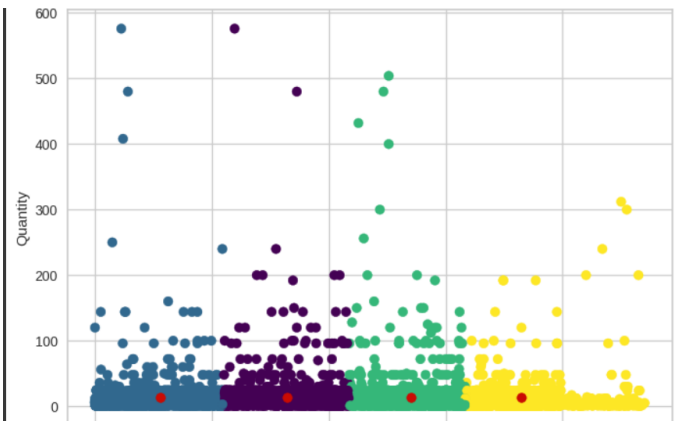
Thus **finally we decided to go with no_of_clusters = 4**, though silhouette coefficient is very less but from InterClusterDistance they cover most of the region and thus would be a better model than no_of_clusters = 3.

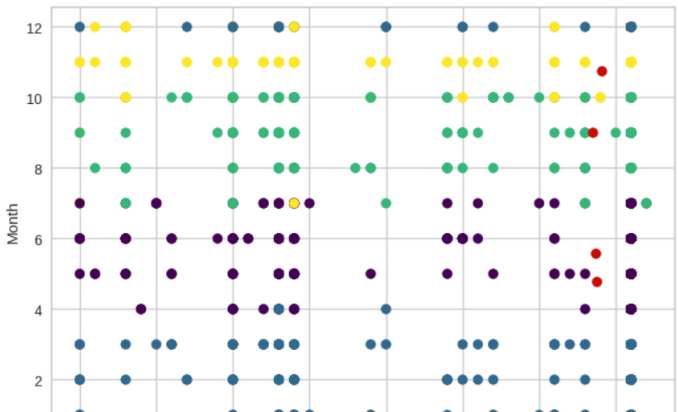
>Unsupervised learning models

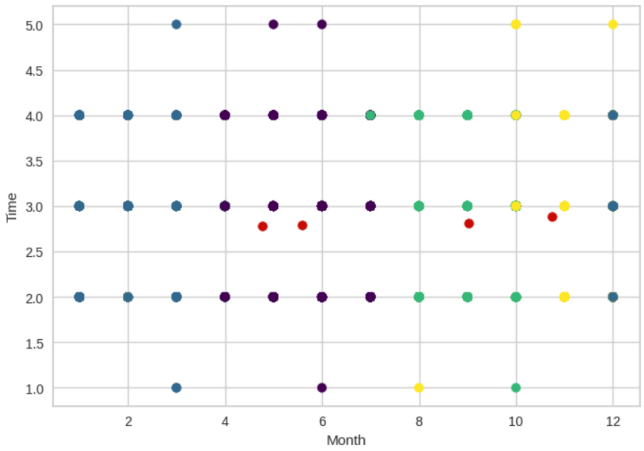
> In this section, we have implemented **KMeans clustering algorithm** and **Agglomerative Hierarchical Clustering algorithm.**

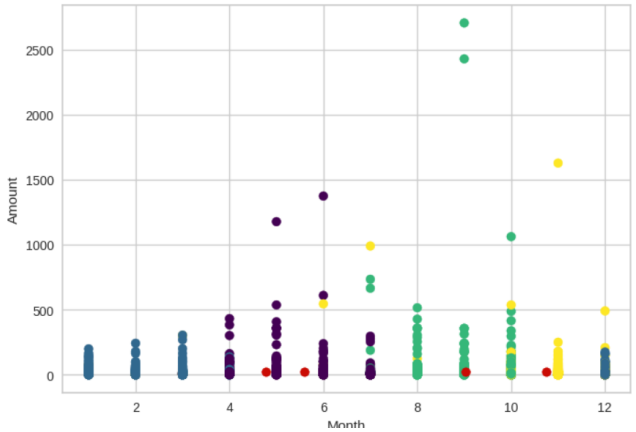
> At first, we applied these algorithms upon our entire pre-processed dataset.
> Now, our data is **9 dimensional** and the maximum number of dimensions we perceive are 3. Also some columns have a greater impact upon our clusters as compared to others and higher dimensions even lead to overfitting in a few cases.

> To visualize the clusters for the whole dataset, we took the top 2 and top 3 features and plotted them accordingly.









› Now, we cannot really predict or analyze anything from the above plotted graphs as our information is divided in between the 9 dimensions and when we take 2 dimensions at a time, we only have access to limited information.

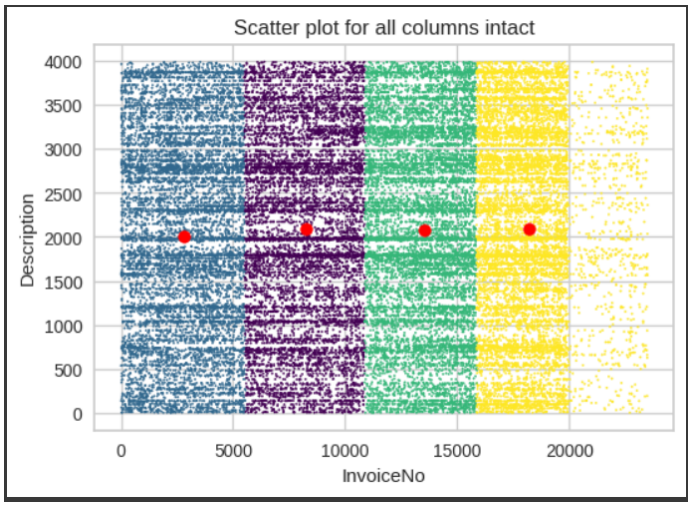
› Hence, we dimensionally reduced our data using the PCA **algorithm** and created 2 new datasets which were **2-Dimensional** and **3-Dimensional** respectively.

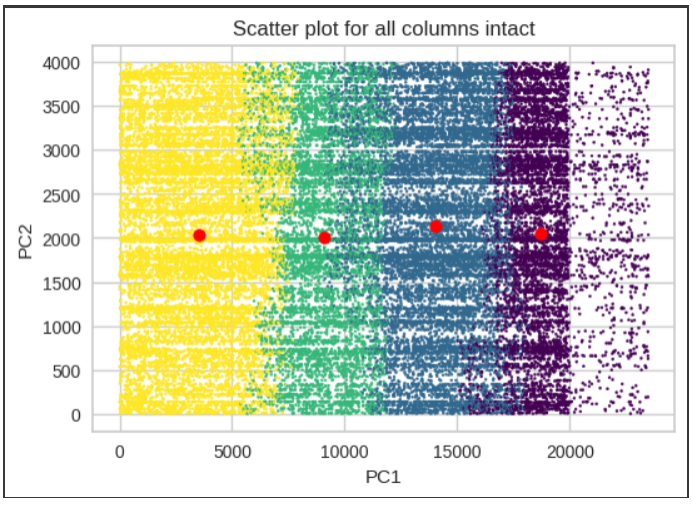
› Then after fitting the algorithm and clustering the data points, we plotted the clusters and calculated the

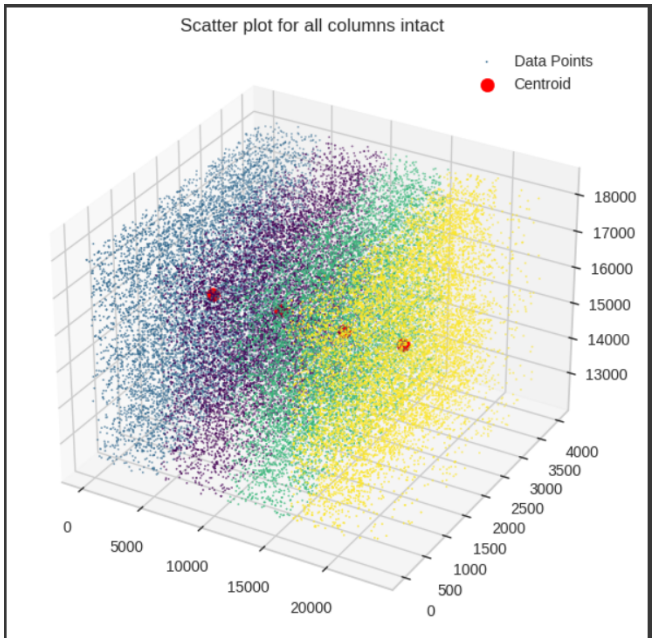
- **Inertia** (or SSE)
- **Silhouette coefficient**
- **Calinski Harabasz score**
- **Davies Bouldin score**

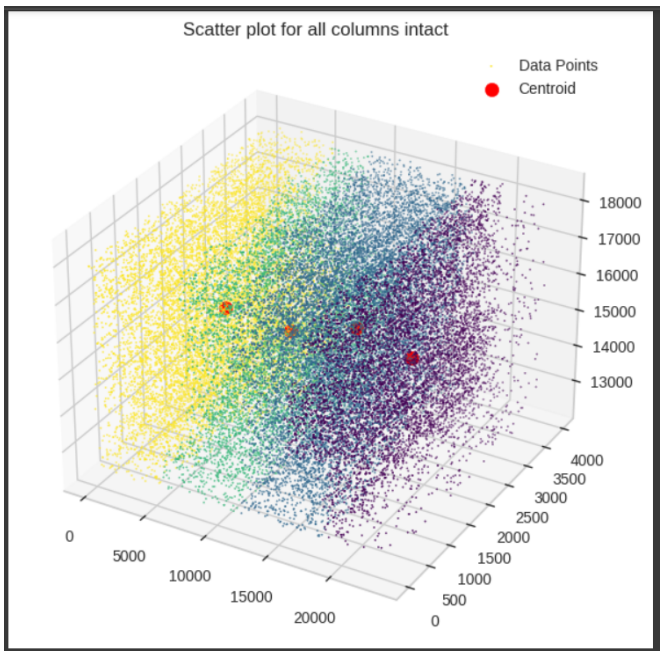
for all three datasets over both algorithms.

› The cluster plots for entire dataset are :



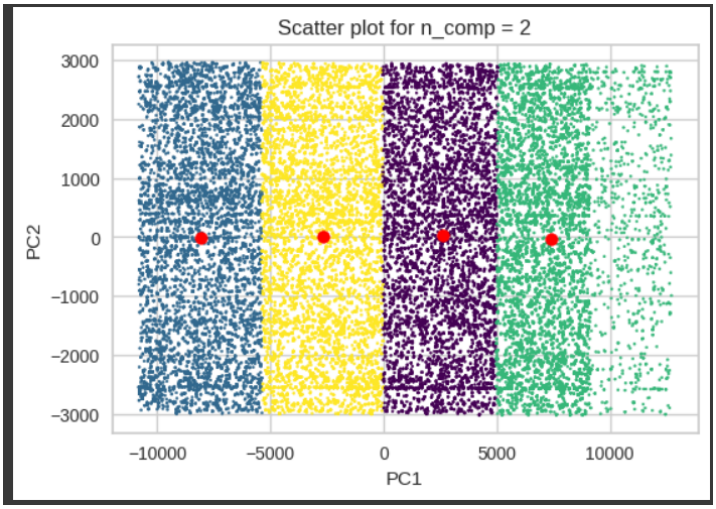


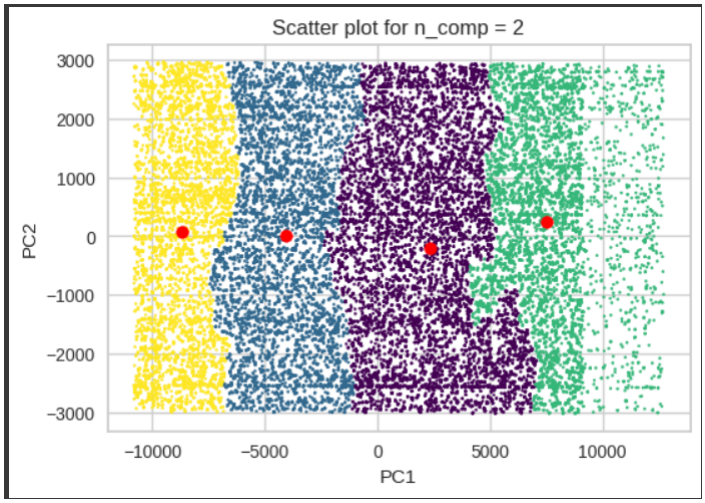


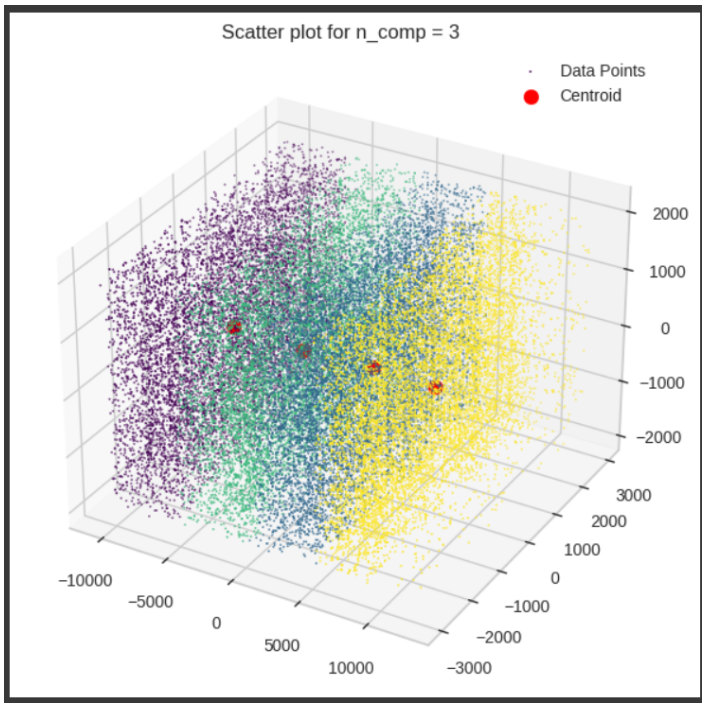


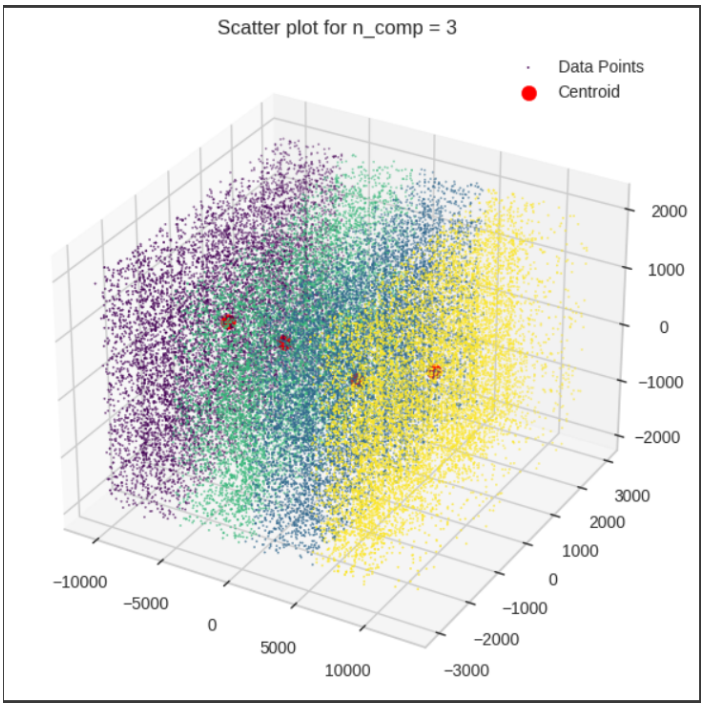
› The plots on the left are KMeans clustering and the right ones are Agglomerative clustering.

› Now the plots for the dimensionally reduced datasets are as follows:









› From the plots between original dataset and dimensionally reduced ones, we can clearly see that the dimensionally reduced plots have clearer, distinct boundaries between the clusters. Henceforth we would be analyzing the PCA reduced datasets.

› Now in between the **KMeans and Agglomerative** plots, we can see in the 2-D plot that **KMeans** has quite sharp boundaries which change at almost regular intervals with respect to **principal component 1.**

› **Agglomerative clustering** has quite **haphazard boundaries with clusters intermixing among each other** in dimensionally reduced dataset and more so in the original dataset.

› In the 3-D plot, we can see that the **cluster centers in the KMeans clustering are more or less uniform with respect to PCA 2.**

› In Agglomerative clustering, the cluster centers vary to a significant extent with respect to the PCA 2.

› Hence we can conclude that the **KMeans algorithm is working better than the Agglomerative Hierarchical clustering** for our dataset.

› One of the reasons behind this is that we are training the **KMeans model with the entire dataset** due to which it creates clusters properly whereas we are training the **Agglomerative clustering on a randomly sampled dataset due to which it is underfit with respect to our dataset.**

› We are doing this because **Agglomerative clustering has a tendency to overfit when large amounts of training data is passed to it** and **works bad for large datasets.**

› Also it has a **high time-complexity** and **importantly high space-complexity** which **limited our ability to find the sweet-spot for training data size and compute the clusters accordingly.**

› Now to support our claim mathematically, we calculated the above mentioned metrics for all six different combinations we created. It is as follows :

	K-Means	Hierarchical	Inertia	Silhouette Score	Calinski Harabasz score	Davies Bouldin score
PCA n_comp = 2	Yes	No	203413243234.0421	0.3957613872823062	78836.11762968544	0.8500866948048321
PCA n_comp = 3	Yes	No	251076884187.83252	0.34749074868107915	63871.13609425703	0.9674557856807204
Original Dataset	Yes	No	286314309385.9605	0.34663609783008315	56009.67449736726	0.9696319284933848
PCA n_comp = 2	No	Yes	-	0.35269704357965137	68243.94843284885	0.8986219124833608
PCA n_comp = 3	No	Yes	-	0.32289952584262016	59254.13604801025	1.0117958337105155
Original Dataset	No	Yes	-	0.2951636777280024	48656.2423305648	1.0213093494951362

› Now, we can see that the corresponding silhouette score for KMeans is better than that of Agglomerative for every dataset. Hence our hypothesis that KMeans works better is proven to be True.

› We can also conclude that the best dataset and number of dimensions are **PCA 2-D dataset** and **2 dimensions** respectively.