Unsupervised learning technique

By
Oluyomi Alabi

Introduction

In this project, an unsupervised learning techniques is applied to a ‘wholesale Data’ real-world data set gotten from Kaggle. Patterns were identified, optimall number of clusters were determined, the most important features that contribute the most to the overall variance in the dataset were also identified.

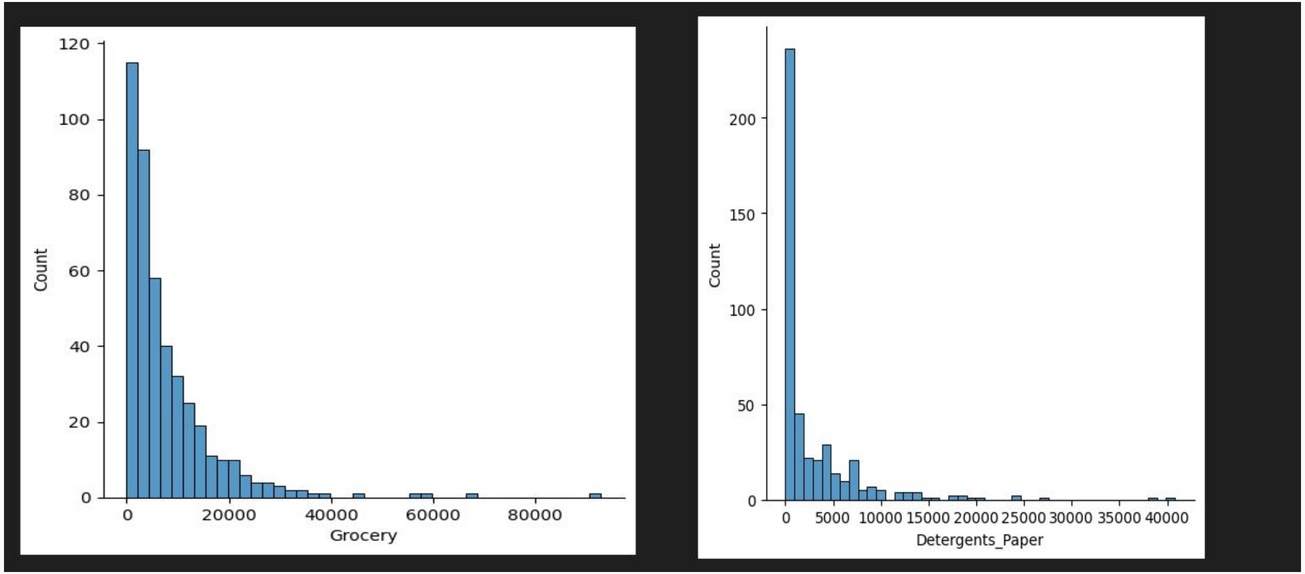
Project Goals

- To identify patterns and correlation between variables.
- To perform k-means clustering, hierarchical clustering, and principal component analysis (PCA) .
- To determine the optimal number of clusters and communicate the insights gained through data visualization.

Process

- Obtaining data
- Loading and understanding the dataset
- Exploratory data analysis and pre-processing
- Performed k-means clustering.
- Hierarchical clustering.
- principal component analysis (PCA).

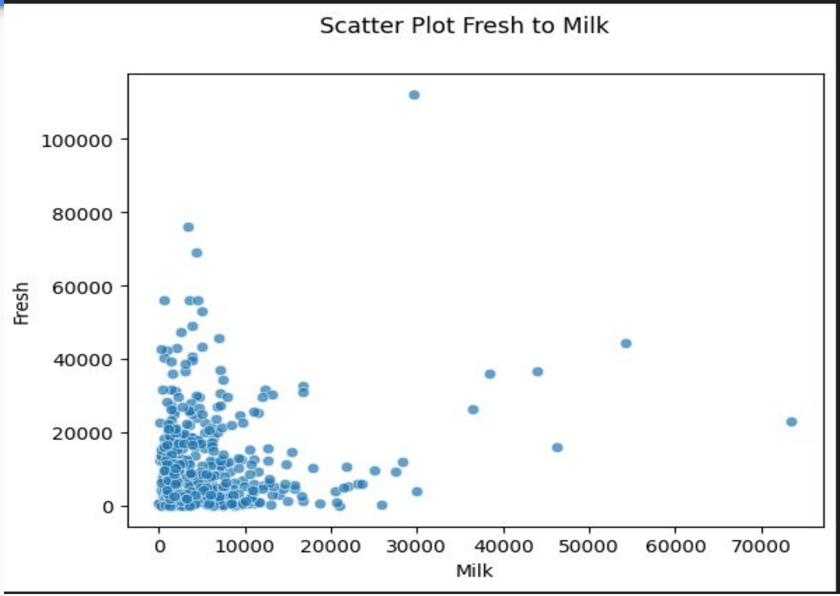
Exploratory Data Analysis Visualization.

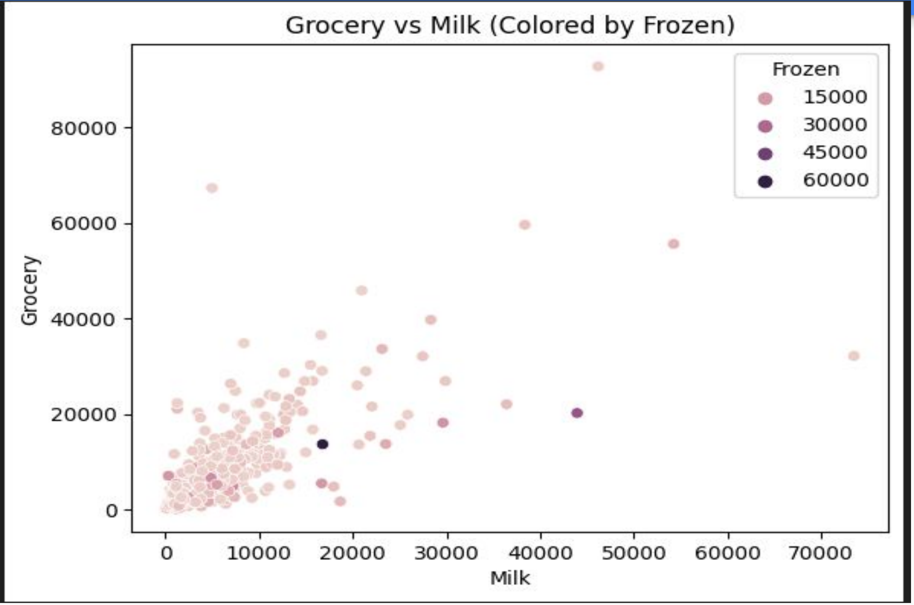


Displot showing distribution of Grocery and Detergents-paper..

It can be seen that this variables are not normal distributed . They are skewed indicating outliers.

Visualization continued





Scatterplots showing the distribution and correlation of warriors features

Results: Distribution statistics showing statistical significant correlation between the feature variables

```
df.describe()
```

✓ 0.0s

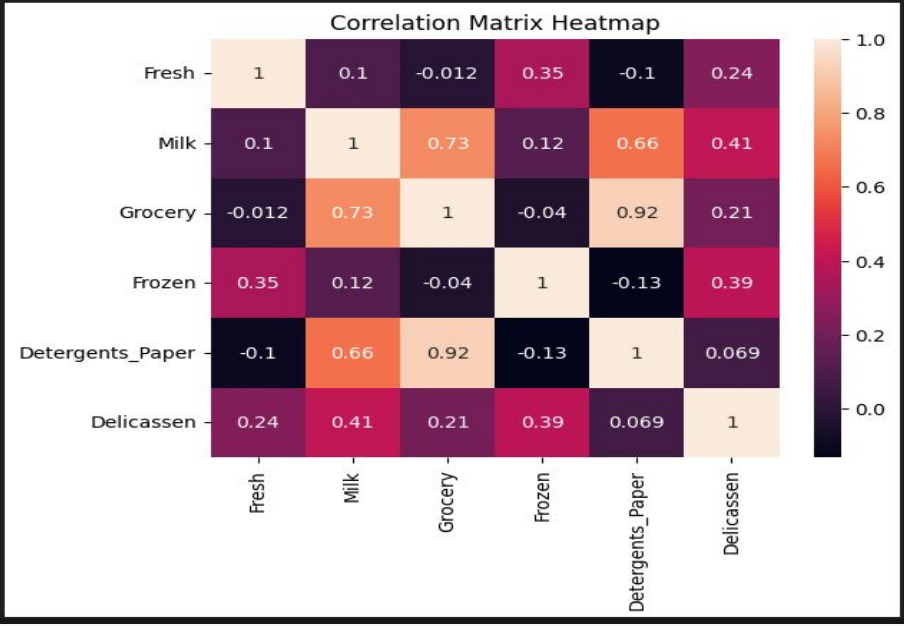
	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.000000
mean	12000.297727	5796.265909	7951.277273	3071.931818	2881.493182	1524.870455
std	12647.328865	7380.377175	9503.162829	4854.673333	4767.854448	2820.105937
min	3.000000	55.000000	3.000000	25.000000	3.000000	3.000000
25%	3127.750000	1533.000000	2153.000000	742.250000	256.750000	408.250000
50%	8504.000000	3627.000000	4755.500000	1526.000000	816.500000	965.500000
75%	16933.750000	7190.250000	10655.750000	3554.250000	3922.000000	1820.250000
max	112151.000000	73498.000000	92780.000000	60869.000000	40827.000000	47943.000000

Correlation Heatmap

```
#Heatmap showing correlation in different variables.
# Create a heat map with correlation data
sns.heatmap(data= corr, annot=True)
plt.title("Correlation Matrix Heatmap")
```

✓ 0.4s

```
Text(0.5, 1.0, 'Correlation Matrix Heatmap')
```



Based on the correlation coefficient, we can interpret thus that there is almost no linear correlation between 'Fresh' and 'Grocery' spending which is -0.012

There is a weak positive correlation between Detergents_paper and Delicassen.0.069

There is a significant relationship between Milk and Detergents_paper 0.66 meaning as the sales of milk increases, so is the sales of Detergents_paper.

There is a strong positive correlation between Grocery and Detergent_paper this indicates that as Grocery sales increases so is Detergents_paper.

-1 indicates a perfect negative correlation .

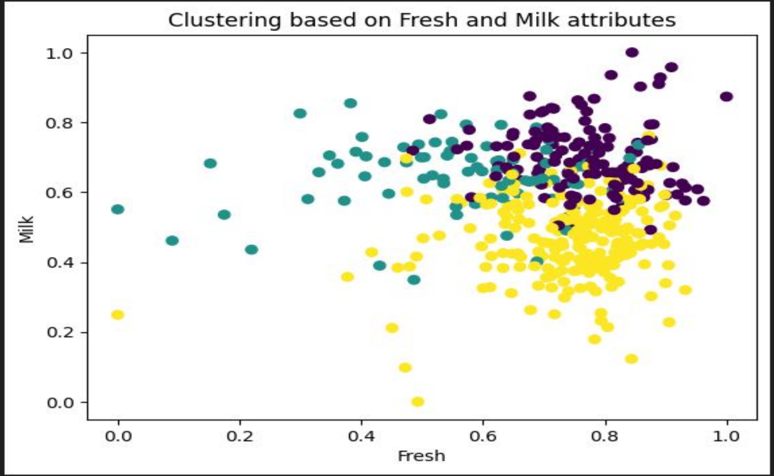
0 indicates no correlation .

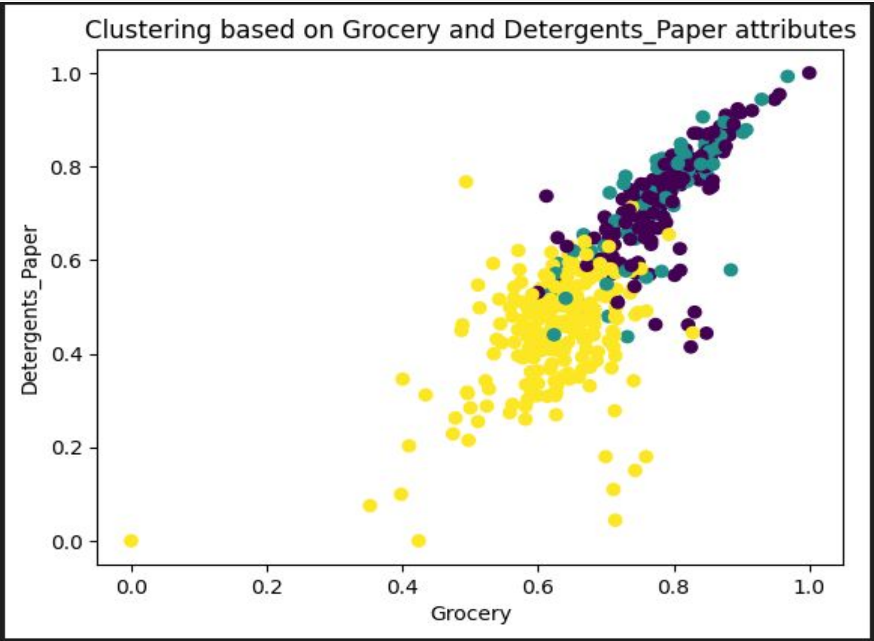
1 indicates a perfect positive correlation

Clusters visualization for 2 features in Kmeans

```
#visualize the clusters (using the first two features for simplicity)
plt.scatter(X['Fresh'], X['Milk'], c=df1['Cluster'], cmap='viridis')
plt.xlabel('Fresh')
plt.ylabel('Milk')
plt.title('Clustering based on Fresh and Milk attributes')
plt.show()
```

✓ 0.2s





3D scatter plot showing features

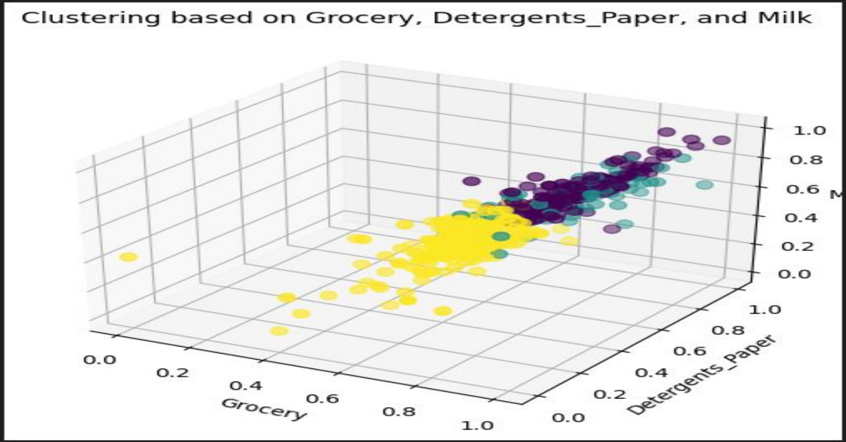
```
from mpl_toolkits.mplot3d import Axes3D

# Visualize the Clusters (3D Scatter Plot)
fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X['Grocery'], X['Detergents_Paper'], X['Milk'], c=df1['Cluster'], cmap='viridis', s=50)
ax.set_xlabel('Grocery')
ax.set_ylabel('Detergents_Paper')
ax.set_zlabel('Milk')
ax.set_title('Clustering based on Grocery, Detergents_Paper, and Milk')

plt.show()
```

0.2s



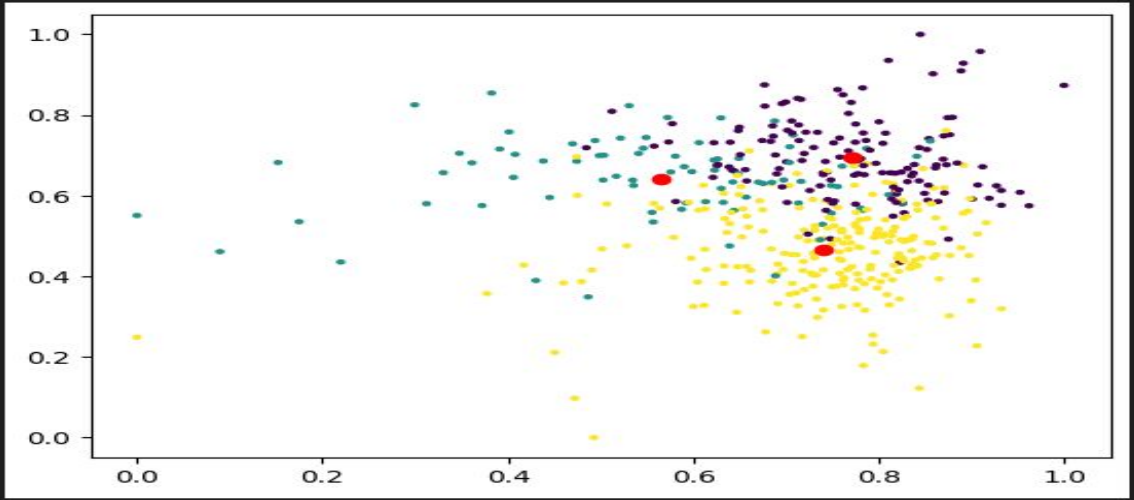
3D plot of feature with positive significant correlation (Grocery, Detergents_paper,and milk)

Plots of predicted clusters along with their centroids.

```
plt.scatter(X.iloc[:,0], X.iloc[:,1], c=y_pred, s=5) #have color (c) represent the predictions (y_pred)
plt.scatter(centroid[:, 0], centroid[:, 1], c='red') #print the centroids model.cluster_centers_
```

✓ 0.5s

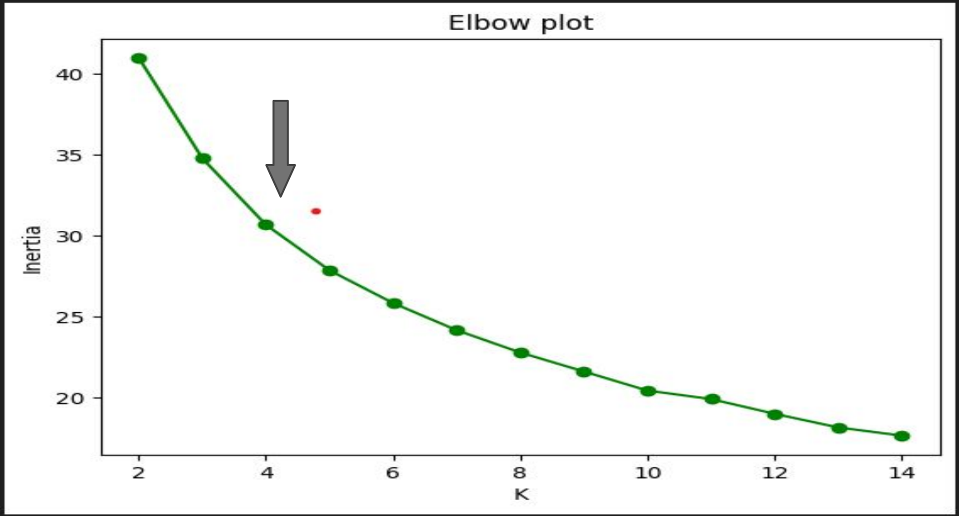
```
<matplotlib.collections.PathCollection at 0x2852f768dc0>
```



Elbow plot.

```
plt.plot(range(2,15), inertia, 'og-')
plt.title('Elbow plot')
plt.xlabel("K")
plt.ylabel("Inertia");
```

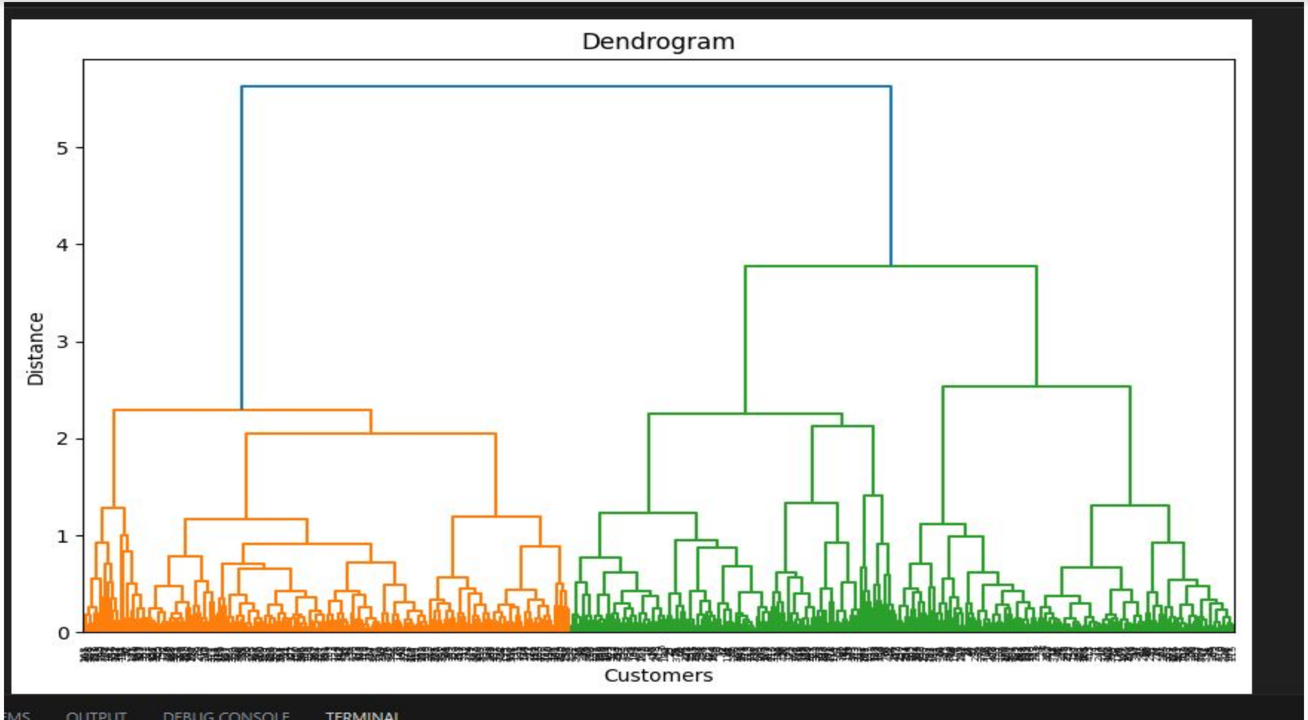
0.7s



As inertia drops, there is an increase in K but when idt decreases, k increase.

The optimum point on K is approximately 4.

Dendrogram plot from hierarchical clustering

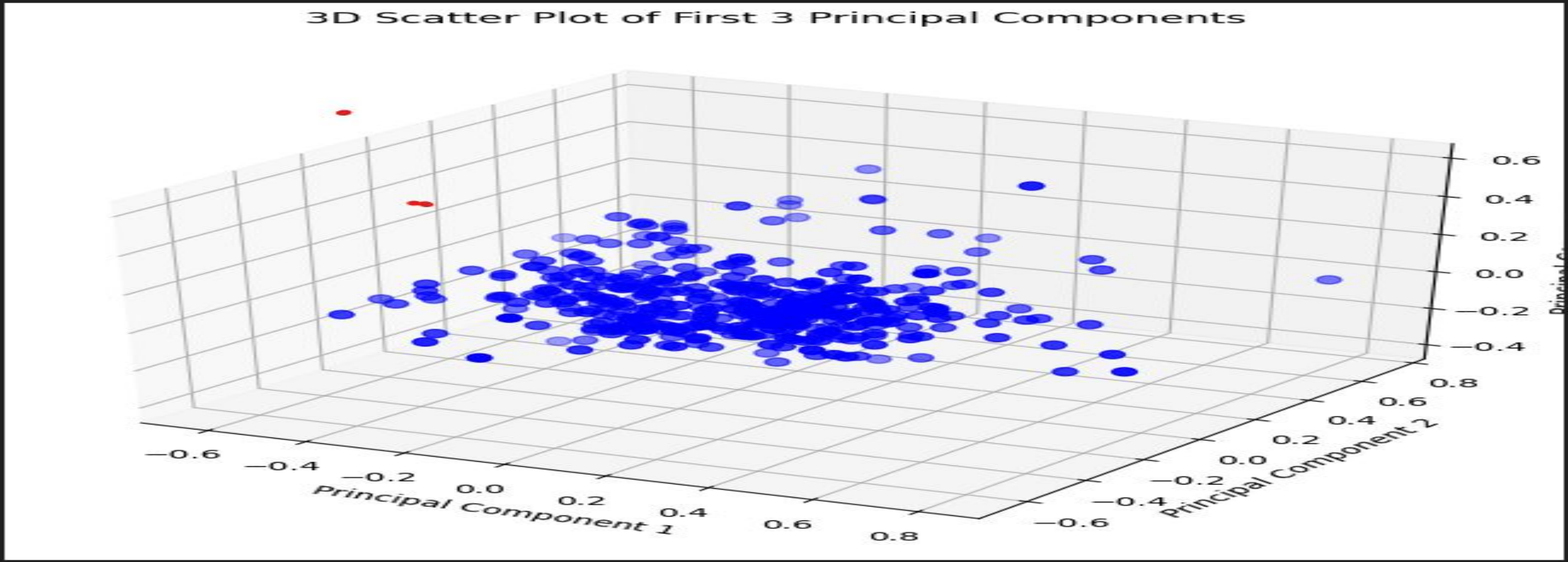


The dendrogram plot showing the relationship of various feature, we can determine the optimum number of clusters based on the vertical lines (threshold_distance) where it forms distinct clusters. The threshold_distance is selected based on the height of the dendrogram where it is clear to separate the clusters and this is around 4.

3D plot of the first 3 principal components

```
ax.set_xlabel('Principal Component 1')
ax.set_ylabel('Principal Component 2')
ax.set_zlabel('Principal Component 3')
ax.set_title('3D Scatter Plot of First 3 Principal Components')
plt.show()
```

✓ 0.2s



Conclusion

- The first 3 principal components captured most of the variance in the data giving approximately 90% of the model prediction which accounts for the larger part of the model that can be retained and this is likely to be Grocery and Detergent_paper and milk . These are the most important feature when considering wholesale distribution by stakeholders.
- There is a strong positive correlation between Grocery and Detergent_paper this indicates that as Grocery sales increases so is Detergents_paper therefore wholesale distributors should take note of these features during distribution.
- The dendrogram plot shows that the optimum number of cluster is around 4.
- The inertia in the elbow plot drops very quickly as we increase k up to 4, but then it decreases much more slowly as we keep increasing k.The optimum number of cluster is likely to be here also 4.

Challenges

- Time constraint to explore more on the principal component.

THANK YOU

THANK YOU