**Capstone Project- Online Retail Store Data Project**

**Submitted By- Diwakar Acharya**

Table of Contents

[Problem Statement 3](#_Toc125142499)

[Project Objective 4](#_Toc125142500)

[Data Description 5](#_Toc125142501)

[**1)** **Statistics insights that can be used by online retailer to improve in various areas:** 6](#_Toc125142502)

[2. **Customer** **Segmentation** 9](#_Toc125142503)

[Data pre-processing Steps and Inspiration 11](#_Toc125142504)

[Choosing the Algorithm for the project 12](#_Toc125142505)

[Motivation and Reasons for choosing the Algorithm 14](#_Toc125142506)

[Assumptions 15](#_Toc125142507)

[Model Evaluation and techniques 16](#_Toc125142508)

[Inferences from the same 17](#_Toc125142509)

[Future possibilities of the project 18](#_Toc125142510)

[Conclusion: 19](#_Toc125142511)

# Problem Statement

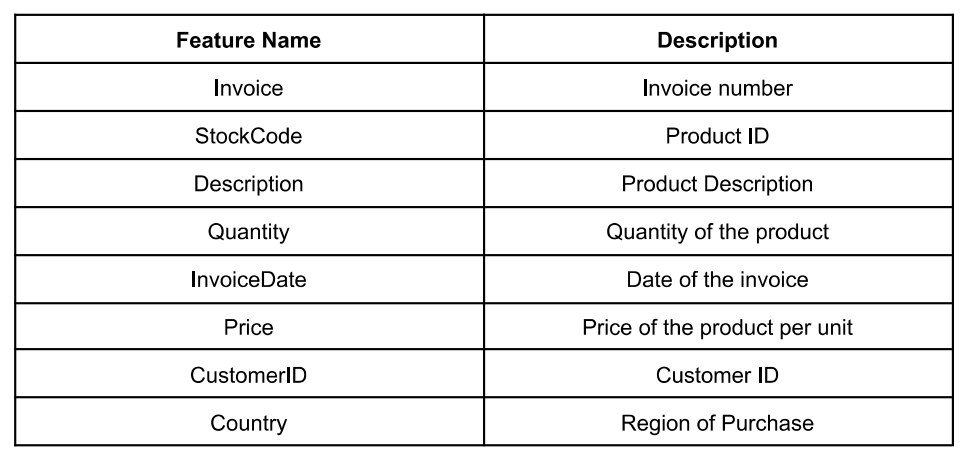
An online retail store is trying to understand the various customer purchase patterns for their firm, we are required to give enough evidence based insights to provide the same.

# Project Objective

1. With the project we have to come up with the useful insights about the customer purchasing history that can be an added advantage for the online retailer.
2. Segment the customers based on their purchasing history.

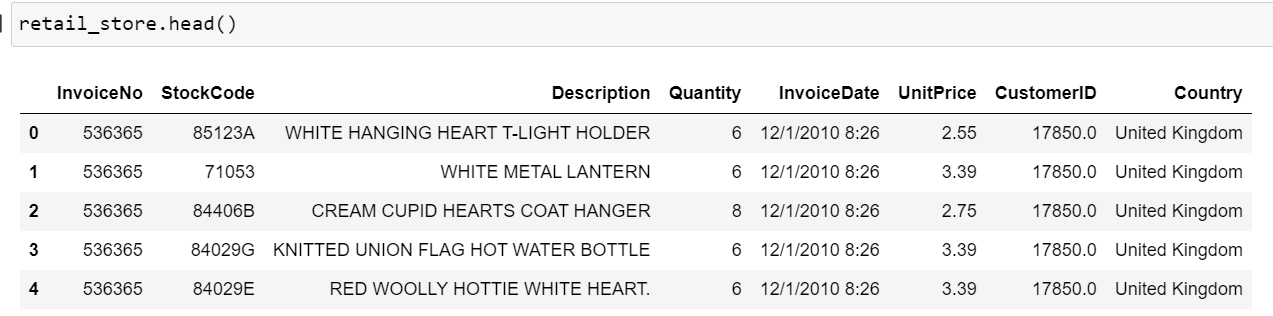
# Data Description

Given data is Online Retail Store data in CSV format. It is having 541909 rows and 8 columns. These columns represent features name.

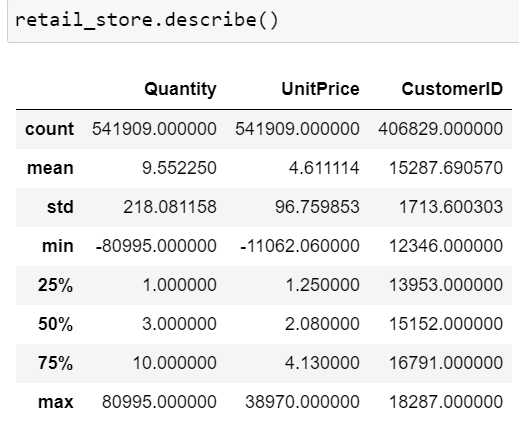


Insights from the data:

1. Basic details of data



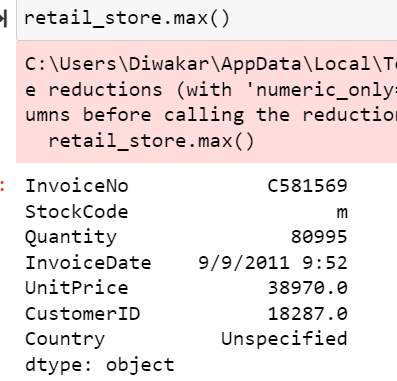
1. All statistical figure



1. Correlation between features:



1. **Maximum**

****

### **Statistics insights that can be used by online retailer to improve in various areas:**

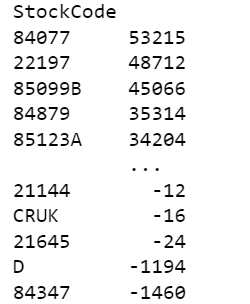
I have analysed below task and basis insights is provided.

1. **Popular products:** The most frequently purchased products can be identified by analysing the product description field. These are the products that the store should keep in stock and possibly promote more.

popular\_products = retail\_store.groupby("StockCode")["Quantity"].sum().sort\_values(ascending=False)

print(popular\_products)

**Most popular products is “84077” and least popular is “84347”.**

****

1. **Sales by country:** The country field can be used to determine which countries generate the most revenue for the store. This information can be used to target marketing efforts in those countries.

sales\_by\_country = retail\_store.groupby("Country")["Quantity"].sum().sort\_values(ascending=False)

print(sales\_by\_country)

**Most of the sales is coming from United Kingdom and least sales coming from Saudi Arabia.**





1. **Customer behaviour:** By analysing the quantity and total cost fields, patterns in customer behaviour can be identified such as how often they make purchases and how much they spend on average.

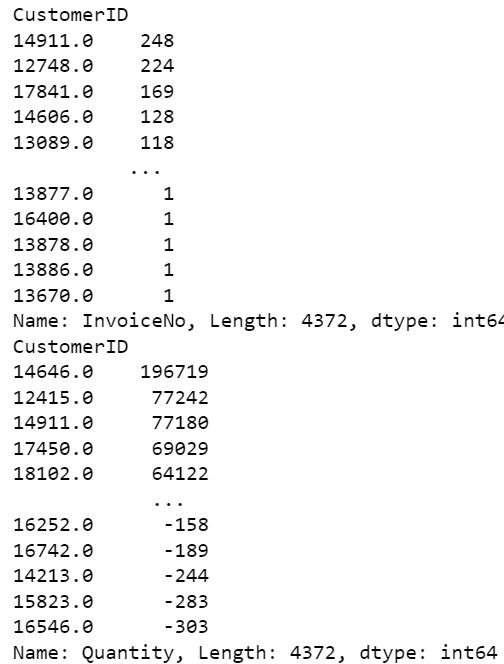
purchase\_frequency = retail\_store.groupby("CustomerID")["InvoiceNo"].nunique().sort\_values(ascending=False)

average\_spend = retail\_store.groupby("CustomerID")["Quantity"].sum().sort\_values(ascending=False)

print(purchase\_frequency)

print(average\_spend)

**Customer “14911” is the most frequent customer and most spent is being done by customer “196719”.**



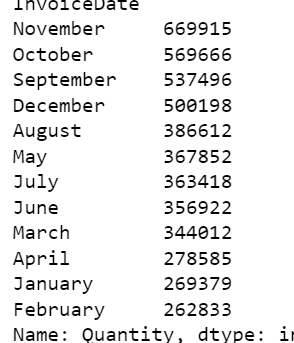
1. **Time-based analysis:** By analysing the date field, patterns in sales can be identified over time. This information can be used to predict future sales trends and plan inventory accordingly.

retail\_store['InvoiceDate'] = pd.to\_datetime(retail\_store['InvoiceDate'])

sales\_by\_month = retail\_store.groupby(retail\_store['InvoiceDate'].dt.strftime('%B'))['Quantity'].sum().sort\_values(ascending=False)

print(sales\_by\_month)

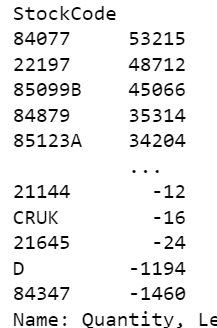
**Sales is highest in November and lowest in February.**



1. **Stock forecasting:** By analyzing the sales data, patterns in stock demand can be identified. This information can be used to predict future stock needs and plan inventory accordingly.

stock\_forecast = retail\_store.groupby("StockCode")["Quantity"].sum().sort\_values(ascending=False)

print(stock\_forecast)

****

### 2. **Customer** **Segmentation**

retail\_store["segment"] = "low value"

retail\_store.loc[retail\_store["UnitPrice"] >= 50, "segment"] = "high value"

purchase\_counts = retail\_store.groupby("CustomerID")["UnitPrice"].count()

purchase\_counts = purchase\_counts.reset\_index()

purchase\_counts.columns = ["CustomerID", "purchase\_counts"]

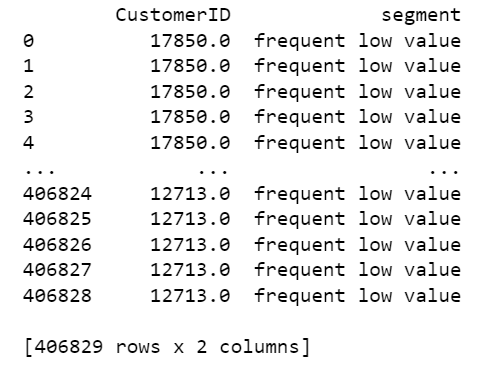
retail\_store = pd.merge(retail\_store, purchase\_counts, on="CustomerID")

retail\_store.loc[retail\_store["purchase\_counts"] >= 10, "segment"] = "frequent" + " " + retail\_store["segment"]

retail\_store.loc[retail\_store["purchase\_counts"] < 3, "segment"] = "infrequent" + " " + retail\_store["segment"]

print(retail\_store[["CustomerID", "segment"]])

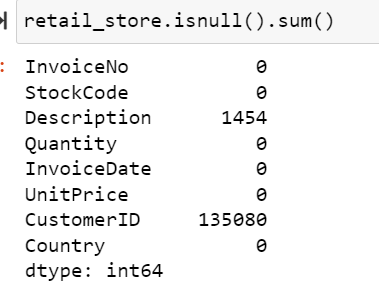
**Output**:



# Data pre-processing Steps and Inspiration

1. Data cleaning- The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc

Here, we have huge percentage approx. (25%) of CustomerID which is blank.



1. Outliers- Identifying and removing outliers

import seaborn as sns

fig, axs = plt.subplots(2,figsize=(6,18))

X = retail\_store[['UnitPrice']]

for i,column in enumerate(X):

sns.boxplot(retail\_store[column], ax=axs[i], width=0.8)

# **drop the outliers**

without\_outlier = retail\_store[(retail\_store['UnitPrice']>5000)]

without\_outlier

1. Data integration: Combining multiple data sources- Here, there is only one source i.e. retail\_store.csv
2. Data transformation: Normalizing or aggregating data- We will do this during model building.
3. Data reduction: Reducing the amount of data through sampling or feature selection- We have train and test set.
4. Data encoding: Encoding categorical data as numerical data to be used in machine learning models.

# Choosing the Algorithm for the project

I have chosen 2 algorithm 1) K-means 2) Hierarchical- clustering

1. **K-means-**

Below code first selects the relevant features for clustering, which are "Quantity" and "UnitPrice" in this example. Then it standardizes the data by subtracting the mean and dividing by the standard deviation to ensure that the algorithm is not affected by the scale of the data. Then it runs the K-means algorithm with 3 clusters, which is the number of clusters we want to segment the customers into.

The labels generated by the algorithm are then added to the dataframe as segment column, and the final segmented data containing CustomerID and segment is printed.

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# select relevant features for clustering

features = retail\_store[["Quantity", "UnitPrice"]]

# standardize the data

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(features)

# run k-means clustering

kmeans = KMeans(n\_clusters=3)

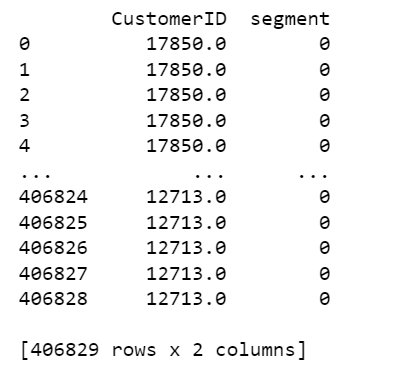
kmeans.fit(scaled\_data)

# assign cluster labels to customers

retail\_store["segment"] = kmeans.labels\_

# print the segmented data

print(retail\_store[["CustomerID", "segment"]])



1. **Hierarchical clustering**

from sklearn.cluster import AgglomerativeClustering

from sklearn.preprocessing import StandardScaler

# select relevant features for clustering

features = retail\_store[["Quantity", "UnitPrice"]]

# standardize the data

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(features)

# run hierarchical clustering

agg\_clustering = AgglomerativeClustering(n\_clusters=3)

agg\_clustering.fit(scaled\_data)

# assign cluster labels to customers

retail\_store["segment"] = agg\_clustering.labels\_

# print the segmented data

print(retail\_store[["CustomerID", "segment"]])

# Motivation and Reasons for choosing the Algorithm

The motivation for choosing k-means algorithm in this case could be because:

1. K-means is a popular algorithm for customer segmentation because it is simple to implement and easy to interpret. It is a form of unsupervised learning, which means it does not require labeled data, so it can be used with datasets that do not have predefined groups.
2. One of the main advantages of K-means is that it is computationally efficient and can handle large datasets. It is also a versatile algorithm that can be used for various types of data such as continuous, categorical, or mixed data.
3. Another advantage of K-means is that it produces clear and well-defined clusters, which makes it easy to interpret the results and assign customers to specific segments. This can be useful for creating targeted marketing campaigns, improving customer service, and optimizing pricing strategies.
4. In addition, K-means is a hard clustering algorithm, which means each data point belongs to only one cluster. This can be useful when we have a clear idea of how many segments we want to create and when we want to assign each customer to a specific segment.

In summary, K-means is a powerful algorithm that can be used to segment customers based on their purchase history, demographics, or other characteristics. It is computationally efficient, easy to interpret, and produces clear and well-defined clusters. This makes it a good choice for customer segmentation when you have a clear idea of how many segments you want to create and when you want to assign each customer to a specific segment.

# Assumptions

The k-means algorithm makes the following assumptions:

1. The data points are homogeneous and isotropic, meaning that they have similar variances and are evenly distributed in all directions.
2. The clusters are spherical and have similar variances.
3. The number of clusters (k) is known in advance.
4. The data points are independent and identically distributed.
5. The algorithm converges to the global optimum, although this is not guaranteed in practice.

# Model Evaluation and techniques

Model evaluation and techniques are important steps in the process of building and using a machine learning model. They help to ensure that the model is accurate, reliable, and generalizable to new data.

For evaluating a k-means model used for customer segmentation based on purchasing history, some common techniques include:

1. Silhouette Score: It measures how similar an object is to its own cluster compared to other clusters. The score ranges from -1 to 1, with higher values indicating a better fit.
2. Calinski-Harabasz Index: It measures the ratio of the between-cluster variance to the within-cluster variance. A higher value indicates a better fit.
3. Davies-Bouldin Index: It measures the average similarity between each cluster and its most similar cluster. Lower values indicate a better fit.
4. Elbow method: It is used to determine the optimal number of clusters. The method plots the within-cluster sum of squares (WCSS) against the number of clusters and the optimal number of clusters is chosen at the "elbow" point of the plot.
5. Visualization: Visualizing the data points and the clusters can also provide insights into the performance of the model and help identify any issues or areas for improvement.

# Inferences from the same

From the evaluation of a k-means model can provide valuable insights into the performance of the model and the characteristics of the segments. Here are some possible inferences that can be drawn from the evaluation of a k-means model:

1. High silhouette score or Calinski-Harabasz index and low Davies-Bouldin index indicate that the model has found well-defined and distinct clusters.
2. Low silhouette score or Calinski-Harabasz index and high Davies-Bouldin index indicate that the model has found overlapping or poorly defined clusters.
3. A high value of WCSS (Within-cluster sum of squares) suggests that the clusters are far away from the centroid and data points are scattered around the centroid.
4. A low value of WCSS suggests that the clusters are tightly packed around the centroid and data points are concentrated around the centroid.
5. Visualization of the clusters can help identify any outliers or anomalies in the data, as well as any patterns or trends in the data that may not be apparent from the numerical evaluation metrics.
6. Cluster completeness and purity can be used to evaluate the quality of the clusters. High completeness and purity imply that the cluster is composed of similar data points and different clusters are composed of different data points.
7. Adjusted Rand index can be used to compare the similarity of two different partitions of the same data set. A high index value indicates that the two partitions are similar, while a low value indicates that they are dissimilar.
8. The number of clusters chosen by the Elbow method can be used to determine the optimal number of clusters, which can then be used as a guide for future analyses.

# Future possibilities of the project

There are several future possibilities for an online retail store using k-means:

1. Personalized Recommendations: By segmenting customers based on their purchasing history, an online retail store can offer personalized recommendations to each group of customers, increasing the likelihood of repeat purchases.
2. Targeted Marketing: By understanding the characteristics of different customer segments, an online retail store can target specific groups of customers with relevant marketing messages and promotions, increasing the efficiency and effectiveness of their marketing efforts.
3. Inventory Management: By identifying patterns in customer purchasing behavior, an online retail store can anticipate the demand for certain products and optimize their inventory management to avoid stockouts or overstocking.
4. Up-selling and cross-selling: By understanding the purchase patterns of different segments, an online retail store can recommend related or complimentary products to customers, increasing the average order value.
5. Fraud Detection: By identifying patterns of abnormal behavior, an online retail store can detect and prevent fraudulent activities.
6. Churn Prediction: By identifying patterns of behavior that are associated with customers who are likely to stop shopping, an online retail store can take proactive steps to retain these customers before they leave.
7. Customer Lifetime Value prediction: By understanding the behavior patterns of different customer segments, an online retail store can predict the lifetime value of each customer and accordingly design retention and acquisition strategies.
8. Customer Segmentation can also be used to identify the most profitable customer segments and focus on retaining and acquiring more customers in those segments.

# Conclusion:

In this project, using k-means for customer segmentation in an online retail store can provide valuable insights into customer behaviour and enable the store to personalize its offerings, improve its marketing efforts, optimize its inventory management, detect fraud, and predict churn.

By understanding the characteristics of different customer segments, an online retail store can make data-driven decisions that lead to increased sales and customer satisfaction.

Additionally, k-means can also be used to predict customer lifetime value and identify profitable customer segments, allowing the store to focus on retaining and acquiring more customers in those segments.

However, it's important to keep in mind that k-means is an unsupervised learning algorithm, so prior knowledge of the dataset is needed to make sure the assumptions are met and the results are meaningful. References

* Intellipaat live session recording & notes
* Wikipedia
* Google.com
* https://www.analyticsvidhya.com/blog/2021/05/k-means-clustering
* Geeksforgeeks.com
* https://towardsdatascience.com/customer-segmentation-using-k-means-clustering