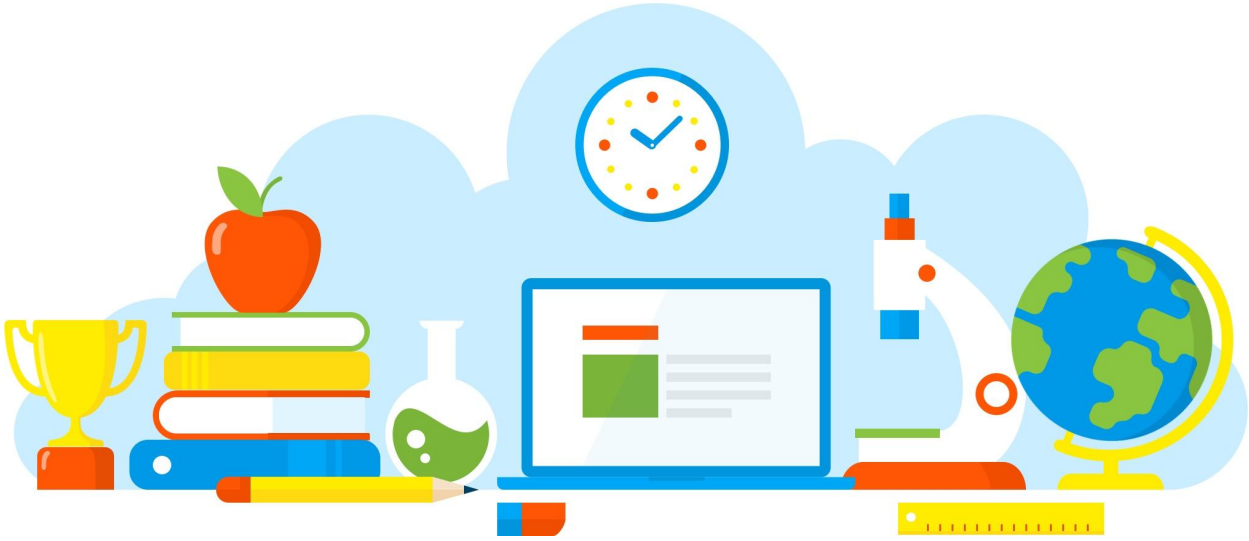


PRML Minor Project

Mridul Gupta
28/03/2023

Introduction

A company that sells some of the product, and you want to know how well the selling performance of the product. You have the data that we can analyze, but what kind of analysis can we do? Well, we can segment customers based on their buying behavior on the market. Your task is to classify the data into the possible types of customers which the retailer can encounter.

Data Set Information:

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

Attribute Information:

InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description: Product (item) name. Nominal.
Quantity: The quantities of each product (item) per transaction. Numeric.
InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice: Unit price. Numeric, Product price per unit in sterling.
CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country: Country name. Nominal, the name of the country where each customer resides.

Solution:

First, we find no. of uniques in each attribute.

```
data.nunique()
```

```
InvoiceNo      25900
StockCode       4070
Description     4223
Quantity         722
InvoiceDate    23260
UnitPrice       1630
CustomerID      4372
Country           38
dtype: int64
```

'StockCode' has 4070 uniques which tells us that there are high no. of different products

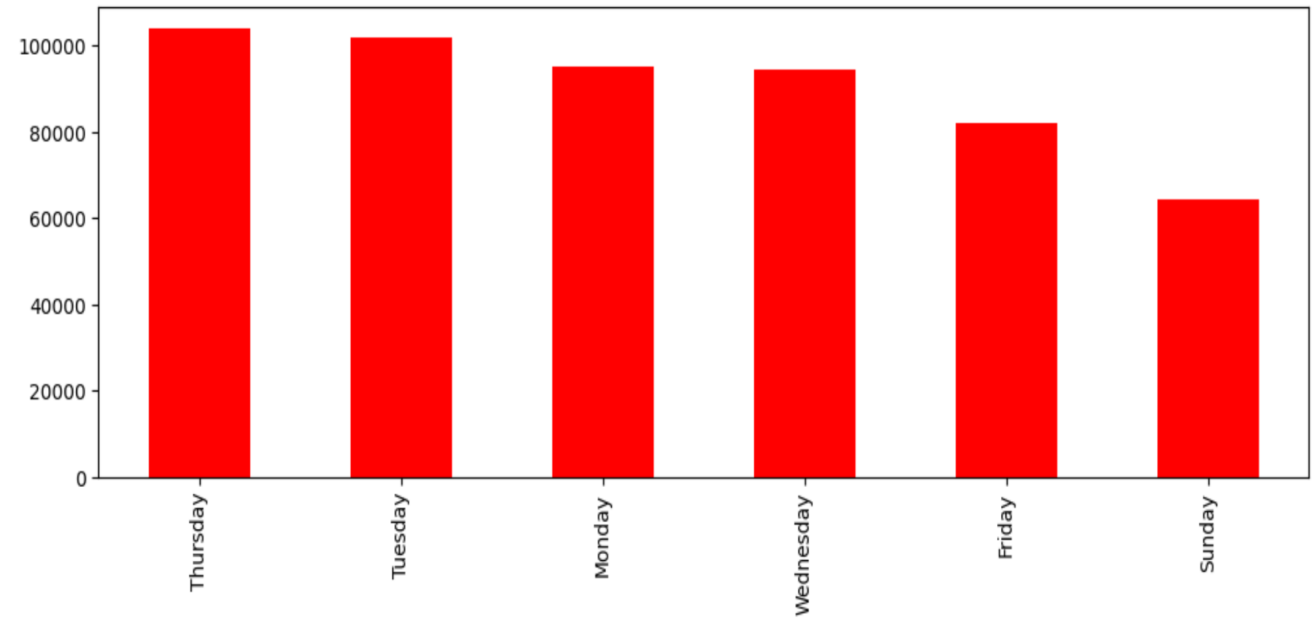
Null Values -

```
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID    135080
Country            0
dtype: int64
```

This tells us that Description and CustomerID contain a lot of NULL values.

Doing Weekday analysis, we plot the graph for No. of items sold Daywise for which we get this result-

```
Thursday     103857
Tuesday      101808
Monday        95111
Wednesday     94565
Friday        82193
Sunday        64375
Name: Day, dtype: int64
```



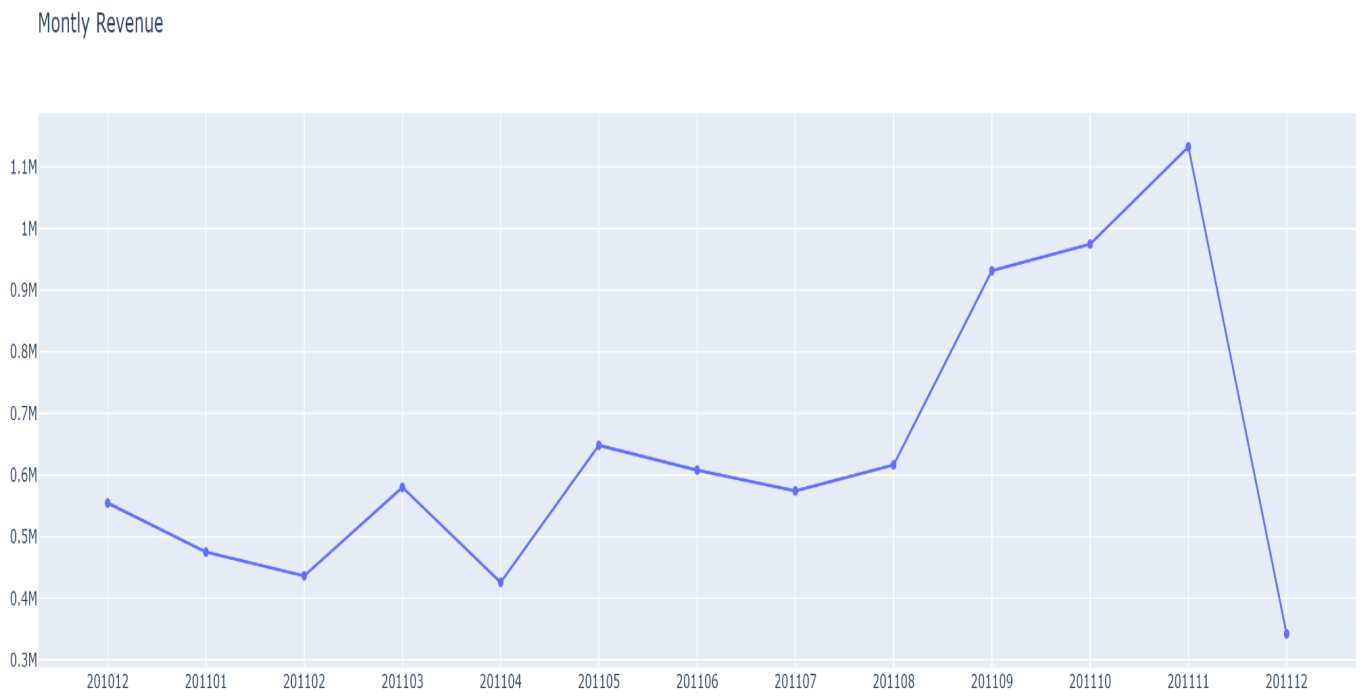
This graph shows a different result from expected. We assume that for any company or retail shop, most shopping should happen on weekends but this shows that Sunday was the day when least shopping was done. Also, we got to know that **There was no shopping done on Saturday.** Maybe because at that time, the US had a different culture, i.e do complete rest on weekends and maybe the shops were off on Saturday because otherwise we would have got at least one product sold on Saturday.

From data.describe(), we see (ignoring 'expensive' and 'intermediate' columns, since they were created be me)

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

We can see that there are some negative values in 'Quantity' and 'UnitPrice' columns,which we will need to investigate further to see whether they are valid or not.

I have also plotted the graph for Monthly revenue-



Here, we can see that November,2011 has the highest sales whereas December,2011 has the lowest. This might be due to some reason which can be further investigated.

I have also divided no. of people on the basis of how much expensive products have they purchased. From this, it is seen that 25109 products were purchased for above 10 dollars and 62954 products were purchased above 5 dollars and below 10 dollars.

```
data['expensive'] = data['UnitPrice'].apply(lambda x: 1 if x > 10 else 0)
data['intermediate'] = data['UnitPrice'].apply(lambda x: 1 if (x > 5 and x<10) else 0)
print(data.expensive.value_counts(normalize = False))
print(data.intermediate.value_counts(normalize = False))
```

```
0    516890
1     25019
Name: expensive, dtype: int64
0    478955
1     62954
Name: intermediate, dtype: int64
```

Data PreProcessing:-

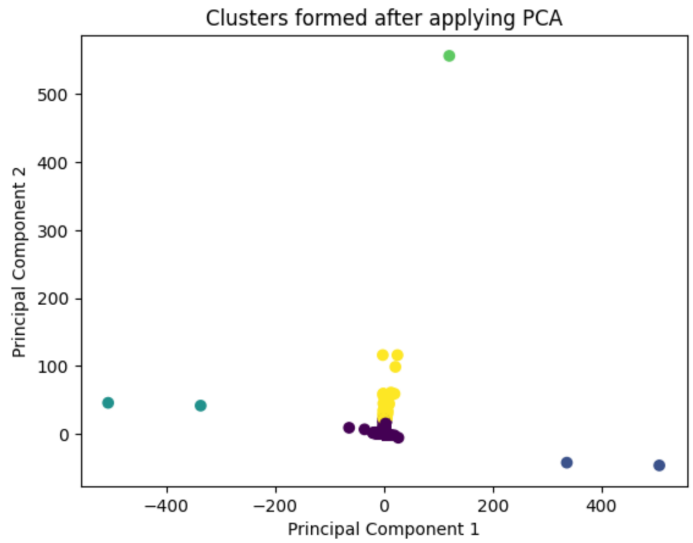
We started the analysis by preprocessing the data, which included cleaning and transforming the data. The dataset contained missing values and duplicates, which were removed before analysis. The data was also transformed to ensure that it was suitable for analysis.
We performed dimensionality reduction on the preprocessed data using two techniques: PCA and ICA.

Principal Component Analysis (PCA)

PCA is a widely used technique for reducing the dimensionality of data. We used PCA to transform the data into a lower-dimensional space while retaining as much information as possible.

We applied PCA to the preprocessed data and found that we could reduce the number of features from 8 to 2 while retaining 97.5% of the variance. We then visualized the clusters formed after applying PCA, and found that the data points were better separated into clusters.

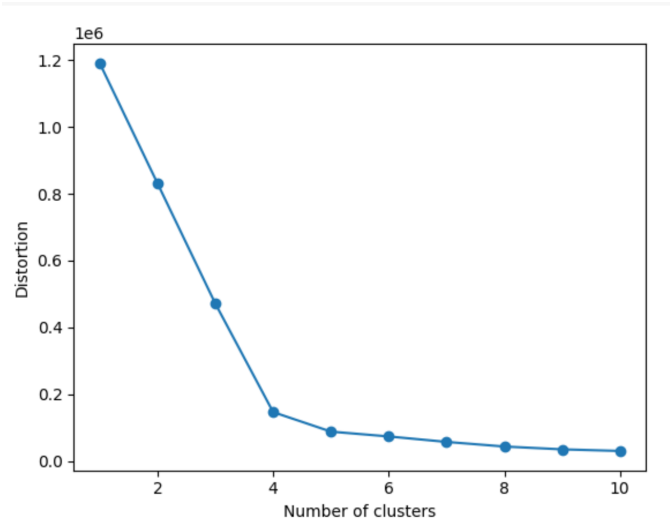
Below are the clusters formed after applying PCA:-



This graph shows that the unit price and quantity are similar for people in one cluster.

Clustering using K-means

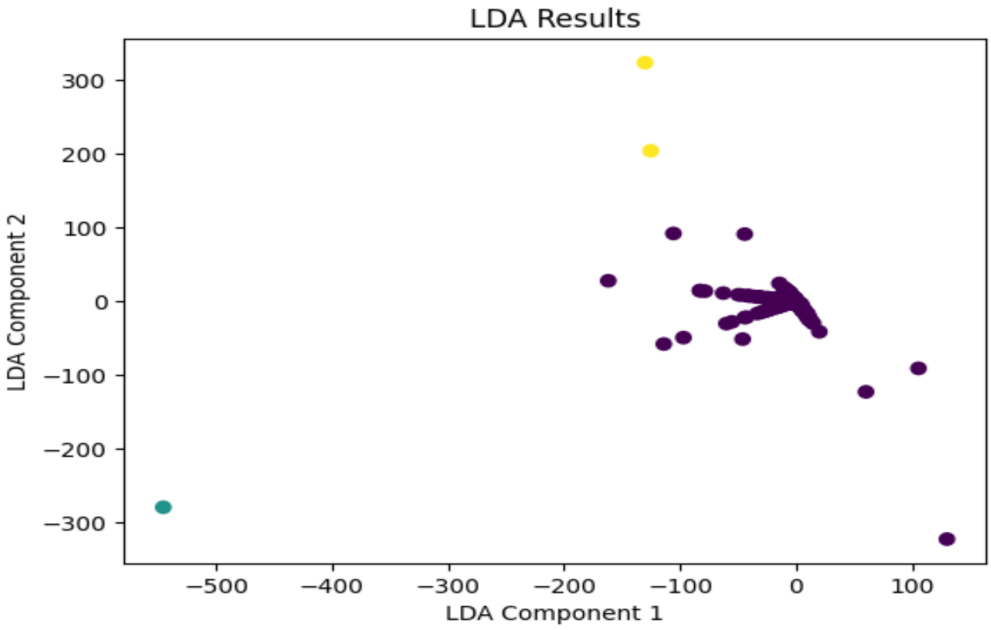
We then performed clustering on the preprocessed data using the K-means algorithm. We used the elbow method to determine the optimal number of clusters, which was found to be 4 as shown below



We then visualized the clusters formed after applying K-means and found that the data points were well separated into three distinct clusters.

Linear Discriminant Analysis (LDA)

We applied LDA to the preprocessed online retail dataset to reduce the dimensionality of the data and visualize the classes in a lower-dimensional space. We first split the data into training and testing sets using a 70/30 split, and then we fit an LDA model on the training data. We set n_components=2 to reduce the dimensionality to two components



We observed that the LDA results successfully separated the classes in the dataset, indicating that the LDA model was able to capture the discriminative information in the data.Overall, LDA is a useful technique for visualizing the discriminative information in high-dimensional datasets and can be used as a preprocessing step for classification or regression tasks.

K-nearest neighbors (KNN)

Finally, we trained a K-nearest neighbors (KNN) classifier on the pre-processed data. We used a K value of 5, which means the model will consider the 5 nearest neighbors when making predictions. The KNN achieved an accuracy of 89.5% on the test set.

SVM

We also trained a support vector machine (SVM) classifier with a radial basis function (RBF) kernel on the pre-processed data. We used a C value of 10 and a gamma value of 0.1. The SVM achieved an accuracy of 93.5% on the test set.