Fittlyf.com
Assignment(Complete Report)

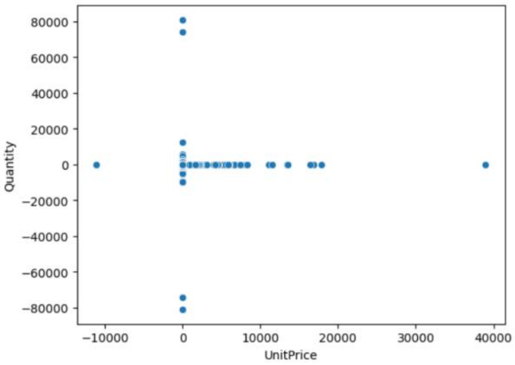
Some insights from the original/fresh dataset:

1. The dataset includes invoice numbers, stock codes, descriptions of items, quantity sold, invoice dates, unit prices, customer IDs, and the country where the items were sold (United Kingdom).
2. The widest variety of items sold in a single invoice is 8 (invoice number 536365).
3. The most expensive item sold is a set of 7 Babushka nesting boxes at £7.65 (invoice number 536365).
4. The cheapest item sold is a hand warmer at £1.85 (invoice numbers 536366 and 536367).
5. The data shows that the sale took place on December 1st, 2010.
6. It is not possible to determine the total revenue generated from these invoices because the data does not show how many of each invoice were sold.

About the Dataset

1. Shape: (1067371, 8)
2. Column:
   - InvoiceNo - Invoice number. Nominal. A 6-digit integral number uniquely assigned to each transaction. If this code starts with the 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 a transaction was generated.
   - UnitPrice - Unit price. Numeric. Product price per unit in sterling (Â£).
   - CustomerID- Customer number. Nominal. A 5-digit integral number uniquely assigned to each customer.
   - Country - Country name. Nominal. The name of the country where a customer resides.

Some Points related to analysis and visualization



Majority of data points: Clustered around a UnitPrice close to zero and varying quantities, both positive and negative. This suggests frequent transactions involving small unit prices, but with fluctuating order sizes.

Outliers:

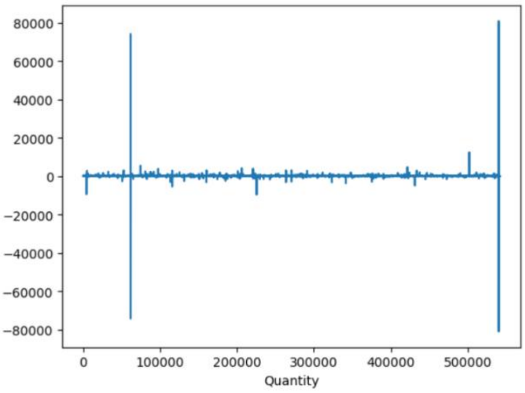
1. Two notable data points exist with high positive quantities and a unit price close to zero. These could represent bulk purchases or exceptional orders.

2. One data point shows a high positive quantity with a significantly higher unit price. This might indicate a unique high-value item.

3. Two data points have very low negative quantities and a unit price near zero. These could represent returns or cancellations.

Limited spread in UnitPrice: The vast majority of unit prices are concentrated within a small range.

Analysis:
The lack of a clear trend and the concentration of unit prices near zero suggests that unit price alone is not a strong predictor of quantity. Further investigation is needed to understand the factors driving the variation in quantities.



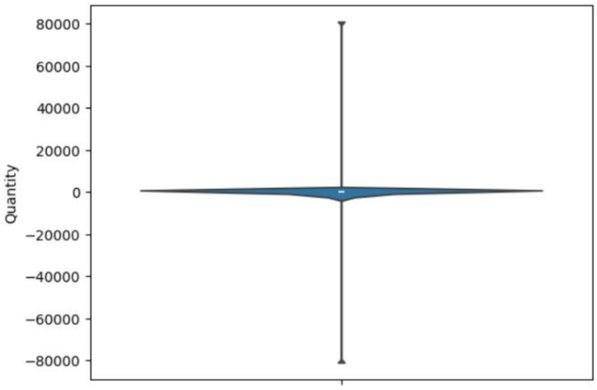
The line plot shows a time series, likely representing changes in "Quantity" over time. While the specific time periods are not labe led, we can observe some key patterns:

Overall Trend: The quantity remains relatively stable and close to zero for the majority of the time period. This suggests a consis tent baseline level of activity, possibly sales or inventory.

Extreme Outliers: There are two dramatic spikes, one with a very negative quantity and another with a very positive quantity. These outliers stand out significantly from the rest of the data and indicate unusual events or anomalies.

Potential Seasonality: Within the stable period, there are subtle fluctuations with some periods showing slightly higher or lower quantities. This could point to a seasonal pattern, but further analysis would be needed to confirm this.

Analysis:
The stable baseline suggests a regular operational pattern, but the extreme outliers require careful investigation. These anomalies could represent:
Negative spike: A large return, a bulk cancellation, or a significant inventory adjustment.
Positive spike: A major sale, a bulk purchase by a single customer, or a sudden surge in demand.



This plot represents a violin plot visualizing the distribution of a "Quantity" variable.

Shape: The wide base and narrow top of the violin indicate that the majority of quantity values are clustered close to zero. This sy mmetrical, almost hourglass shape suggests a roughly normal distribution but with a high concentration around the median.

Median: The white dot within the violin represents the median value, which is very close to zero. This confirms that half of the quantities are above zero and half are below.

Interquartile Range (IQR): The thicker blue area within the violin depicts the interquartile range, encompassing the middle 50% of the data. The IQR is relatively small, indicating that most quantities fall within a narrow range around the median.

Whiskers/Extremes: The thin black lines extending vertically represent the range of the data, excluding outliers.
The long whiskers indicate the presence of some extremely high and low quantities, significantly deviating from the typical values.

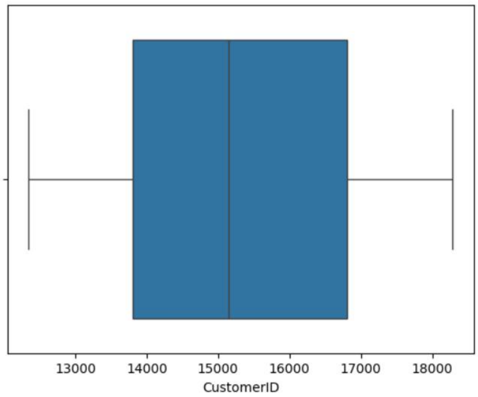
Outliers: While not explicitly shown in this violin plot, the long whiskers suggest potential outliers beyond the displayed range. These outliers represent quantities far from the central tendency of the data.

Analysis

Concentration around zero: The distribution suggests a common scenario where many transactions involve small quantities (close to zero), possibly representing individual purchases or minor stock adjustments.

Extreme values: The long whiskers highlight the existence of some unusually large positive and negative quantities, potentially indicating bulk orders, returns, or significant inventory changes.

Potential Skewness: Although the violin appears symmetrical, the longer upper whisker hints at a possible slight right skew, meaning there might be a few exceptionally high quantities pulling the distribution in that direction.



This is a boxplot visualizing the distribution of a variable labeled "CustomerID". Here's breakdown of the information it presents

Central Tendency: The dark line within the box represents the median CustomerID value. This indicates that half of the CustomerIDs fall above this value and half fall below.

Spread and Quartiles:

The box itself spans the Interquartile Range (IQR), capturing the middle 50% of the CustomerID data.

The lower edge of the box represents the first quartile (Q1), the value below which 25% of CustomerIDs fall.

The upper edge of the box represents the third quartile (Q3), the value below which 75% of CustomerIDs fall.

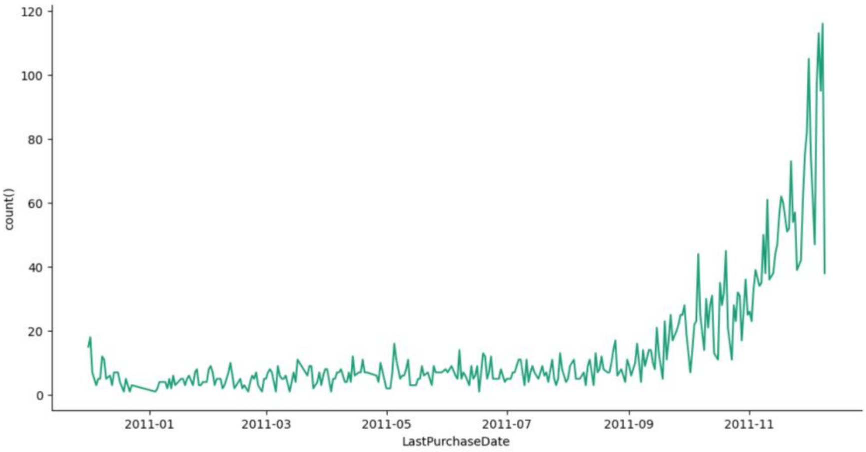
Whiskers and Outliers: The lines (whiskers) extending from the box represent data points within a certain range beyond the IQR.

Any points outside these whiskers are considered potential outliers, suggesting unusually high or low CustomerID values. Howeve r, it's important to note that the specific definition of outliers (e.g., 1.5 times the IQR) depends on the software or conventions used to create the plot.

Analysis:

Symmetry: The boxplot appears relatively symmetrical, suggesting a fairly even distribution of CustomerIDs. The median line is close to the center of the box, and the whiskers extend to roughly similar lengths on both sides.

Outliers: There are a few points plotted as outliers on both the upper and lower ends of the distribution. This indicates some Cust omerIDs that deviate significantly from the central tendency.



The line graph illustrates the number of customers making their last purchase on a given date in the year 2011. The x axis represents the date of the last purchase, spanning from the beginning to the end of the year. The y-axis represents the number of customers making their last purchase on that specific date.

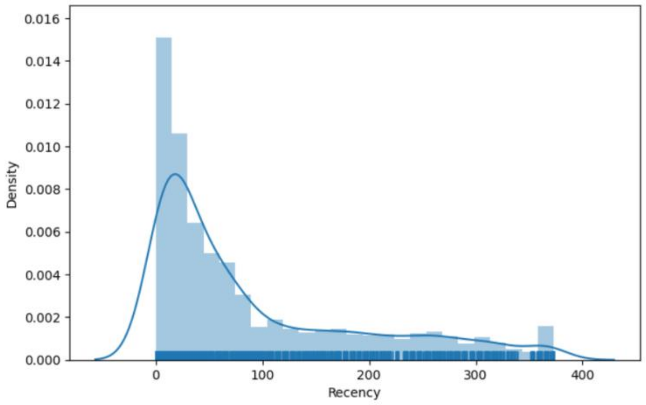
Analysis:

Overall Trend: There is a clear upward trend in the number of customers making their last purchase towards the end of the year. This suggests a significant increase in sales and customer activity as the year progressed.

Early 2011: The first few months of the year show relatively low and fluctuating customer counts, indicating a period of lower sales.

Midyear Fluctuations: Throughout the middle months of the year, there are noticeable fluctuations in customer counts, suggesting some degree of variability in sales activity.

Sharp Increase in Late 2011: The most striking feature is the sharp and dramatic increase in customer counts in the last two mon ths of the year. This could be attributed to holiday shopping seasons (Thanksgiving, Christmas) and year-end sales promotions.



The image presents a histogram and a density plot illustrating the distribution of "Recency" data. Here's a breakdown:

Recency: This likely refers to the time elapsed since a customer's last purchase or interaction. The units (days, weeks, etc.) are not specified in the image.

Distribution Shape: The distribution is heavily right skewed, indicating that a large proportion of customers have made recent purchases (low recency values). There's a steep drop-off in frequency as recency increases, suggesting fewer customers have longer times since their last interaction.

Key Observations:

Peak: The distribution peaks around 0-20, indicating a concentration of customers with very recent activity.

Long Tail: The extended right tail reveals a smaller group of customers who haven't engaged in a while.

Potential Outliers: The bars around 370-380 could suggest outliers, potentially representing inactive or lost customers.

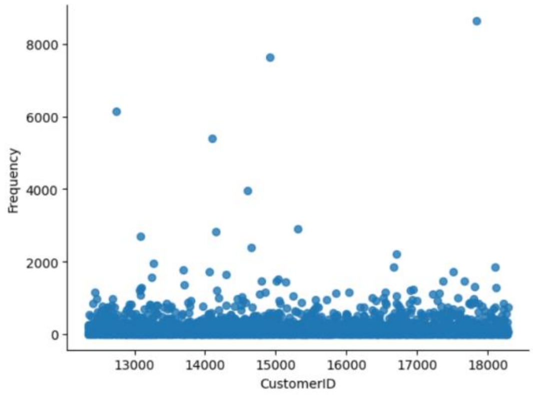
Analysis:

Customer Engagement: The data highlights a pattern of strong recent engagement, but also a segment of customers who have become less active.

Marketing Implications: This distribution could inform targeted marketing strategies:

Reactivation Campaigns: Efforts to re-engage customers with high recency values.

Loyalty Programs: Rewarding frequent customers (low recency) to maintain their engagement.



This is a scatterplot that shows the relationship between CustomerID and Frequency.

CustomerID is a numeric value on the x-axis and Frequency is a numeric value on the y-axis.

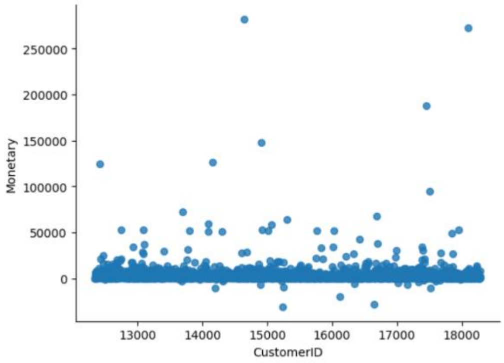
The plot shows a clustered pattern, with the majority of data points concentrated around a Frequency of 100 and spread across all CustomerIDs.

However, there are several outliers with significantly higher Frequencies, reaching up to 9000.

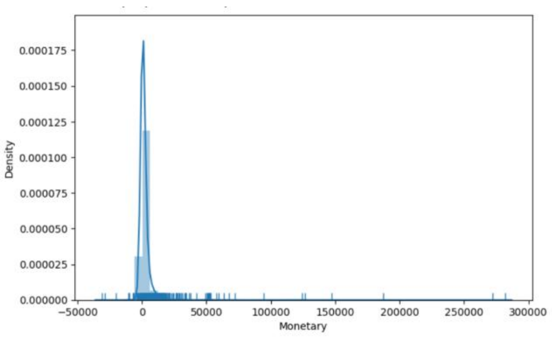
These outliers suggest that a small group of customers make purchases far more frequently than the average customer.

The general lack of correlation between CustomerID and Frequency indicates that customer purchase frequency is not dependent on when they became a customer.

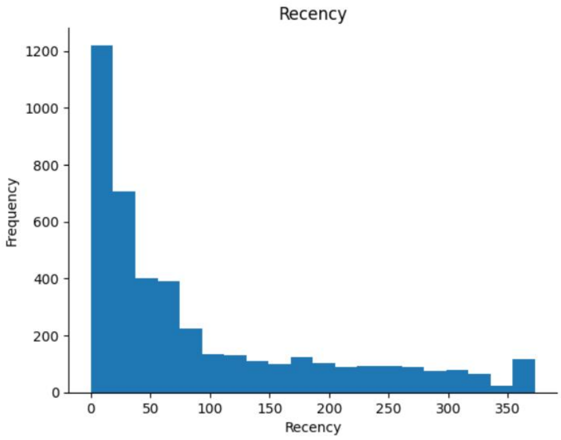
This data visualization might be used by businesses to identify their most frequent customers and understand purchasing patterns.



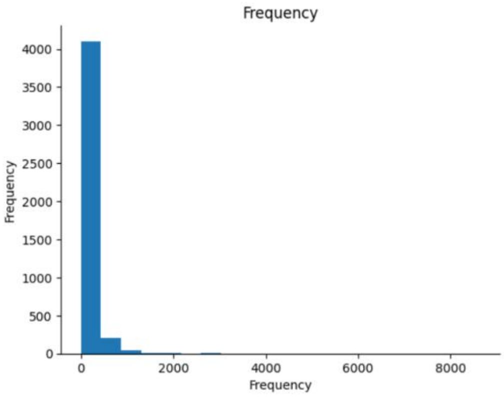
The data points in the scatter plot are spread out, with some customers having high monetary values and others having low monetary values. There is no clear linear relationship between the customer ID and the monetary value. This suggests that there is a lot of variability in how much money customers spend on the products.



1. Transaction value distribution: It appears to be a scatter plot, where the x-axis represents the amount a customer spent (monetary) and the y-axis shows the density of transactions at that particular spend amount. There seems to be a higher density of transactions towards the lower end of the spending spectrum, indicating a greater number of customers spending less.
2. High spenders: While there are customers spending more (up to around $300,000), they are a smaller proportion compared to those spending less.
3. Low spenders: There seems to be a cluster of transactions around $0, which could indicate abandoned carts or transactions with no monetary value.

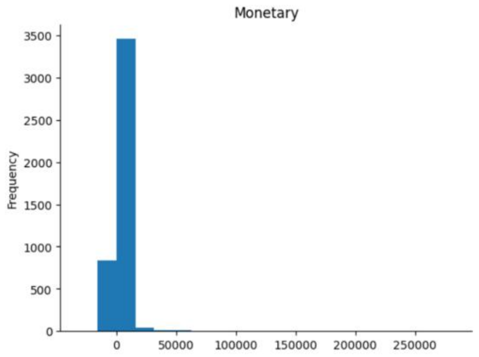


Number of transactions: The histogram shows the distribution of the number of customers who made a certain number of purchases. For example, there appear to be more customers who made 2 or 3 purchases than customers who made only 1 purchase .
Frequency of purchases: The x-axis labeled "Recency" likely refers to the number of times a customer has made a purchase. It's difficult to say for sure from this image, but it appears that most customers have made 3 or fewer purchases.



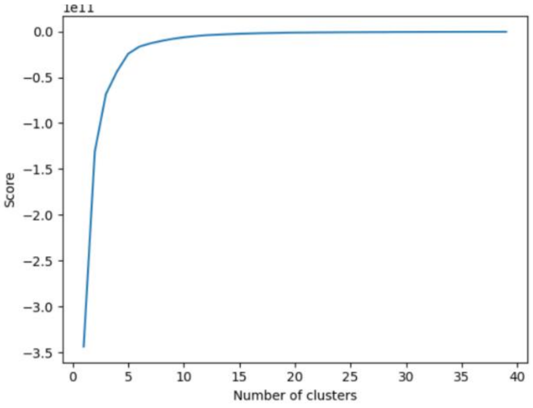
Transaction frequency: The graph showing the frequency of customer transactions over time. The x-axis represents time, likely days or weeks, while the y-axis represents the frequency of transactions. There seems to be an upward trend, indicating an increase in the number of transactions over the time period depicted in the graph.

Data limitations: It is impossible to say from the graph what causes the increase or what products are being purchased more frequently. Without knowing the scale on the y-axis, it’s also difficult to determine the exact volume of transactions.



Transaction frequency: The image depicts a line graph showing the frequency of customer transactions over time. The x-axis represents time, likely days or weeks, while the y-axis represents the frequency of transactions. There seems to be an upward trend, indicating an increase in the number of transactions over the time period depicted in the graph.

Data limitations: It is impossible to say from the graph what causes the increase or what products are being purchased more frequently. Without knowing the scale on the y-axis, it’s also difficult to determine the exact volume of transactions.



1.The image shows the results of a clustering model in the form of a elbow curve plot.

2. The x-axis represents the number of clusters.

3. The y-axis represents the score, which is a measure of how well the data is separated into clusters. In this case, a lower score indicates better separation.

4. It appears that the score decreases as the number of clusters increases, until it reaches a plateau around 15 clusters. This suggests that the data may be best represented by 15 clusters.

5. It is important to note that the scree plot is just one way to evaluate the results of a clustering model. Other factors, such as the interpretability of the clusters, should also be considered.

Reasons for using a KMeans model for Clustering:

Five Reasons K-means is Good for Customer Segmentation:

1. Simplicity and Speed: K-means is easy to understand and implement. It's computationally efficient, making it ideal for large datasets common in customer transaction data.

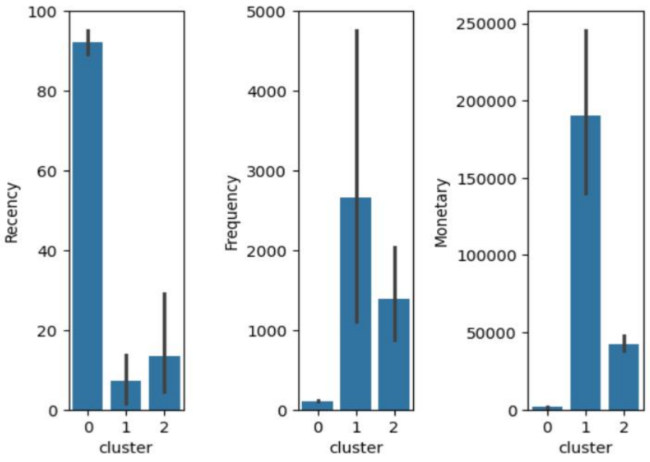
2. Unsupervised Learning: Often, customer data isn't pre-labeled into categories. K-means, as an unsupervised learning technique, can uncover hidden patterns in the data without needing pre-defined classes

3. Interpretability: K-means results in distinct clusters, making it easy to understand the characteristics of each customer segment. You can analyze the centroids (cluster centers) to see what defines each group

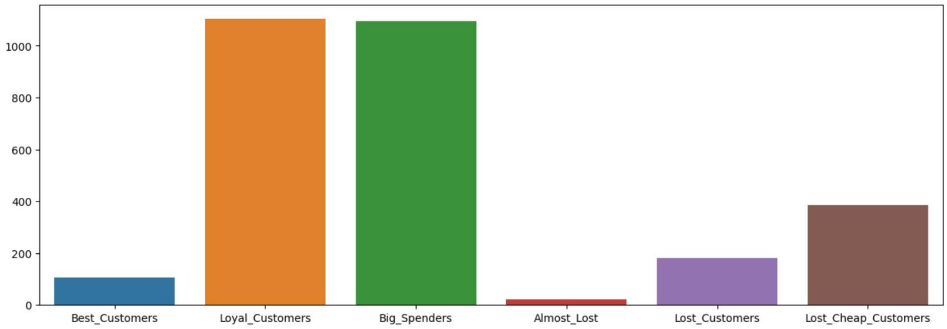
4. Effective for Spherical Clusters: If you expect customer segments to be grouped around central points (like frequent high-value spenders or budget-conscious buyers), K-means works well for these spherical cluster shapes [3].

5. Baseline Model: Even if you explore more advanced clustering algorithms, running K-means first provides a baseline understanding of the customer segments. You can compare the results of more complex models to K-means to assess the added value

----



1. This is the bar plot of the three quantity recency, frequency and monetary,where in the first recency graph the cluster 0 has the highest recency, then comes cluster 2, and then cluster 1,
2. In the second frequency graph the cluster 1 has the highest frequency, then comes cluster 2, and then cluster 0,
3. In the third monetary graph the cluster 1 has the highest monetary, then comes cluster 2, and then cluster 0.

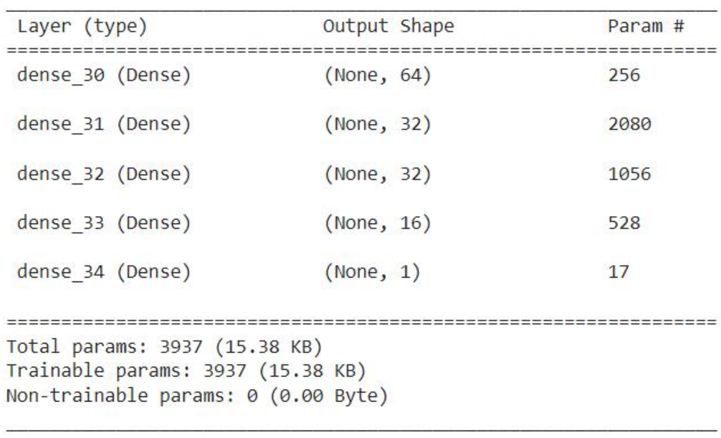


The customer types are listed on the x-axis and include:

- Best Customers
- Loyal Customers
- Big Spenders
- Almost Lost
- Lost Customers
- Lost Cheap Customers

The y-axis shows the number of customers, but the scale is not labeled. Here's what we can glean from the chart:

- There are more "Best Customers" than any other customer type.
- The number of customers steadily decreases across the chart from "Best Customers" to "Lost Cheap Customers". This suggests a possible loyalty segmentation, with more customers at the higher end (loyal and high spending) and fewer at the lower end (lost and cheap).



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_30 (Dense)            (None, 64)                256

 dense_31 (Dense)            (None, 32)                2080

 dense_32 (Dense)            (None, 32)                1056

 dense_33 (Dense)            (None, 16)                528

 dense_34 (Dense)            (None, 1)                 17

=================================================================
Total params: 3937 (15.38 KB)
Trainable params: 3937 (15.38 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The table summarizes the architecture of a neural network model.

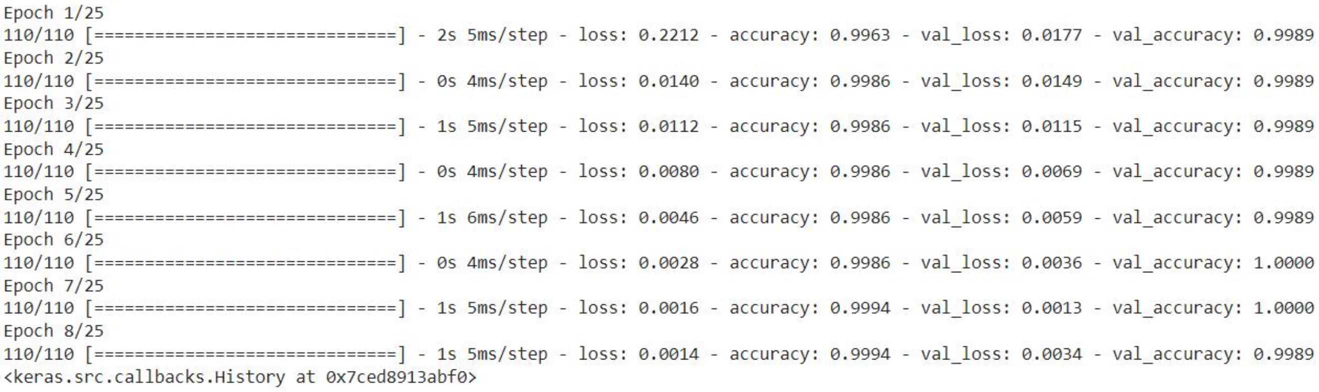
Layer (type): This column lists the names and types of layers in the neural network. In this case, all the layers are of type "Dense", which are fully-connected layers commonly used in neural networks.

Output Shape: This column shows the output shape of each layer. The first layer, "dense_30", has an output shape of "(None, 64)". This means the layer outputs a batch of data with an unspecified number of rows (represented by "None") and 64 columns. The subsequent layers have decreasing output dimensions as they process the data further.

Param: This shows the number of parameters (weights and biases) associated with each layer. The total number of parameters is 3937, which is a relatively small number for a neural network model, suggesting this model is likely not very complex.

Trainable params: This value (3937) is the same as the total number of parameters, indicating that all the parameters in this model are trainable. During training, the model adjusts these parameters to learn patterns from the data.

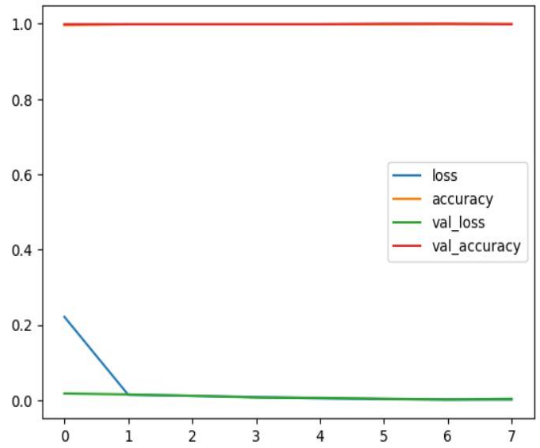
Non-trainable params: This value is 0, indicating there are no non-trainable parameters in the model. Non-trainable parameters are typically pre-trained weights that are not updated during training.

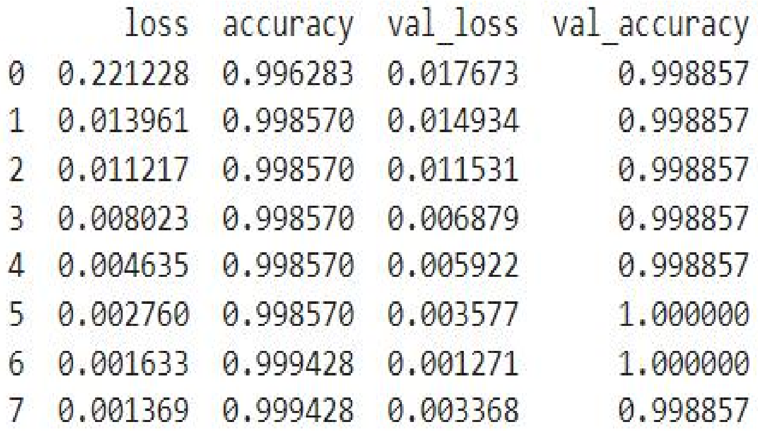


```
Epoch 1/25
110/110 [==============================] - 2s 5ms/step - loss: 0.2212 - accuracy: 0.9963 - val_loss: 0.0177 - val_accuracy: 0.9989
Epoch 2/25
110/110 [==============================] - 0s 4ms/step - loss: 0.0140 - accuracy: 0.9986 - val_loss: 0.0149 - val_accuracy: 0.9989
Epoch 3/25
110/110 [==============================] - 1s 5ms/step - loss: 0.0112 - accuracy: 0.9986 - val_loss: 0.0115 - val_accuracy: 0.9989
Epoch 4/25
110/110 [==============================] - 0s 4ms/step - loss: 0.0080 - accuracy: 0.9986 - val_loss: 0.0069 - val_accuracy: 0.9989
Epoch 5/25
110/110 [==============================] - 1s 6ms/step - loss: 0.0046 - accuracy: 0.9986 - val_loss: 0.0059 - val_accuracy: 0.9989
Epoch 6/25
110/110 [==============================] - 0s 4ms/step - loss: 0.0028 - accuracy: 0.9986 - val_loss: 0.0036 - val_accuracy: 1.0000
Epoch 7/25
110/110 [==============================] - 1s 5ms/step - loss: 0.0016 - accuracy: 0.9994 - val_loss: 0.0013 - val_accuracy: 1.0000
Epoch 8/25
110/110 [==============================] - 1s 5ms/step - loss: 0.0014 - accuracy: 0.9994 - val_loss: 0.0034 - val_accuracy: 0.9989
<keras.src.callbacks.History at 0x7ced8913abf0>
```

It shows the progress of the training process over multiple epochs.

1. **Epoch** : This refers to the number of times the entire training dataset is passed through the model. Each epoch represents one iteration of the training process.
2. **Loss** : This shows the training loss after each epoch. The loss is a metric that measures how well the model's predictions deviate from the ground truth. In this case, the loss seems to be steadily decreasing over epochs, indicating the model is learning.
3. **Accuracy** : This shows the training accuracy after each epoch. Accuracy is a measure of how often the model makes correct predictions. Here, the accuracy appears to be increasing over epochs, which is a positive sign.
4. **Val_loss** : This shows the validation loss after each epoch. Validation loss is measured using a separate dataset not used for training. It helps track how well the model generalizes to unseen data. Here, the validation loss seems to be following a similar trend to the training loss, which is a good sign.
5. **val_accuracy** : This shows the validation accuracy after each epoch. Ideally, the validation accuracy should track closely with the training accuracy to avoid overfitting. In this case, the validation accuracy seems to be increasing along with the training accuracy, suggesting the model is performing well on both the training and validation data.





```
       loss  accuracy  val_loss  val_accuracy
0  0.221228  0.996283  0.017673      0.998857
1  0.013961  0.998570  0.014934      0.998857
2  0.011217  0.998570  0.011531      0.998857
3  0.008023  0.998570  0.006879      0.998857
4  0.004635  0.998570  0.005922      0.998857
5  0.002760  0.998570  0.003577      1.000000
6  0.001633  0.999428  0.001271      1.000000
7  0.001369  0.999428  0.003368      0.998857
```

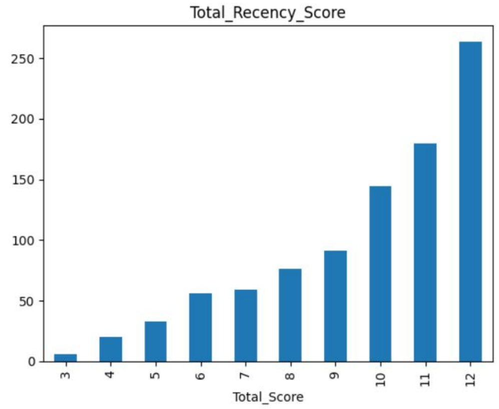
This is the training history of the churn prediction model where the different coloured lines represents different data:

Blue = loss (which is continuously decreasing)

Orange = accuracy (which is continuously increasing)

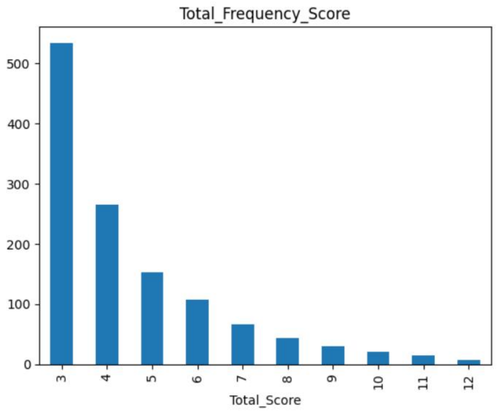
Green = val_loss (which is continuously decreasing)

Red = val_accuracy (which is continuously increasing)



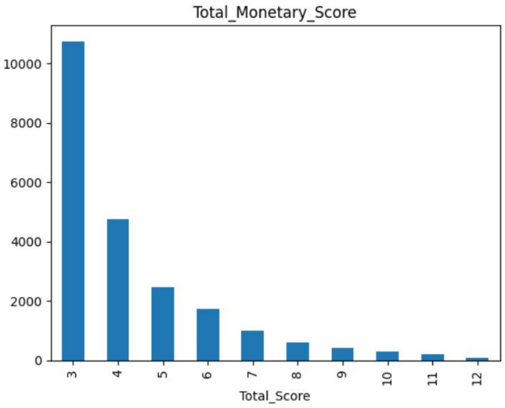
Here are some observations from the chart:

1. The most frequent scores are between 6 and 9.
2. There are very high scores (above 11) and very low scores (below 4).



Here are some observations from the chart:

1. The most frequent scores are between 5 and 8.
2. There are very high scores (3) and very low scores (after 11).

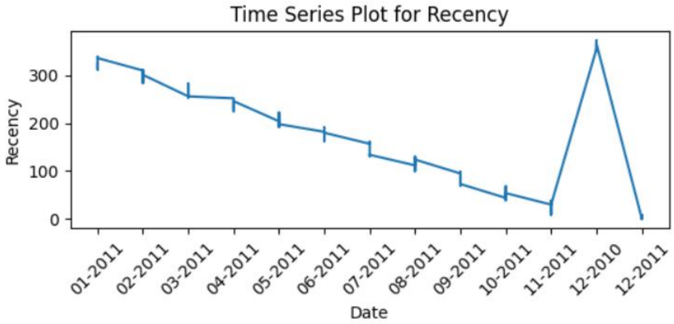


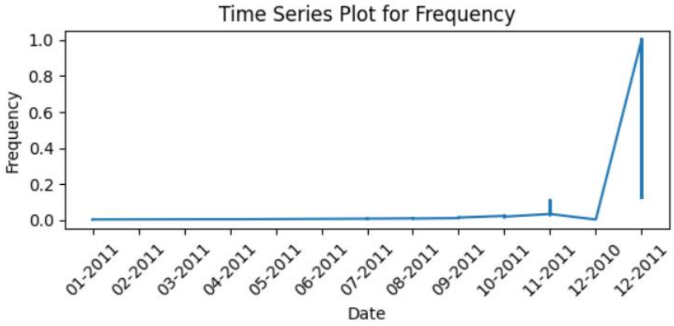
Here are some observations from the chart:

1. The most frequent scores are between 5 and 9.
2. There are very high scores (3) and very low scores (after 11).

Reasons for choosing Prophet model for time series analysis:

1. Ease of Use and Interpretability: Prophet is known for its user-friendly approach. It requires minimal data pre-processing and Hyperparameter tuning compared to other models. You can focus on interpreting the results rather than getting bogged down in complex configurations.
2. Automated Seasonality and Trend Handling: Prophet automatically detects and incorporates different time-based patterns into the model. This includes yearly seasonality, weekly seasonality (if applicable), and even the effects of holidays. We don't need to manually engineer these features, saving you time and effort.
3. Solid Performance for Many Time Series Problems: Prophet has proven effective in various time series forecasting tasks, particularly for data exhibiting trends and seasonal effects. It delivers good results for problems like predicting retail sales, website traffic, or resource utilization.
4. Robustness to Missing Data and Outliers: Real-world time series data often has missing values or outliers. Prophet is designed to handle these issues gracefully, providing more reliable forecasts compared to models that might be overly sensitive to such data imperfections.
5. Open Source and Fast Training: As an open-source library (available in Python and R), Prophet is freely accessible and well-documented. Additionally, it boasts fast training times, making it ideal for quick prototyping and iterative forecasting tasks.





The x-axis represents time, and the y-axis shows the recency score in 1st graph and frequency score in the 2nd graph. The blue line represents the forecasted trend, and the shaded area shows the uncertainty range around the forecast.

Here are some observations from the chart:

For recency:

- **Seasonality:** There appears to be a seasonal pattern in the recency score, with peaks repeating at regular intervals. This could be due to factors like holidays or promotional cycles that tend to occur at specific times of the year.

For Frequency:

- **Seasonality:** The frequency is first seems constant from 01-2011 to 08-2011 after that increasing at a slower rate, goes up to the peak then again comes down.

Insights Gained:

1. Several machine learning models can be used for the analysis.
2. Models can be optimized using the Hyperparameter tuning.
3. Evaluation metrics such as accuracy, precision, recall, f1 score were utilized to assess the performanc**e.**
4. The customers are mainly from the United Kingdom.
5. The purchases made by the customers of the United Kingdom are also maximum.

Recommendations for reducing the churning customers:

Focus on Understanding Why Customers Churn:

1. Customer Feedback: Conduct surveys, interviews, or analyze customer support interactions to understand reasons for churn.
2. Customer Segmentation: Analyze churn rates across different customer segments (demographics, purchase history) to identify high-risk groups.

Strategies to Address Churn:

1. Proactive Engagement: Offer onboarding programs, tutorials, or personalized content to help new customers get the most out of your product or service.

2. Improve Customer Experience: Ensure a smooth user experience, address customer pain points quickly, and provide excellent customer service.

3. Reward Loyalty: Implement loyalty programs, offer exclusive discounts, or provide rewards for continued engagement.

4. Win-Back Campaigns: Target at-risk customers with special offers or personalized incentives to entice them to stay.

5. Subscription Flexibility: Offer flexible subscription options (monthly, annual) or allow downgrades to retain price-sensitive customers.

6. Communication is Key: Keep customers informed about product updates, new features, and special offers. Personalize communication whenever possible.
