TopUp Mama

Data Scientist Assessment Report

BY SYLVIA ACHACH

DATA MERGING

The dataset was comprised of were 6 text files:

- Kenya Orders.csv
- Kenya Deliveries.csv
- Kenya Customers.csv
- Nigeria Orders.csv
- Nigeria Deliveries.csv
- Nigeria Customers.csv

For data merging I used a Python notebook. This process involved various steps:

1. Importing the data from their respective data sources.
2. Column correction:

   Because the data was in CSV, data especially in Kenya Deliveries had typos that include commas. This caused columns to be split e.g “6,30” which is obviously “6.30” was split into 6 and 30.

   I created a new csv file that corrected this and saved it as “Kenya Deliveries Cleaner.csv” in the data directory.

3. Concatenating data sources:

   The data can be divided into Kenya data sources and Nigeria data sources. They mostly have the same columns for the respective fields.

   In this step, I concatenated the data sources and reduced them to Orders, Deliveries and Customers. I added a new column country code to depict the country each row belongs to. This is important for money conversion

4. Identifying common columns:

   To ensure the data sources were joined, I needed to know the common columns so as to know where to perform the join on.

   For Customers and Orders data, I identified the “Customer ID” column was in both sources.

   For Orders and Deliveries data, I identified the “Order ID” column was in both sources.

   Deliveries and Customers data sources have no common columns.

5. Clean common columns:

   To ensure a correct join between the data sources, I had to clean the columns to ensure they have the same format. This was especially important for the “Order ID” column.

6. Merge on common columns:

   I used a pandas join function similar to an SQL join. I specifically used outer joins so that I can include all the data.

7. Save in a csv file:

   I saved it in “merge.csv”. It can be found the data directory

DATA CLEANING

This process involved making the data useful for model and analysis.

This process involved:

1. Replacing missing data symbols:

   The hyphen symbol “-” was used for missing data. This statistical and machine learning models cannot interpret this.I replaced the symbol with “NaN” from numpy

2. Identifying kinds of data. I categorized them into:
   - Monetary data
   - Other numerical data
   - Categorical data
   - Unique identifiers
   - Timestamps
3. Cleaning monetary data: This involved removing currency symbols.
4. Type casting

   This step involved correcting incorrect data types. This is important for appropriate interpretation of data by the models. The type casting involved converting:

   - Cleaned Monetary data to Float data type
   - Unique Identifiers to String data type
   - Timestamp data to datetime data type
5. Save the clean data to a file - “formatted_data.csv”. It can be found in the data directory.

PREDICTIVE ANALYSIS

This process involved building machine learning models to predict various factors. It also involves the steps taken to prepare data for model fitting and prediction.

PREPROCESSING

Before analysis, I proceeded to remove columns that were either all empty or had the same value for all columns.

CUSTOMER RETENTION ANALYSIS

In this analysis I am checking whether a customer is likely to order again. To make this analysis I try to understand the customer purchase history.

STEPS INVOLVED

Identify whether a customer that made at least one order in January is likely to make at least one other order in February. The steps involved:

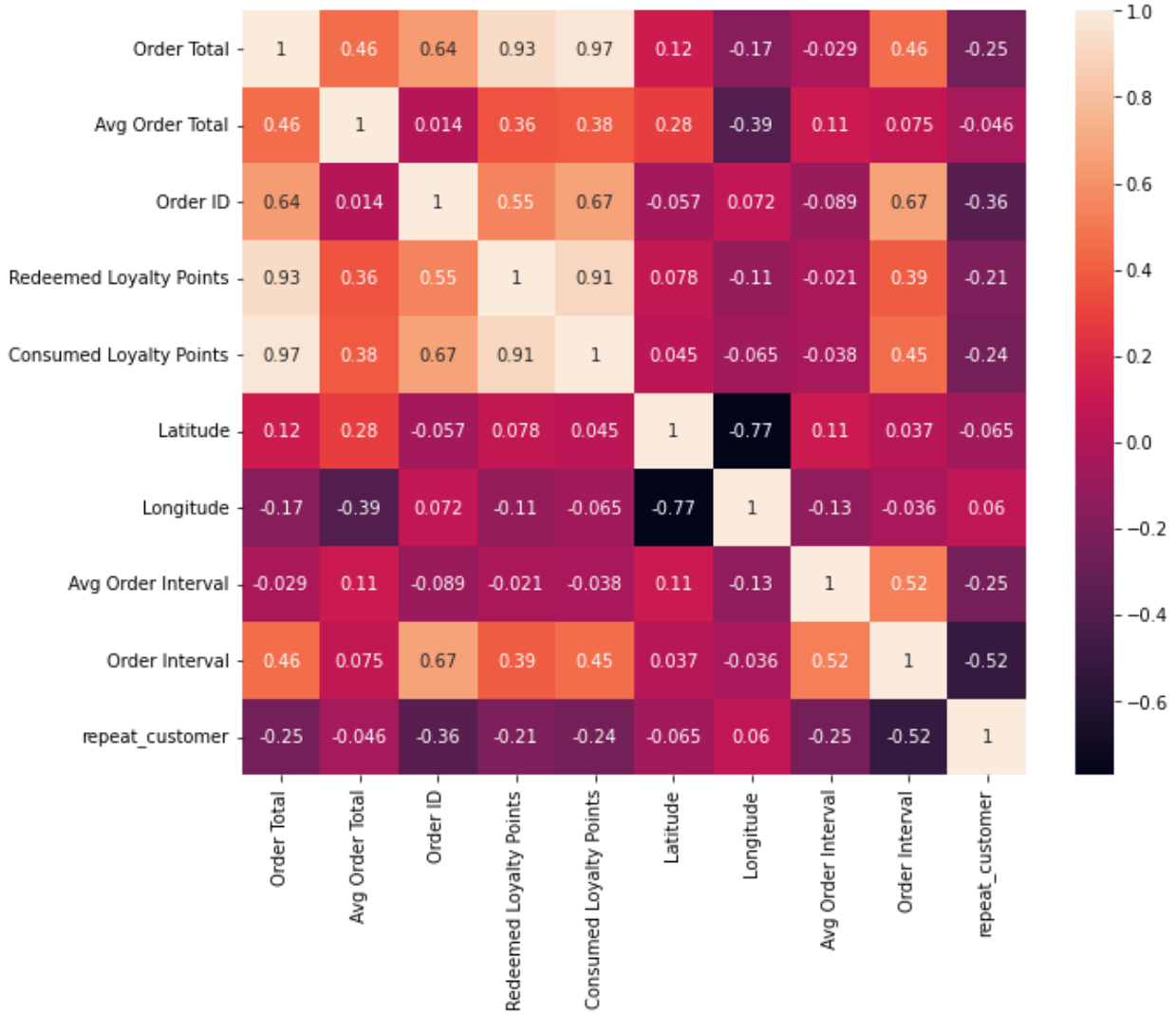
1. Aggregate columns and group by Customer ID
2. Split data into orders made in January and orders made in February
3. Generate features from January orders
4. Create a column of customers and whether they made an order in February or not.

DATASET DESCRIPTION

To predict whether a customer will make an order in February, I get the purchasing history of each customer. This involved:

1. Create *Order* column - this achieved by subtracting *Total Cost Price from Total*
2. Grouping data by *Customer ID* and:
    a. Sum the *Order Total, Consumed Loyalty Points, Redeemed Loyalty Points*
    b. Get min and max of *Order Time*
    c. Count *Order ID* per *Customer ID*
    d. Get mean of *latitude* and *longitude*
3. Create *Order Interval* which is *Order Time(max) - Order Time(min)*
4. Create *Average Order Interval* which is *Order Interval/Order ID(count)*

These features become the input for the model. The output of the column is whether the customer ordered in February or not based on their ordering habit in January.



PREDICTION

I used K-Fold validation with 5 folds. This dataset was passed into a Logistic Regression model to predict whether the customer ordered in February or not.

The accuracy of the model was 74% on average on test data.

CUSTOMER CLASSIFICATION ANALYSIS

In this analysis I break down the different segments of customers.

STEPS INVOLVED

To Identify different classes of customers. The steps involved:

1. Aggregate columns and group by Customer ID
2. Generate features from aggregated data

DATASET DESCRIPTION

1. Create *Order* column - this achieved by subtracting *Total Cost Price from Total*
2. Grouping data by *Customer ID* and:
    a. Sum the *Order Total, Consumed Loyalty Points, Redeemed Loyalty Points*
    b. Get min and max of *Order Time*
    c. Count *Order ID* per *Customer ID*
3. Create *Order Interval* which is *Order Time(max) - Order Time(min)*
4. Create *Average Order Interval* which is *Order Interval/Order ID(count)*

These features become the input for the model. The output of the model is the class the customer belongs to.

PREDICTION ANALYSIS

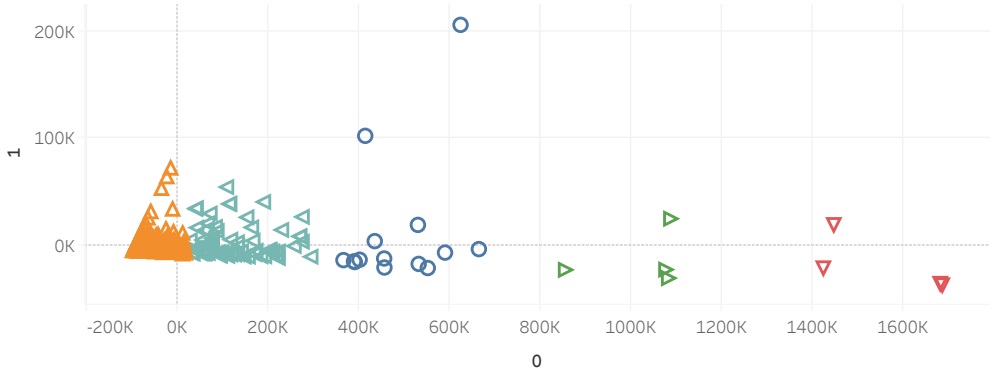
The generated dataset was passed into a K-Means model to predict customer classes. Using the elbow method I was able to identify the optimal number of classes to be 5.

I reduced the features for the model to be able to see the different classes.

Customer Classification Analysis

This is a visual analysis of customer classification using K-Means Classifier.

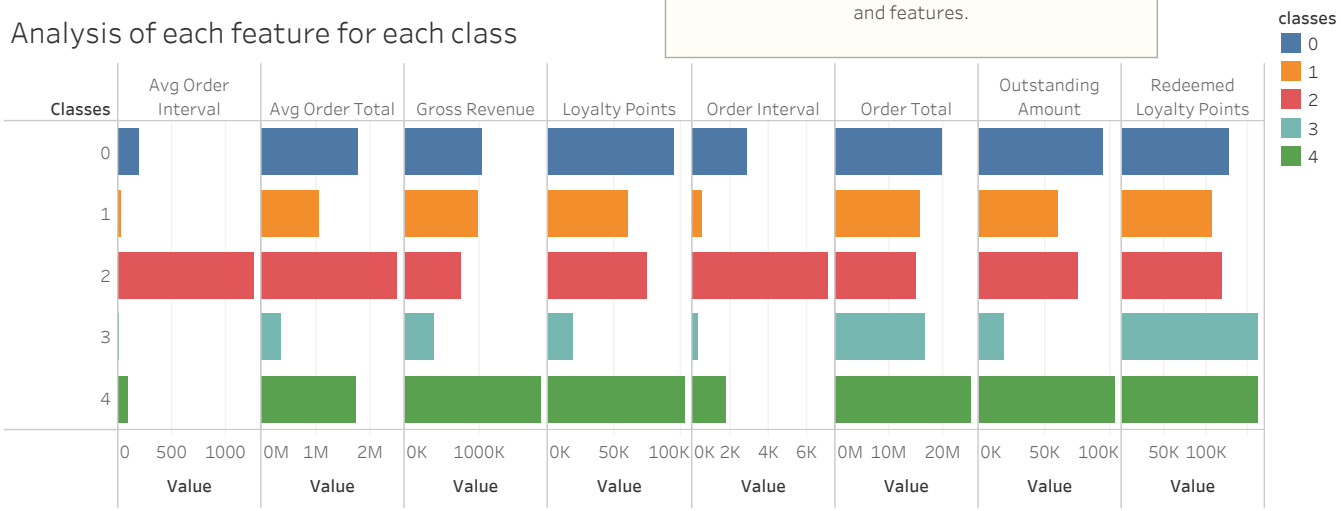
PCA Decomposition of Customer classification



Using PCA decomposition to demonstrate the different stratifications of customers based on the features used to predict the classes.

Analysis of each feature for each class

This chart shows the relationship of different classes and features.



CLASSES SEGMENTATION BREAKDOWN

Classes	Redeemed Loyalty Points	Revenue Generated	Re-order Interval	Order total	Loyalty Points
Zero	Slightly High	Moderate	Short	High	Slightly High
One	Slightly High	Moderate	Short	Moderate	Low
Two	Slightly High	Moderate	Long	Moderate	Slightly High
Three	High	Shortest	Extremely Short	Moderate	High
Four	High	Highest	Short	Extremely High	High

PRODUCT RECOMMENDATION

STEPS INVOLVED

1. Get the number of times a customer orders from a particular category of products.
2. Normalize the dataset so that the maximum number of orders per class is one.
3. Rescale by adding one so that now the minimum count is 1 and maximum is 2. This is because the Non-negative matrix factorization model does not accept zero.

DATASET DESCRIPTION:

The model accepts 3 features:

- The id of a user - *Customer ID*
- The iid - Category Name
- Rating - How many times the customer ordered a product

PREDICTION ANALYSIS

I split the data into a training and a testing set. The generated dataset was passed into an NMF(Non-negative matrix factorization) model.

RESULTS

Because there was no rating in the dataset, I interpreted the dataset in such a way that if the predicted value of the test set is greater than 1.5 then the customer is likely to purchase the product at least once otherwise no. This made it a classifier rather than a regression model.

The accuracy of the model was 87%

REVENUE OPTIMIZATION ANALYSIS

STEPS INVOLVED

To predict the revenue generated from data. I took the following steps to understand the customers purchasing history:

1. Aggregate columns and group by Customer ID
2. Generate features from aggregated data

DATASET DESCRIPTION

1. Create *Order* column - this achieved by subtracting *Total Cost Price from Total*
2. Grouping data by *Customer ID* and:
    a. Sum the *Order Total, Consumed Loyalty Points, Redeemed Loyalty Points*
    b. Get min and max of *Order Time*
    c. Count *Order ID* per *Customer ID*
    d. Get mean of *latitude* and *longitude*
3. Create *Order Interval* which is *Order Time(max) - Order Time(min)*
4. Create *Average Order Interval* which is *Order Interval/Order ID(count)*

These features become the input for the model. The output of the model is the revenue generated.

PREDICTION AND RESULTS

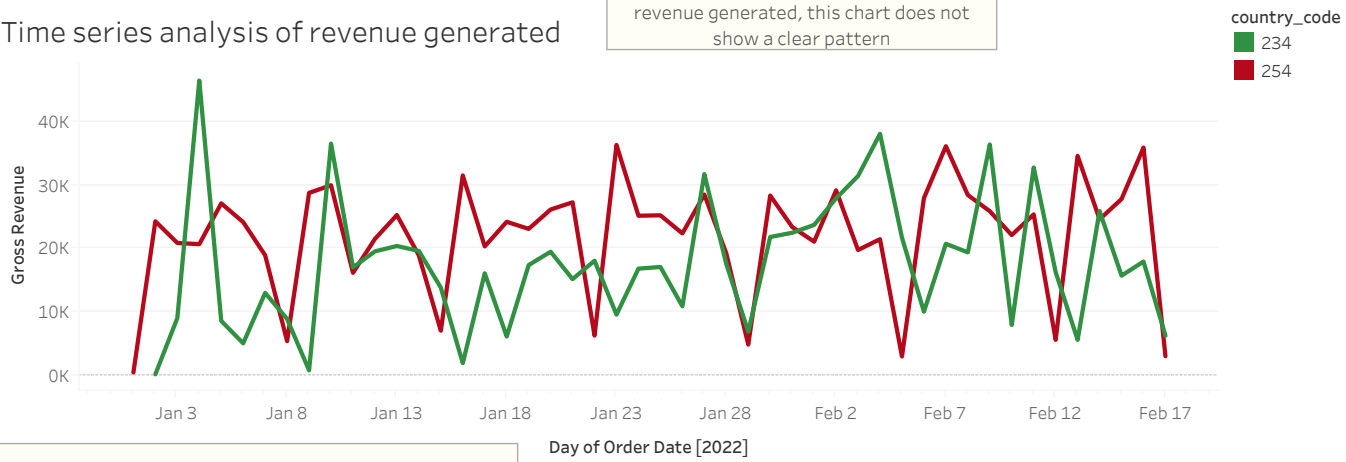
I used K-Fold validation with 5 folds. I passed the data into a Linear Regression model. The average r2 score of the model is -0.35. Analysis of the dataset is on the next page.

Revenue Optimization Time Series Analysis

In this analysis I used time components and compare with revenue generated.

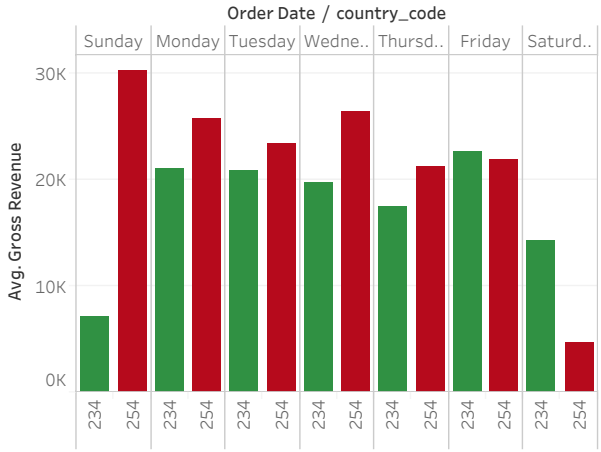
Time series analysis of revenue generated

This is a time series check for patterns in revenue generated, this chart does not show a clear pattern



This chart checks for pattern in revenue generation based on day of week.
It is worth noting that revenue is generated the most on Sunday in Kenya but least in Kenya. This implies that more resources should be redirected to Kenya on Sunday.

Revenue Generated for each day of week



This chart checks for pattern in revenue generation based on day of month. There is generally no clear pattern. In Nigeria revenue generation is more erratic.

Revenue generated for each day of month