**Project Proposal**

**Customer Segmentation and Products Recommendation Engine on Brazilian Ecommerce DataSet**

**By**

**Srikaran Elakurthy**

**Avinash Yavvari**

**Course Project Professor’s Name**

**Dr. Duo Liao**

**Course Name**

# Big Data Essentials

**Course Number and Section**

**AIT614**

**Date**

**4/15/20**

A close up of a sign

Description automatically generated

**Customer Segmentation and Products Recommendation Engine on Brazilian Ecommerce DataSet**

**Introduction:**

This Project focuses on the analyzing the heterogenous features of Brazilian ecommerce dataset. Dataset consists of 9 files each file describes the details of orders, customers, Payments, Products and reviews respectively. Dataset has around 100k orders from Brazilian marketplace at multiple locations in Brazil. This Dataset was provided by Olist, large departmental store in Brazil.

Project is divided into 2 parts, In first part of project it mainly focuses on customer segmentation and finding customer patterns by analyzing the behavioral and psychographics data of a customer, which helps the organization to analyze, on which products the customers are mostly likely to buy it and for which category of product they need to increase their quantity to maximize their profits. It is basically we segment the customers into different cluster groups and apply apriori algorithm to find the association rules and patterns.

In the second part of project mainly focuses on the recommending the product to a customer by analyzing the customer interests and reviews and even taking the consideration of delivery performance and finally analyzing[[1]](#footnote-1)the quality of products.

None of the papers considered applying association rule algorithm APRIORI upon the clusters and use those rules for recommendation engine. Most of the projects on recommendation doesn’t consider the delivery performance, geological distance and analyzation of reviews with in it. In this project we try embed all these concepts and try to make a recommender system which not only analyze the results but also make an analysis from the clusters obtained and rules and then makes a recommendation.

**Objectives:**

* Forming a rich dataset by Integrating the data from different files which can be used in modeling the customer segmentation.
* Performing customer segmentation to create a cluster of customers specifying what are there interests on products for each group.
* Analyzing which products are more interested and prone to get selled.
* Analyzing the patterns of the clusters and find the valuable insights which would be helpful in increase of business market.
* Recommendation of products per customer by considering reviews, geological distance and previous purchases and using the previous clustering information.

**Proposed Selected Dataset:**

Dataset consists of 9 files and each file and its features are described as follows:

**Customers Dataset:** This dataset has information about the customer and its location. Use it to identify unique customers in the orders dataset and to find the orders delivery location.

**Features:**

1. customer\_id : key to the orders dataset. Each order has a unique customer\_id.
2. customer\_unique\_id: unique identifier of a customer.
3. customer\_zip\_code\_prefix : first five digits of customer zip code
4. customer\_city: city that customer stays
5. customer\_state: state that customer stays

# Geolocation Dataset: Dataset consists of zipcodes with their latitude and longitude and which state and city.

# Features:

1. geolocation\_zip\_code\_prefix: first 5 digits of zip code
2. geolocation\_lat: latitude
3. geolocation\_lng: longitude
4. geolocation\_city: city name
5. geolocation\_state: state

# Order Items Dataset: Data consists of details for each order.

# Features:

1. order\_id:order unique identifier
2. order\_item\_id:sequential number identifying number of items included in the same order.
3. product\_id:product unique identifier
4. seller\_id:seller unique identifier
5. shipping\_limit\_date:Shows the seller shipping limit date for handling the order over to the logistic partner.
6. Price:item price
7. freight\_value:item freight value item (if an order has more than one item the freight value is splitted between items)

# Payments Dataset: Data consists about the payment options.

# Features:

1. order\_id: unique identifier of an order.
2. payment\_sequentiala : customer may pay an order with more than one payment method. If he does so, a sequence will be created to accommodate all payments.
3. payment\_type: method of payment chosen by the customer.
4. payment\_installments: number of installments chosen by the customer.
5. payment\_value: transaction value.

# Order Reviews Dataset: Data consists of the reviews for each order made by the customer.

# Features:

1. review\_id: unique review identifier
2. order\_id: unique order identifier
3. review\_score: Note ranging from 1 to 5 given by the customer on a satisfaction survey.
4. review\_comment\_title : Comment title from the review left by the customer, in Portuguese.
5. review\_comment\_message: Comment message from the review left by the customer, in Portuguese.
6. review\_creation\_date: Shows the date in which the satisfaction survey was sent to the customer.
7. review\_answer\_timestamp: Shows satisfaction survey answer timestamp.

# Order Dataset : This data set is the main dataset it contains all the information of dataset.

# Features:

1. order\_id: unique identifier of the order.
2. customer\_id: key to the customer dataset. Each order has a unique customer\_id.
3. order\_status: Reference to the order status (delivered, shipped, etc).
4. order\_purchase\_timestamp: Shows the purchase timestamp.
5. order\_approved\_at: Shows the payment approval timestamp.
6. order\_delivered\_carrier\_date: Shows the order posting timestamp. When it was handled to the logistic partner.
7. order\_delivered\_customer\_date: Shows the actual order delivery date to the customer.
8. order\_estimated\_delivery\_date: Shows the estimated delivery date that was informed to customer at the purchase moment.

# Products Dataset: Dataset consists about the details about the product used by OList.

# Features:

1. product\_id: unique product identifier
2. product\_category\_name: root category of product, in Portuguese.
3. product\_name\_lenght: number of characters extracted from the product name.
4. product\_description\_lenght: number of characters extracted from the product description.
5. product\_photos\_qty: number of product published photos
6. product\_weight\_g: product weight measured in grams.
7. product\_length\_cm: product length measured in centimeters.
8. product\_height\_cm: product height measured in centimeters.
9. product\_width\_cm: product width measured in centimeters.

# Sellers Dataset:This dataset includes data about the sellers that fulfilled orders made at Olist. Use it to find the seller location and to identify which seller fulfilled each product.

# Features:

1. seller\_id: seller unique identifier
2. seller\_zip\_code\_prefix: first 5 digits of seller zip code
3. seller\_city: seller city name
4. seller\_state: seller state

# Category Name Translation: Translates the category name to English.

# Features:

1. product\_category\_name: category name in Portuguese
2. product\_category\_name\_english: category name in English.

**Selected Features:**

# For recommendation engine we will use all the features except the state and city feature and for predicting the sales we won’t be using the unique id ‘s and some of the features we will be deriving such as distance from the 2 zip codes latitudes and longitudes , number of days consumed for shipment etc. For clustering technique we would be not using reviews dataset except the rating score and also not use the id’s in different data sets.

# These features are not final selected features it may change after my 1st training. Then again we apply some feature importance algorithms or feature selection algorithms to obtain selected features for our model, considering even the correlations among them.

# Description of Proposed System:

# First we load all our files in AWS S3 module and we will store and access them using HIVE in EMR. By using HIVE will be accessing and querying the Datasets and Integrating the datasets to make a rich dataset which will be used for our models. Once the Final Dataset is ready we will access this data set by using Pyspark and perform our Data Analytics where we will be doing the Customer Segmentation and Recommendation Engine in it. Once it is done we will store the results in AWS S3 and go for visualizing the results obtained and find the insights from it. Each of our step will be clearly briefed in below sections.

# A screenshot of a cell phone Description automatically generated

# Methods to be proposed are:

# Initially we load the all 9 data files into AWS S3 Data lake. Then create EMR clusters in AWS . In the EMR clusters we will access hive and do the Data preprocessing on all 9 datafiles such as find the missing values and impute them with the mean if it is a numerical variable or mode with a categorical variable and removing the duplicate rows. Then we formulate a rich dataset by aggregating all the 9 files.

# Preparation of rich dataset by using HIVE:

# A picture containing text Description automatically generated

# New Features Derived:

# *Is\_latedelivered*: if it is a late delivery will have value 0 and if it’s not value will be 1. This feature will be derived by subtracting the dates of estimated delivery date and delivered date by using order dataset.

# *No\_of latedeliverydays*: If it’s not a late delivery the number of days the ordered delivered late is 0 otherwise it will be the negative . This feature will be derived by subtracting the dates of estimated delivery date and delivered date by using order dataset.

# *TFidf frequency for review text:* By using review text we will create a tfidf frequency for the words in our review corpus.

# *Distance:* calculate the distance between customer and seller. This will be calculated by matching the customer id in the orders and customer dataset and then link the order id with order item dataset and then linking with seller id of order item dataset and sellers database. Extracting customer zip codes and seller zipcodes for each order and then we use geological dataset to extract longitude and latitude and calculate distance.

# *Polarity\_of\_review\_text*: By analyzing the review description and review title and will be making an sentiment analysis from it whether it is a positive polarity or negative polarity.

# *No\_of \_Positive\_words:* Will calculate the no of positive words in each review.

# *No\_of \_negative\_words:* Will calculate the no of negative words in each review

# After formulating final dataset table we will again store it into Amazon S3 data lake for future access.

# Descriptive Analytics:

# What is the amount spent on each product category purchased by all customers?

# Which state of sellers has maximum profit?

# From which state the maximum number of customers are buying?

# How many people are using installment payment method?

# How many product are there in each review score?

# Inferential Statistics:

# Does the customers are okay with the late deliverys?

# Products having more than 1 photos has more sales than products having only 1 picture.

# Products having good review score always delivered on time?

# Sellers near to customers does they deliver always on time?

# Advanced Data Analytics:

# Customer Segmentation Model:

# For customer segmentation model we will create a data-table(Customer\_product) using final dataset in S3 and Hive in EMR cluster where each row say about a customer how much he is spending on each product category. This table is again gets stored in AWS S3 data lake. By using AWS EMR cluster we will access Spark and using machine learning library. In this EMR we access the S3 and extract the Customer\_product table and perform feature engineering (Standardization and PCA) and then we apply K-Means clustering technique with using Silhouette and SSE we will find out the right number of clusters to decide. As sometime the SSE value may not indicate a clear “Elbow” to choose. So we decide the right number of clusters by having less SSE using elbow and high silhouette. Then we visualize these clusters in Radial chart to find out each cluster is interested on what products. We append the cluster results and append each row of Customer\_Product table with cluster number and interested products. Then store again to AWS S3 data lake.

# After this we will create another cluster model which will help us to find some patterns. Here we will access AWS S3 data lake from AWS EMR Spark cluster and use the final data set and change all product category section to one hot encoding variables where each category becomes a column and if that order has that product category it has value 1 or other wise 0. Then we will we perform the same action on payment type and state and order status and remove the review creation and review answered timestamp columns from the dataset. Then we will take review text we will calculate the polarity, No of positive words and no of negative words of each review text by using NLTK and text blob in Spark. Then we perform feature Engineering that is Standardization and PCA and perform K-Means clustering with number of clusters are decide by Silhouette and SSE. Once the clusters are decided we try to visualize the clusters and find the patterns among it.

# Then we perform the Apriori algorithm to see the rule mining associations and rule patterns which we will use in our recommendation model. Then we will store the cluster results and rules in AWS S3 datalake. By using the rules we perform visualizations to bring out the insights from it.

# Product Recommendation Engine:

# In the AWS EMR spark cluster, access the final dataset from AWS S3 data lake and calculate polarity, no of positive words and no of negative words and TFIDF frequency matrix with different ngram range(1to 3) from reviews text and horizontal stack all these to the final dataset and perform one hot encoding on categorical variables. After that we pass the obtained data into hybrid recommendation system and content based filtering and check the results and finalize the model which do’s the better. Once we get the recommended products we will check that customer falls in which cluster list in Customer\_Product table and check whether the recommended products lies in the interest list and then we match with the patterns or rules obtained in 2nd clustering model to find the sellers and by the pattern and rule we can find out whether the customer is willing ok for late delays and even long shipment time or not. According to above analysis we will be recommending the products.

# Data Visualizations:

# Preparing a radial chart to represent the relations between clusters from 1st method and product categories.

# Visualizing the clusters and finding the patterns after the 2nd clustering method.

# Representing a Geo map of USA and using fill as the total value of orders till now processed from each state.

# Representing bar plot of how many of each product category has sold.

# Plotting an Density plot to see how is the distribution of product categories in dataset.

# Proposed Data Platforms:

# *Frameworks*: AWS S3, AWS Spark

# *Languages*: Python and R

# *Visualization*: Tableau and AWS S3 visualization tool

# Appendix:

# Customers data set:

# A close up of text on a white background Description automatically generated

# Geolocation Dataset:

# A close up of text on a white background Description automatically generated

# Order item data set:

# A screenshot of a cell phone Description automatically generated

# Products Data set:

# A screenshot of a cell phone Description automatically generated

# Source:

# Source of the data set is [Brazilian Ecommerce data](https://www.kaggle.com/olistbr/brazilian-ecommerce)

1. (au, n.d.) [↑](#footnote-ref-1)