# Project Title

Hyperlink to pre-recorded presentation video

## **Introductory Problem statement**

Research object:

The research object of this project is to analyze the operational data of Olist Store, the largest department store in the Brazilian market. This dataset has data of almost 100,0000 orders from 2016 to 2018 made at multiple marketplaces in Brazil and its features include order status, price, payment, freight information, product attributes, and ratings & reviews written by customers. This project investigates the factors affecting customer shopping experience ratings of customers' shopping experience using machine learning methods including Logistic Regression, K-Means Clustering, and Word embedding.

Background and Significance of Project:

When a customer purchases a product online, the seller is notified, and the product will be shipped. When the buyer receives products and the order is completed, the customer will receive a survey via email, rate the shopping experience and write a review. Reviews and ratings are important to the decision-making process for both customers and sellers. It is helpful for customers to get a better idea about the product and shopping experience from other buyers, including product quality, and timeliness of delivery. For sellers, they can obtain relevant feedback through the buyer's review and rating of the order, including information on product quality and timeliness of the whole delivery period. This information could help them improve their products and operations to attract more customers and achieve a larger market share.

## **1 Exploratory Data Analysis**

Website address:

<https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=olist_customers_dataset.csv>

Description of dataset:

These csv files are all from the same Brazilian ecommerce public dataset, which allows viewing the same order from multiple dimensions, including customers, sellers, products, orders etc. This dataset is that all data in this dataset is real commercial data although it is all anonymized.

## Data Schema:

The data schema graph below indicates the way that different datasets connect to each other. In short, ID of order, product, customer, and zip code is used to provide interconnection.

Customer (The number of columns: 3; The number of rows:99441)

|  |  |
| --- | --- |
| Column name | Description |
| Customer\_id | key to the orders dataset. Each order has a unique customer\_id. |
| Custormer\_zip\_code\_prefix | first five digits of customer zip code. |

Sellers (The number of columns: 2; The number of rows:3095)

|  |  |
| --- | --- |
| Column name | Description |
| Seller\_id | seller unique identifier. |
| seller\_zip\_code\_prefix | first five digits of seller zip code. |

Items (The number of columns: 4; The number of rows:988666)

|  |  |
| --- | --- |
| Column name | Description |
| order\_id | order a unique identifier. |
| Order\_delivered\_carrier | When it was handed to the logistic partner. |
| Order\_delivered\_customer | Shows the actual order delivery date to the customer. |
| Order\_estimated\_delivery | Shows the estimated delivery date that was informed to the customer at the purchase moment. |

Orders (The number of columns: 3; The number of rows:99441)

|  |  |
| --- | --- |
| Column name | Description |
| order\_item\_id | sequential number identifying number of items included in the same order. |
| Shipping\_limit\_date | Shows the seller shipping limit date for handling the order over to the logistic partner. |
| price | Item price. |

Products (The number of columns: 2; The number of rows:32951)

|  |  |
| --- | --- |
| Column name | Description |
| Product\_id | product unique identifier. |
| Product\_category\_name | root category of product, in Portuguese. |

Reviews:

The number of columns: 2; The number of rows:98410

|  |  |
| --- | --- |
| Column name | Description |
| review\_id | unique review identifier |
| Review\_score | Note ranging from 1 to 5 given by the customer on a satisfaction survey. |

## **2 Customer Rating Analysis**

### **2.1 Data Preparation / Cleaning**

1. Create all data frames from **all CSV files in the data description part** and drop empty rows.

2. Change the types of some columns to appropriate types. For example, change the ‘review\_score’ column from string type to integer type and all columns related to time to timestamp type.

3. Create new columns to represent different time intervals. In this step, four new columns were created. ‘Actual\_delivery\_period’ represents the actual time for logistics to deliver the courier. ‘Estimated\_delivery\_period’ indicates the estimated time for logistics to deliver. ‘seller\_package\_period’ shows the time interval sellers spend on the packaging. And the last one ‘max\_packaging\_period’ explains that according to ‘shipping\_limit\_date’, the longest time for sellers to package. **‘Review\_score’ was predicted by setting 1-3 as 0, and 4-5 as 1 which means a positive review.**

4.. The last step is to create two new columns to represent whether sellers and logistics deliver the goods on time. If their actions are timely, then the ‘package\_timely’ and ‘delivery\_timely’ columns will be 1.

### **2.2 Machine Learning models**

1. For categorical columns, there are seven columns such as 'product\_category\_name\_english', 'customer\_zip\_code\_prefix', 'actual\_delivery\_period', 'order\_item\_id', 'package\_timely 'and 'delivery\_timely' and ‘price’. Where the first two columns are string classes, we must transform their classes.

2. Using **StringIndexer and OneHoter** functions to transform these columns, then save new columns into a new data frame.

3. Using the **VectorAssembler** function to assemble all seven columns and save a new column named ‘features’ into a new data frame called ‘features\_df’. There are 10081 rows of data in this data frame and accordingly, the **baseline** of this data frame is **69.51%.**

4. Then split the features\_df into train\_df and test\_df randomly, where train\_df accounts for about 70%.

5. Using the **LogisticRegression** function to regress the train set, where the label column is ‘review\_score’.

### **2.3 Model Validation**

1. Using the **LogisticRegression** function to regress the test set, where the label column is ‘review\_score’.

2. Using **accuracy, precision, recall and ROC score functions** to get the results of evaluation of this model.

The results are as follows:

|  |  |
| --- | --- |
| Evaluation Methods | Results |
| Accuracy | 83.9592% |
| Precision | 87.5116% |
| Recall | 93.5116% |
| ROC Score | 90.7261% |

3. Figure out the best model. Using **ParamGridBuilder** and **CrossValidator** functions to find the **best model** and get the new ROC score. The result of **ROC score** raises up to **91.6063%**.

### **2.4 Discussion of Result**

After data cleaning, around 10,000 rows are used for getting the model and predictions. The ROC score and other estimation methods show a good result for this model. And the best model after adjustment raised the ROC score up to 91.6063%.

## **3 Customer Review Clustering**

In this part, we want to use NLP techniques to analyze customer review texts. Firstly, by exploratory data analysis, we should know whether customers are satisfied and what the possible reasons are. Secondly, we want to cluster reviews into several groups to further understand the reasons why customers are satisfied or upset.

We choose unsupervised methods for several reasons: Firstly, we did not use ratings as labels to develop a supervised model because ratings already reveal customers’ sentiment, and ratings are also more abundant than comments. There is no need to predict ratings using comments. Secondly, to classify texts into more than 2 groups that represent different meanings, labels are needed in supervised learning. However, labeling is time-consuming. For the reasons above, we plan to practive unsupervised methods for text understanding.

### **3.1 Exploratory Data Analysis**

#### 3.1.1 Data Cleaning & Preprocessing

The review dataset contains around 10,000 non-null comment texts which still contain irrelevant data due to web crawler errors. The column ‘review\_score’ has several rows of text messages, therefore we also removed irrelevant rows. The column ‘review\_comment\_message’ is pre-processed according to following procedures: (1) normalizing and converting words to lowercase; (2) removing stopwords (in Portuguese); (3) Word stemming. After cleaning and pre-processing, we tried two different inputs of features, which are n-gram TF-IDF and Word2Vec.

#### 3.1.2 N-Gram Exploratory Analysis

In this part of the analysis, we try to understand that customers give top-rated reviews (which is of rating =5) and the least-rated review (which is of rating = 1) by forming different N-Gram words. We formed unigrams, bigrams, and trigrams words and tried to analyze the comments.

From the table below, we can conclude that most of the high-score reviews are due to good delivery service and excellent product quality. On the other hand, low-score reviews are all caused by the bad delivery service.

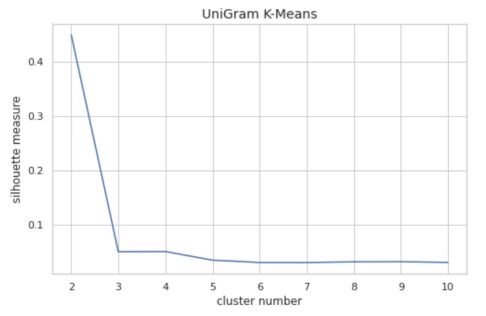
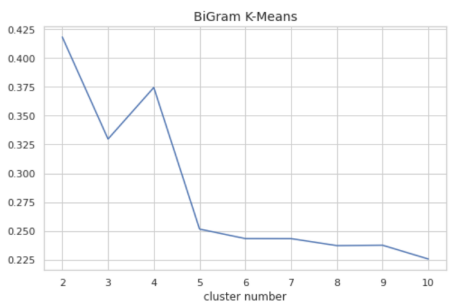
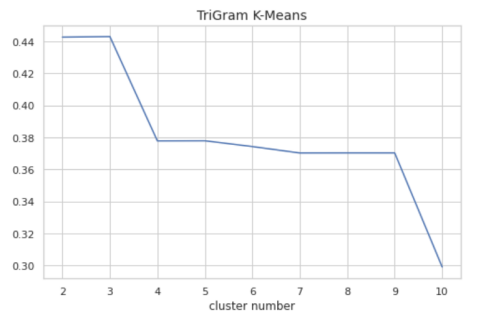
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **High Score Reviews (TOP 3 single and 3-word phrase)** | | | | | |
| **Words in Portuguese** | **Words in English** | **Count** | **Words in Portuguese** | **Words in English** | **Count** |
| chegou antes prazo | arrived before deadline | 903 | produto | product | 7890 |
| bem antes prazo | well before deadline | 569 | recomendo | recommend | 5032 |
| entregue antes prazo | delivered before deadline | 509 | qualidade | quality | 4849 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Low Score Reviews (TOP 3 single and 3-word phrase)** | | | | | |
| **Words in Portuguese** | **Words in English** | **Count** | **Words in Portuguese** | **Words in English** | **Count** |
| nao recebi produto | did not receive the product | 840 | nao | no | 6687 |
| ainda nao recebi | still have not received | 361 | produto | product | 4889 |
| produto nao entregue | product not delivered | 244 | recebi | receive | 2686 |

### **3.2 N-Gram K-Means Clustering**

N-grams are continuous sequences of words, symbols, and tokens in a document. They can be defined as the adjacent sequences of items in a document in technical terms. Compared with Bag of Words which only uses single words, it might give more information about the text meanings.IDF is used to surface key information of each comment. We will apply TF-IDF to uni-gram, bi-gram, tri-gram raw features respectively to compare the results.

#### 3.2.1 Number of Clusters

For the unigram features, the highest silhouette is 0.45 when k =2, after which the silhouette drops swiftly to around 0. Therefore, we choose k=2 as cluster number for unigram feature K-means clustering. K=2 is also the best cluster number for both bigram and trigram models.

The results show that the bigram and trigram features did not improve the highest silhouette score much, but the difference is that when dividing into more clusters, the silhouette score drops less sharply. When dividing into 10 clusters, the trigram model still has a silhouette score of around 0.30.

#### 3.2.2 Model Results & Validations

Since bi-gram and trigram do not improve the model much, we still use the uni-gram model to examine the results. To better explain the predicted clusters, it is supposed that the comments are clustered into positive and negative types. We also assume that the attitudes in review scores are consistent with review texts. Therefore, using review scores as true labels, we can examine the clustering results in the binary classifications’ way. The confusion matrix is shown below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | True Label | |  | Metrics | Score |
| Positive | Negative |  | Precision | 79.29% |
| Clustered  Label | Positive | 5522 | 1142 |  | Accuracy | 75.32% |
| Negative | 748 | 1163 | Recall | 88.08% |
|  | Specificity | 44.64% |

The data is unbalanced, with around 73% of positive comments. The accuracy score 75.32% mildly exceeds the benchmark. The model is good at recalling positive comments. However, it cannot detect negative reviews well. The model does not significantly cluster comments into positive and negative groups.

The most frequent 10 words of positive types are ‘producto’,’prazo’,’ant’,’entrega’,’chegou’,’bom’,’recommendo’,

’bem’,’qualidad’,’tudo’,’excelent’,otimo’,e’. Most frequent 10 words of negative types are ‘producto’,’e’,’vio’,

’compre’,’entrega’,’prazo’,’chegou’,’compra’,’dia’,’bem’,’ant’,’loja’,’entregu’,’ainda’. The words featuring positive sentiments are ‘recommendo’ (recommend), ‘bom’/’bem’ (well)， ‘qualidad’ (quality)，’excelent’ (excellent), ‘ótimo’ (great). The words of clustered negative types do not show equally strong sentiment directions. However, both types of comments are highly related to the delivery problems. The words ‘entrega’/’prazo’ (deadline), ‘chegou’ (arrived), ‘recebi’ (received) appear frequently in both clusters.

#### 3.2.3 Model Robustness

K-means clustering results can be unstable due to random selection of initial centers. Therefore, we further examined whether our results are normal. We run the model using 10 different random seeds. The silhouette scores for the 10 models are [0.44, 0.75, 0.45,0.45, 0.44, 0.45, 0.45, 0.45, 0.45, 0.45]. The mean of 10 silhouettes is 0.48 and variance is 0.83 indicating that the results are unstable. However, our model (silhouette=0.45) is among the normal ones.

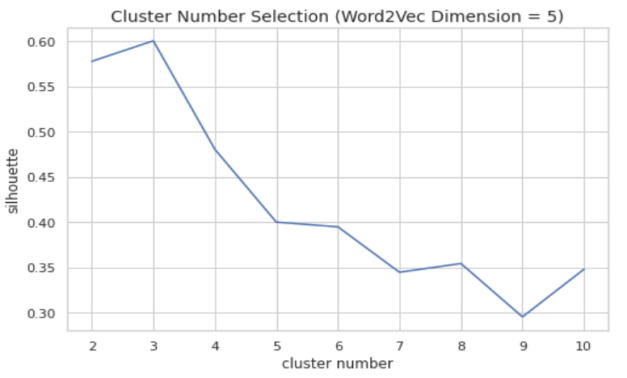
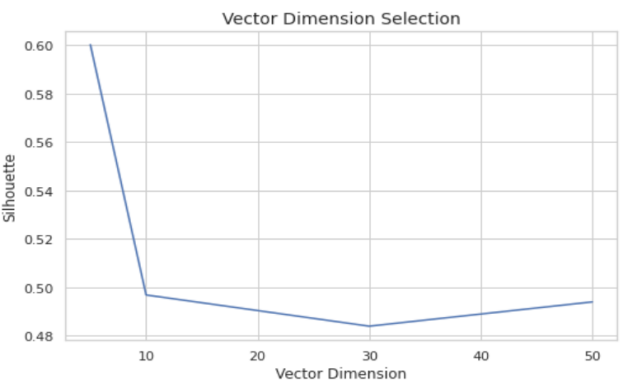
### 3.3 Word2Vec K-Means Clustering

The K-means model mildly succeeded in dividing comments into positive and negative ones. Though to further explore the reasons why customers are satisfied or not, we want to cluster the comments into more groups beyond positive and negative. At first, we tried LDA topic modelling, however, the results are hardly explainable. Alternatively, we intend to improve the k-means clustering results by changing the feature inputs.

Word2Vec method is practiced as a comparison to N-gram TF-IDF features. rd2Vec represents the texts by several dimensions of representations. It can reduce dimensions which serve to filter out noises and is expected to improve the model's performance. In this part, we use word embeddings as input features to test whether the prediction performance can be improved or that texts can be classified into more clusters.

#### 3.3.1 Model Selections

Firstly, we test the silhouette measure when cluster number is set to 3 under different dimensions of Word2Vec input. When dimension =5, the final silhouette of K-means clustering is significantly higher.



Using 5-dimension Word2Vec, the k=3 model silhouette is slightly higher than n-gram k-means, and the k=3 model silhouette wins the highest. The selected best model is when Word2Vec dimension = 5 and cluster number =3.

#### 3.3.2 Model Results & Explanations

After word embedding, the k-means model now can cluster comments into three groups, with a significantly higher silhouette of 0.60. The clustered results are shown in the Table for explanations.

Cluster 0 and Cluster 2 are positive comments. The differences are that Cluster 0 is about delivery and Cluster 2 is more related to sellers and more extremely positive words are used. However, Cluster 2 might not be accurate since it contains a lot of positive words mixed with negative words

|  |  |  |
| --- | --- | --- |
| **Cluster 0** | **Cluster 1** | **Cluster 2** |
|  |  |  |
| ‘product’, ‘deadline’, ‘well’, ‘delivery’, ‘recommend’, ‘before', 'fast’, ‘packed’ | ‘product’,’received’,’came’,’delivery’,’day’,’bought’,’before’,’arrived’,’store’,’and’,’request’,’replacement’ | ’well’,’fast’,’quality’,’great’, ‘liked it’,’seller’,’everything’,’attendance’,’price’,’expectation’ |
| 3507 comments in total  score 4-5: 94.67%  score 1-3: 5.33% | 4493 comments in total  score 4-5: 47.2%  score 1-3: 52.79% | 875 comments in total  score 4-5: 94.74%  score 1-3: 5.26% |

### 3.4 Discussion of Results

Our initial intention was to cluster texts by meanings (e.g. delivery/quality/service..etc..), but the results are unsatisfactory, the models failed to cluster comments beyond positive or negative. This might be due to limited data (LDA needs a larger scale of data to train) and the simplicity of the model. K-means can only capture linear relations and the contexts are also neglected by either TF-IDF or word2vec methods which only focus on words or phrases themselves. Secondly, the model cannot de negative sentiments well, this might be due to unbalanced data. There is an inadequate number of negative comments. To reach the initial goal of clustering texts by meanings, more complex models or larger data is needed.

However, our model succeeds in clustering comments into positive and negative sentiments, and the Word2Vec K-means model further divides positive comments into two groups, which potentially represent delivery and product/service, respectively.

## **4 Conclusion**

From part two of this article, we can see that the product category and timeliness of packaging and delivery are key factors that affect customer’s rating. Using these factors to run the Logistic Regression function, we could make predictions much better than the baseline level. Then, in part three, from both n-gram exploratory analysis and k-means clustering, we noticed that most of customers’ confirmations and complaints are also related to delivery.

Therefore, combined with the research themes of this article, it is suggested that If sellers want to make their products more attractive, they should try their best to reduce the packaging period and choose better courier services to ensure that products can reach consumers in time.