Capstone Project 2: Final Report

Customer review rating prediction

By Simantini Patil

Introduction

Many companies today measure customer satisfaction. Some of the quick ways of collecting the feedback from customers is through their reviews or by surveying them. These are very important customer service metrics, but are not typically being used to improve operations or help reduce customer churn.

Instead of waiting until the customer interaction is over for feedback/ review, companies can predict how likely an operation is to receive a good or bad rating while they are still in contact with the customer.

My main hypothesis is that the product and how the order was fulfilled influences customer review rating.

In this project, machine learning techniques will be applied to the dataset to predict customer review ratings.

Dataset:

Brazilian E-Commerce Public Dataset by Olist

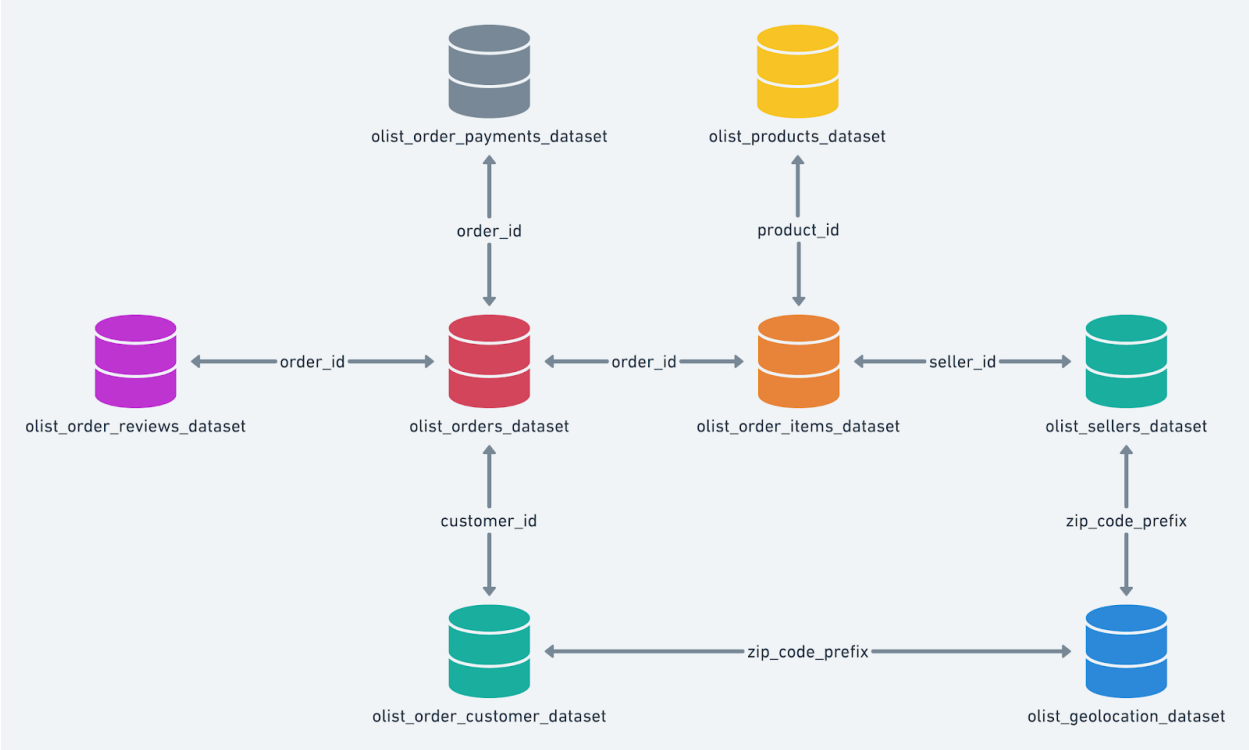
https://www.kaggle.com/olistbr/brazilian-ecommerce

This is a Brazilian ecommerce public dataset of orders made at Olist Store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. It's features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

This is real commercial data, it has been anonymised, and references to the companies and partners in the review text have been replaced with the names of Game of Thrones great houses.

Data Schema:

The data is divided in multiple datasets for better understanding and organization.



Data descriptions:

Customers:

Customer_id : key to the orders dataset. Each order has a unique customer_id.
Customer_unique_id: unique identifier of a customer.
Customer_zip_code_prefix: first five digits of customer zip code
Customer_city: customer city name
Customer_state: customer state

Order Items

Order_id: order unique identifier
Freight_value: item freight value item (if an order has more than one item the freight value is splitted between items)

Order Reviews

review_comment_title:Comment title from the review left by the customer, in Portuguese.
Review_comment_message: Comment message from the review left by the customer, in Portuguese.
Review_creation_date : Shows the date in which the satisfaction survey was sent to the customer.
Review_answer_timestamp: Shows satisfaction survey answer timestamp.

Order

Order_approved_at :Shows the payment approval timestamp.
Order_delivered_carrier_date: Shows the order posting timestamp. When it was handled to the logistic partner.
Order_delivered_customer_date : Shows the actual order delivery date to the customer.
Order_estimated_delivery_date: Shows the estimated delivery date that was informed to customer at the purchase moment.

Products

product_photos_qty:number of product published photos
Product_weight_g: product weight measured in grams.
Product_length_cm: product length measured in centimeters.
Product_height_cm:product height measured in centimeters.
Product_width_cm: product width measured in centimeters.

Category Name Translation

Product_category_name: category name in Portuguese
Product_category_name_english : category name in English

Methodology

I approached the problem from a supervised learning regression view. The values that are predicted show review score of the orders placed on Olist e-commerce site.

pandas for: data loading, wrangling, cleaning, and manipulation
feature selection and engineering
descriptive statistics

numpy for: array data structure, the primary input for classifiers
model comparison
matrix manipulation

scikit-learn for: regression models
parameter grid search

plotly for: data visualization

Seaborn: data visualization

matplotlib for: data visualization

Data Wrangling

This section describes data cleaning and wrangling methods applied to olist data set.

Firstly, the data which was in csv format was loaded and saved as a pandas dataframe. All the columns with datetime values were converted from object to datetime datatype.The majority of cleaning required the careful removal of NaN's and unnecessary columns.

The product names in the dataset were in portuguese. A new dataset products_in_english was loaded and mapped with current dataset to replace portuguese names to their English Meaning.

No. of orders delivered after estimated delivery date was calculated by calculating the difference between order_estimated_delivery_date and order_delivered_customer_date'.

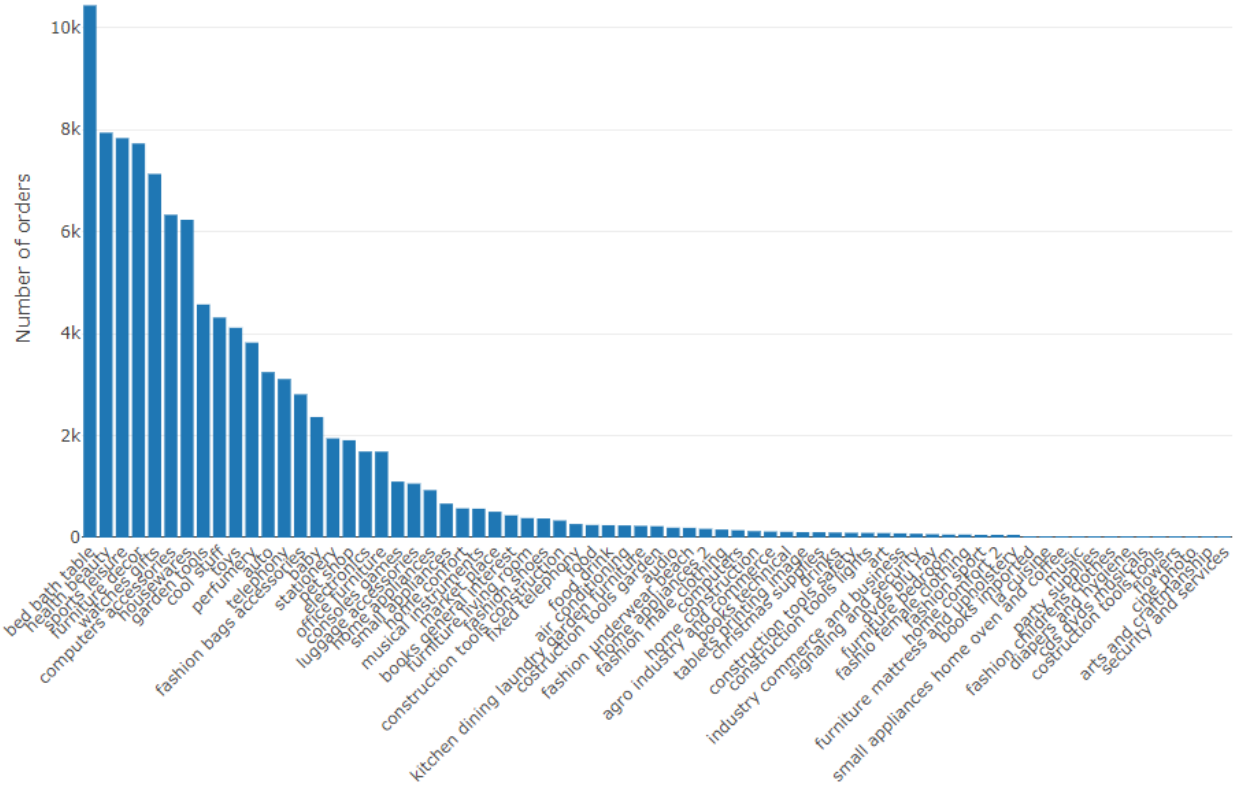
If the value is negative, it means the order was delivered after estimated delivery date. If positive, then the order was delivered before estimated delivery date.

The updated data frame was saved for further analysis.

Exploratory data analysis and visualization:

Once data wrangling and cleaning part was done, I started to explore the data to find correlation between different variables (features) from the dataset.

I tried initially plotting **Most bought product categories**



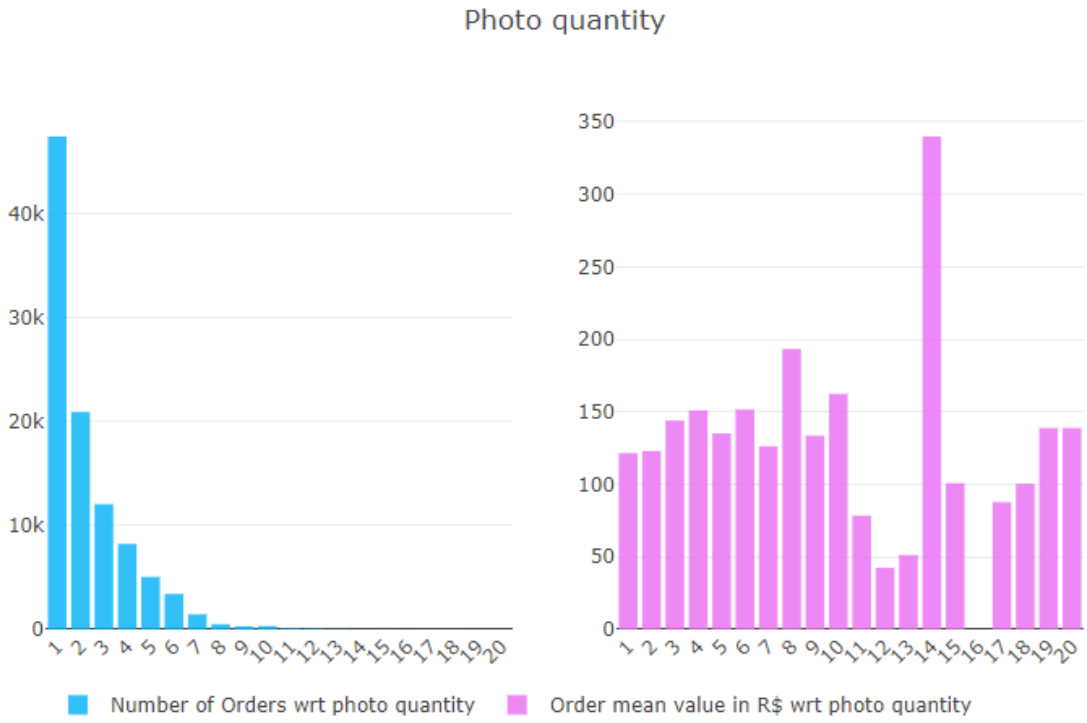
Bed Bath table,health beauty,sports are some categories that are bought most often by the customers.The difference in order count between bed bath table and health beauty is high.

Number of products people usually order

Most of them have ordered only 1 product.The number of people ordering more than 2 items is very less.

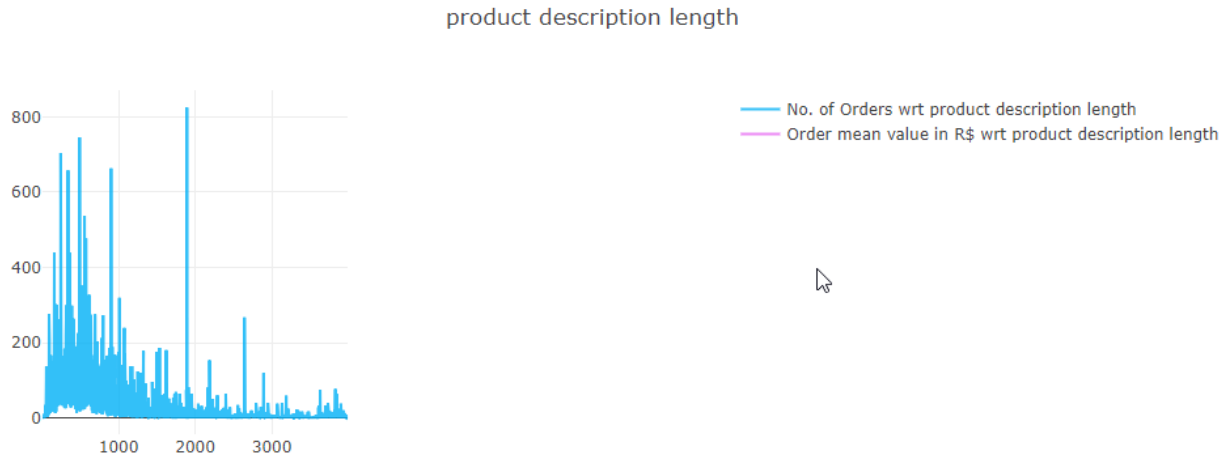
Number of photos and orders

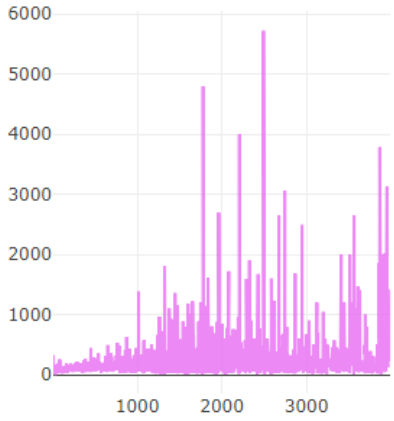
Later, I wanted to check how the number of photos available on site are related to the orders.



There are more orders for the products with less than 2 photo quantities.

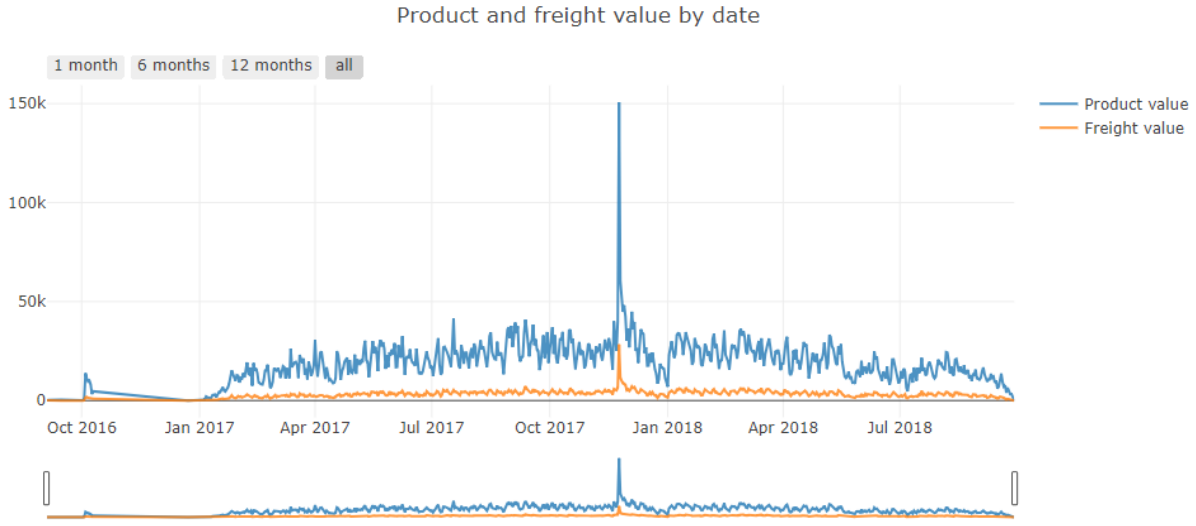
Product description length and order count:





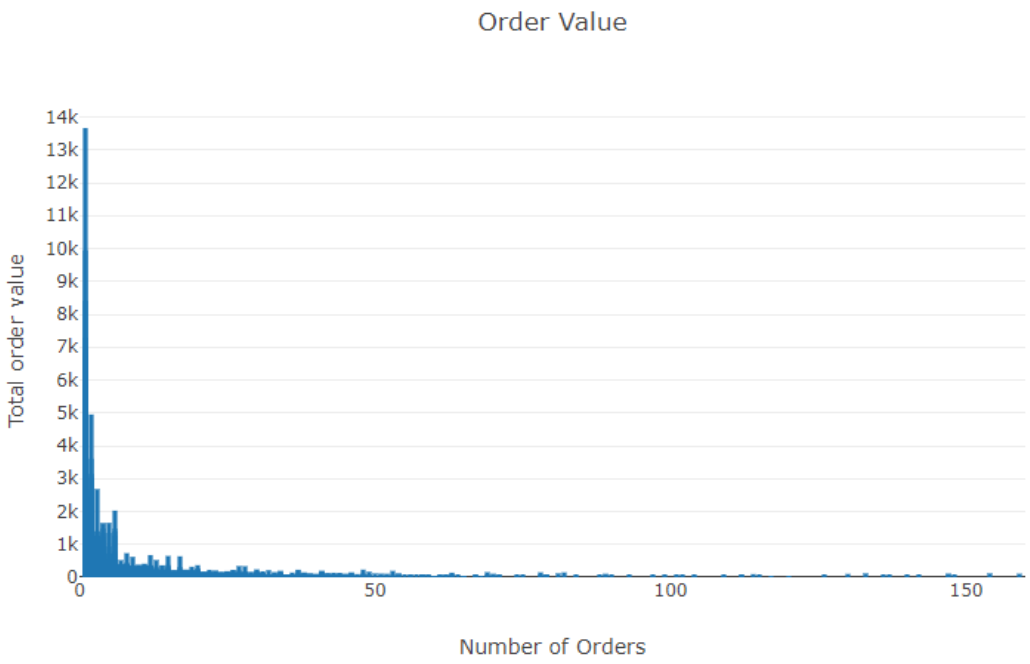
There is no significant relation seen between the product description length and order count. Products with description length less than 2000 words are ordered frequently than the products with brief descriptions.

Product and freight values by date



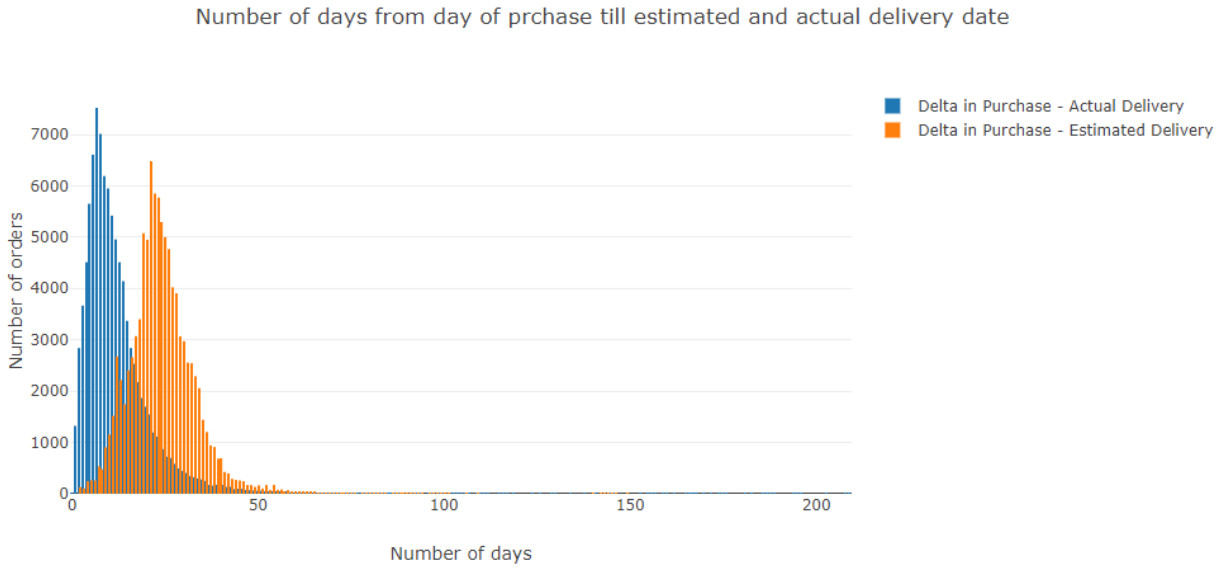
The freight value seems increasing with product value. There seems to be a similar trend in freight value and product value.

Money spent on each order/ Transaction value per order



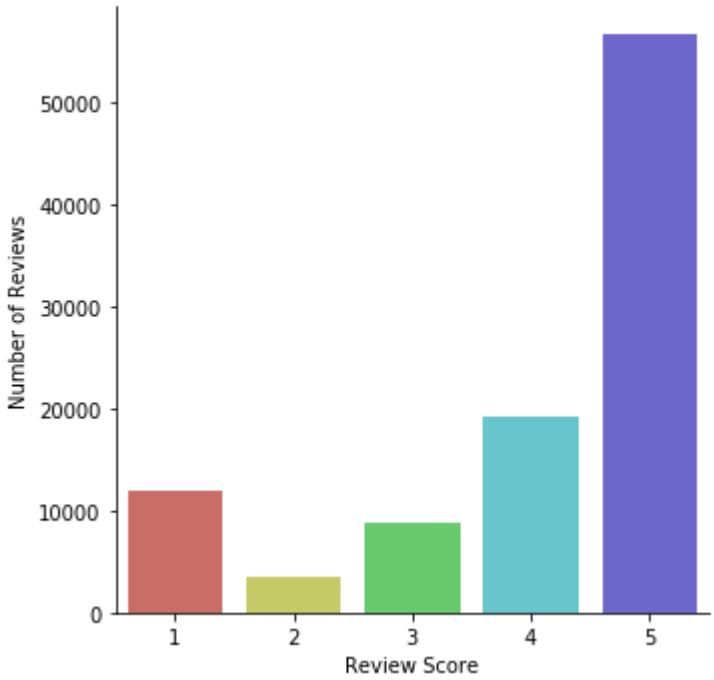
There are more orders for less valued products so it seems that most often customer buy cheap product on Olist.

Let's check the average number of days between order and delivery



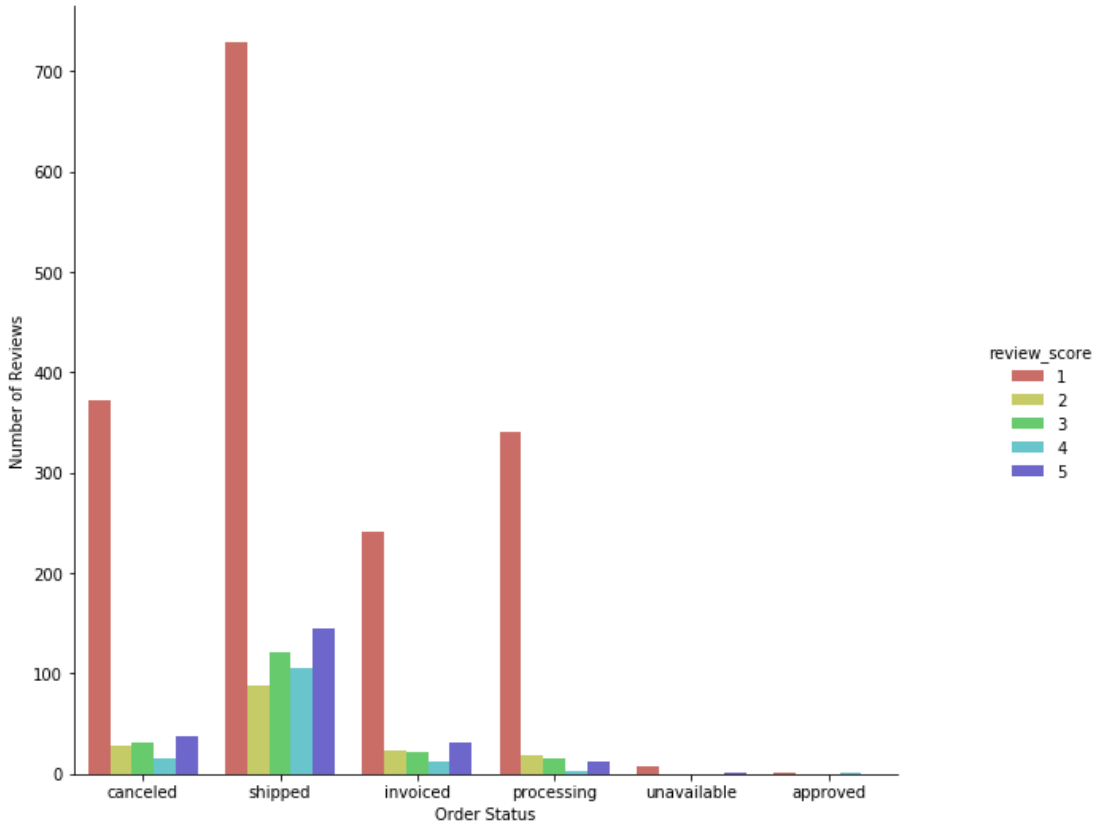
Most of the products seemed to be delivered before the estimated delivery date.

Explore review data



Here we can see the review score distribution. It is interesting to observe that there's more 1 star reviews than 2/3 stars reviews.

If we plot the review score distribution of orders that do not have a *'delivered'* status, we can see that most of them have a 1 star rating.



Just for my curiosity I plotted the review score and the length of the review. From the plot,it looks like unhappy customers are more likely to write comments and there comment are long.

From visualizing data, I could address the question of whether or not the delivery delays cause customer dissatisfaction.

Statistical Inference:

From data exploration, I did not get much insight on the relationships between dependent variable (‘review score’) and independent variable but I could come up with a few hypothesis mentioned below.

Feature Hypothesis :

What are the expected relationships of independent variables with review score?

1. order_items_qty ("+") - if consumer gets more than one item from the same seller, it should mean that he/she knows the quality of the good. Therefore, increase in item quantity should increase review score
2. product_description_lenght ("+") - buyer having more information about buying product should have positive relationship with review score
3. product_photos_qty ("+")
4. product_name_lenght ('unknown')- shorter names sometimes can be ambiguous or can be easy to understand,so relationship with review score is unknown at this point
5. delivery_accuracy ("+") - item coming on time or earlier that it was described should have positive relationship with review score
6. order_products_value ("-") - more expensive perhaps means better quality or higher expectation towards order fulfilment;
7. order_freight_value ("-") - more freight means more expectation towards delivery accuracy

To check above, I ran an OLS on these features.

```
result.summary()
```

Dep. Variable:	review_score	**R-squared:**	0.069
Model:	OLS	**Adj. R-squared:**	0.068
Method:	Least Squares	**F-statistic:**	1197.
Date:	Sat, 09 Feb 2019	**Prob (F-statistic):**	0.00
Time:	23:46:46	**Log-Likelihood:**	-1.6176e+05
No. Observations:	97595	**AIC:**	3.235e+05
Df Residuals:	97588	**BIC:**	3.236e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.0635	0.024	172.695	0.000	4.017	4.110
order_items_qty	-0.2464	0.009	-27.231	0.000	-0.264	-0.229
product_name_lenght	-0.0019	0.000	-4.616	0.000	-0.003	-0.001
product_description_lenght	4.038e-05	6.29e-06	6.424	0.000	2.81e-05	5.27e-05
product_photos_qty	0.0106	0.002	4.531	0.000	0.006	0.015
order_delivery_before_estimated_date_in_days	0.0317	0.000	79.803	0.000	0.031	0.033
order_products_value	-2.767e-05	2.2e-05	-1.261	0.207	-7.07e-05	1.54e-05

Omnibus:	17863.435	**Durbin-Watson:**	1.998
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	29327.189
Skew:	-1.288	**Prob(JB):**	0.00
Kurtosis:	3.757	**Cond. No.**	6.07e+03

Let's interpret the table.Overall the model is significant.

Prob(F-statistcs)= 0.000 This is the p-value associated with the F-statistic(F(6,97588)= 1197). It is used in testing the null hypothesis that all of the model coefficients are 0.

This tells us that there is a significant difference in the group means.

Coef. : These are the values for the regression equation for predicting the dependent variable from the independent variable. The regression equation is presented in many different ways, for example:

Ypredicted = b0 + b1x1 + b2x2 + b3x3 + b4x4

The column of estimates provides the values for b0, b1, b2, b3 and b4 for this equation.

order_items_qty:-0.2464 So for every unit increase in order item quantity, a 0.2464 unit decrease in review_score is predicted, holding all other variables constant.

product_name_lenght :-0.0019 for every unit increase in product_name_lenght, a 0.0019 unit decrease in review_score is predicted, holding all other variables constant.

product_description_lenght :4.038e-05 for every unit increase in product_description_lenght, a 4.038e-05 unit increase in review_score is predicted, holding all other variables constant.

product_photos_qty : 0.0106 for every unit increase in product_photos_qty, a 0.0106 unit increase in review_score is predicted, holding all other variables constant.

order_delivery_before_estimated_date_in_days: 0.0317 for every unit increase in order delivery accuracy, a 0.0317 unit increase in review_score is predicted, holding all other variables constant.

order_products_value :-2.767e-05 for every unit increase in product_name_lenght, a 2.767e-05 unit decrease in review_score is predicted, holding all other variables constant.

Looking at the p-values : P>|t| – This column shows the 2-tailed p-values used in testing the null hypothesis that the coefficient (parameter) is 0. Using an alpha of 0.05:

The coefficient for all the variables is significantly different from 0 because their p-value is 0.000 or smaller than 0.05.This means here's no difference between the means and conclude that a significant difference does exist.

All of our independent variables are statistically significant which is a great news.But It's clear that we have to use more informative features to model this problem.

Feature Engineering:

The first feature I want to engineer is **Estimated Delivery Time in working days**. Gets the days between order approval and estimated delivery date. A customer might be unsatisfied if he is told that the estimated time is big.

After researching on olist site, it was found out that the most common carrier used for delivery does not deliver on Sunday so we have to consider this while calculating the difference between time of purchase and time of estimated delivery.

The second feature I want is **Estimated Delivery Time in working days**. Gets the days between order approval and delivered customer date. A customer might be more satisfied if he gets the product faster.

The third feature is **Delivery Time Delta in working days**. The difference between the actual and estimated date. If negative- order was delivered early, if positive - order was delivered late. A customer might be more satisfied if the order arrives sooner than expected, or unhappy if he receives after the deadline.

The fourth feature is **delayed**. Variable indicating if the order was delivered after the estimated date.

The fifth feature is **Average Product Value.** Cheaper products might have lower quality, leaving customers unhappy.

The sixth feature is **Order Freight Ratio** If a customer pays more for freight, he might expect a better service.

The final feature I want to look at is **Purchase Day of Week.** Does it affect how happy are the customers? I do not have a lot of intuition behind this step but I have a feeling this feature may be useful.

After creating new features, it’s time to build a model. This will be covered as part of Machine learning section.

Machine Learning:

In this section I applied Machine learning models to the dataset to predict customer review rating score.

Hyperparameter tuning:

After feature creation, it was observed that different features in the data set had values in different ranges. For example, in 'delayed' column the value ranged from (-189.0, 146.0) and for product_value the value ranged from 2 - 13444.That means some column were more weighted compared to other.

Differences in variable ranges could potentially affect negatively to the performance of an algorithm so I used scikit-learn's Normalizer to normalize the data.

After normalizing the data between [0,1] I took the log values of the data. This introduced a few -inf and NaN values which I later on replaced with 0.

The data was still widely distributed so I set a lower and upper bound on feature values using clip function.

Also, the categorical columns were encoded using Labelencoder.

Dataset was then split into test and train datasets which was later using in modeling.

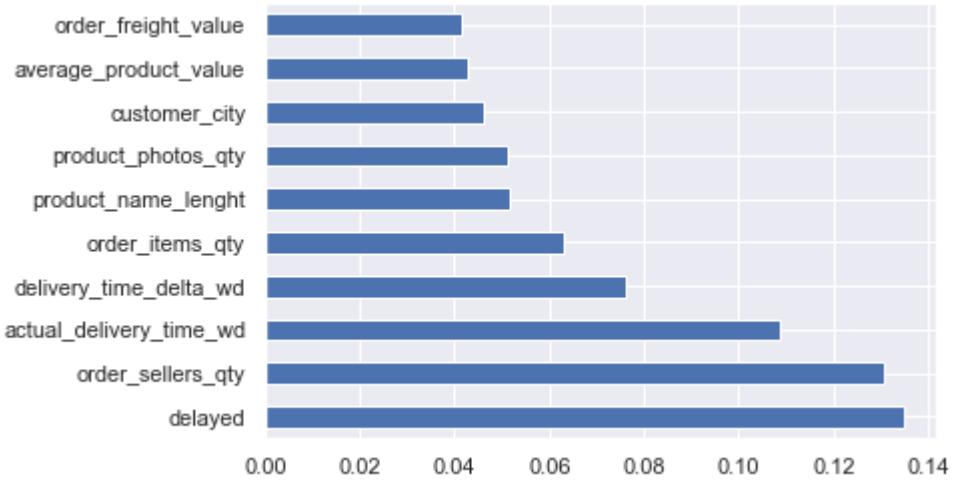
Modeling:

To predict the review rating score, I had to select a classifier that performs the best, given the features. I trained the model using three classifiers: Logistic Regression, Random Forest and Gradient Boosting, and chose the one with the best accuracy.

Accuracy score per model:
Random Forest : 0.5526
Logistic Regression: 0.59326
Gradient Boosting Classifier: 0.6074

GradientBoostingClassifier performed best with 60% accuracy so I chose Gradient Boosting Classifier as my model of choice.

To describe the features and their relative importance to a model, I used scikit-learn's feature_importances_ attribute and plotted the results on a graph as below



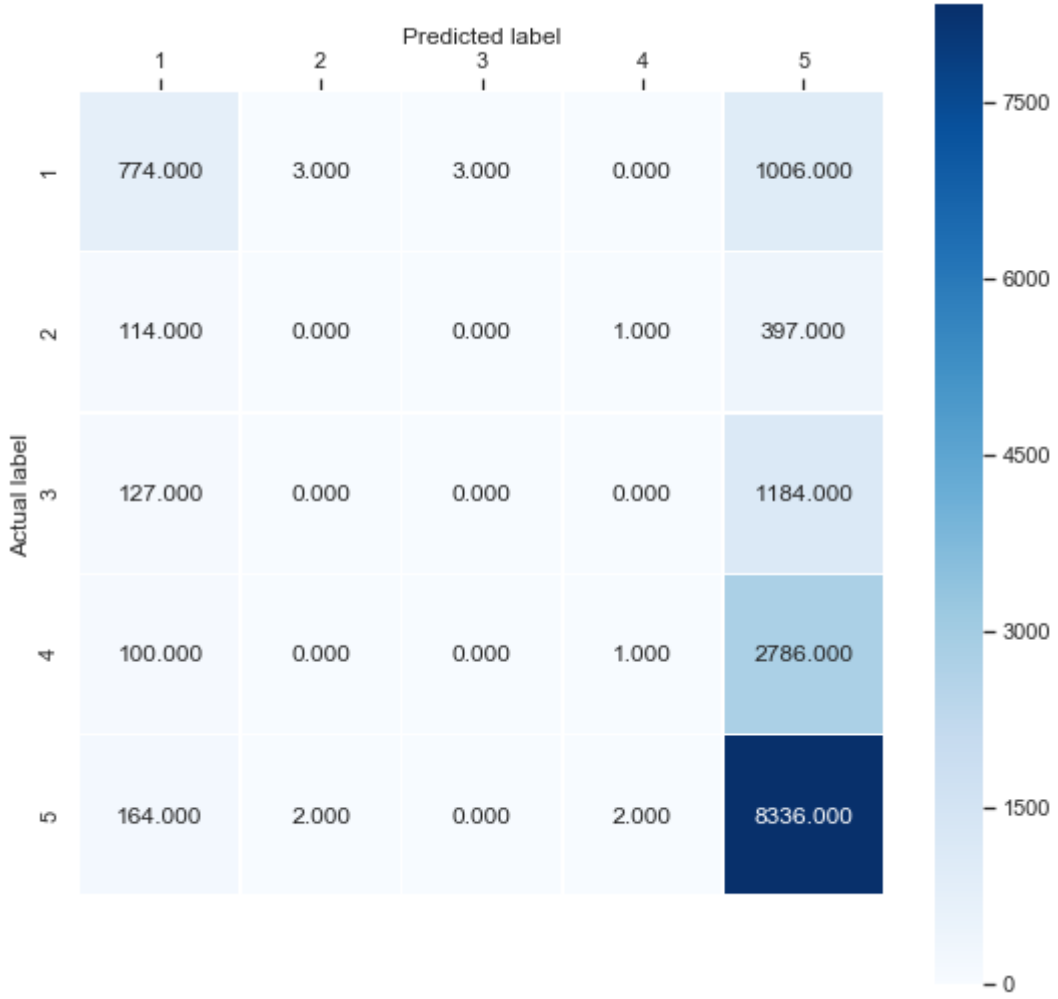
From above we can say 'delayed' is the most important feature.

A common approach to eliminating features is to describe their relative importance to a model, then eliminate weak features or combinations of features and re-evalute to see if the model fairs better during cross-validation. But since I had a less number of features I kept all for modeling.

To improve the model accuracy, I tried hyperparameter tuning using Grid Search Cross Validation. Fortunately, like always, scikit-learn has the tools available to us that reduces the amount of code to a bare minimum.

The score achieved with GSCV was 0.6088 which was not a significant improvement.

Later to summarize the performance of the algorithm, I calculated a confusion matrix and plotted it as a heatmap as below.



Review score 5 was classified most accurately than other review scores.

Later I calculated precision and recall:

	precision	recall	f1-score	support
1	0.61	0.43	0.51	1786
2	0.00	0.00	0.00	512
3	0.00	0.00	0.00	1311
4	0.25	0.00	0.00	2887
5	0.61	0.98	0.75	8504
avg / total	0.46	0.61	0.49	15000

Precision and recall is higher in both extreme scores: 1 and 5.

Conclusion:

The comparison between the three models in score classification is interesting. Although Random Forest is a complex model and it should be able of capturing more complex patterns, it does a really poor job in comparison with more simple and linear approaches such as Logistic Regression (55% vs 59% respectively in test set).

GradientBoostingClassifier performed best with 60% accuracy.

Another interesting point of the score classification can be seen in the confusion matrix. Precision and recall is higher in both extreme scores: 1 and 5. The possible explanation to this could be that the customers usually are satisfied (score 5) or not satisfied(1) so the number of reviews with score 1 and 5 are comparatively more than 2,3 and 4.

Limitations:

Unfortunately, unlike NLP, it is by no means easy to determine the state of the art model for review prediction.To predict review rating on the basis of product and delivery posses a data challenge.

Purchase decision processes are composed of several variables that influence customer's choice for certain products and many factors influence review rating such as

- Incorrect descriptions of the product
- Poor client service
- Poor response to information request
- Lack of communication
- Rude or Uninformed customer care staff
- Needs not accurately defined
- Promises not carried out
- Repeated complaints from the same customer

Thus, customer satisfaction can not be predicted solely based on objective facts without taking the vast amount of subjective sensor and service process data into account.