Capstone Project

Optimizing Customer Delivery Time

OLUWAFEMI JEGEDE

Email: jegede45@gmail.com
Data Science Bootcamp Sep-
Dec 2021

BrainStation

Last Mile Delivery: Optimizing Customer Delivery Time

Last mile delivery refers to the final step of the supply chain process where the order arrives at the customer's door steps. With the surge in e-commerce, there is a lot of pressure on logistic operations. The last mile is arguably the biggest problem in e-commerce as it addresses the question, ***"When will I receive my order?"***. Last mile delivery is the most expensive and time consuming part of the shipping process. According to Mantoria inc., a logistics company, up to 30% of overall delivery costs can be incurred in that final stage of order fulfillment. **So how does one reduce the cost and improve customer satisfaction by on-time delivery?**

Business Relevance

The focus of the last mile is to deliver orders as fast and as accurate as possible. We live in a fast paced world where time is very crucial. Customers need to know the delivery time window of their orders, to help them plan. In recent times, customers are constantly requesting for a narrower time window, accurate to 5 minutes especially in cases where the order is a perishable item. Setting thresholds is dependent on the type of business. For this report we will work with data from an e-commerce company with the Amazon kind of model.
There are a lot of factors that affect CDT, e.g driving time, size of order, weight of order, customer location, traffic, holidays, weathers, time of year etc. To be able to predict CDT, historical data on the listed factors are crucial.

Problem Statement

"How might we optimize customer delivery time by predicting future orders from historical data (including both external and internal factors)?"

We would like to build a model from historical orders and various derived features. The required features will be identified by an interactive process, from a broad range of available features in the data provided. The required range of features used for this project is divided in 3 broad categories:

- **Customer:** This includes but not limited to, customer location, type of customer, type of apartment etc.
- **Order:** This includes order size, order weight, purchase date, approval date etc.
- **External Factors:** This includes mostly weather conditions; Rainfall and Temperature. Others are Holidays, traffic etc.

The above includes only measurable and predictable features, features such as sick drivers, accidents etc. are not going to be included in our modeling.

The Data

The data to be used is the Olist public data sets available on Kaggle. This dataset was generously provided by Olist, the largest department store in Brazilian marketplaces. Olist connects small businesses from all over Brazil to channels without hassle and with a single contract. Those merchants are able to sell their products through the Olist Store and ship them directly to the customers using Olist logistics partners.**The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil on 8 tables**.

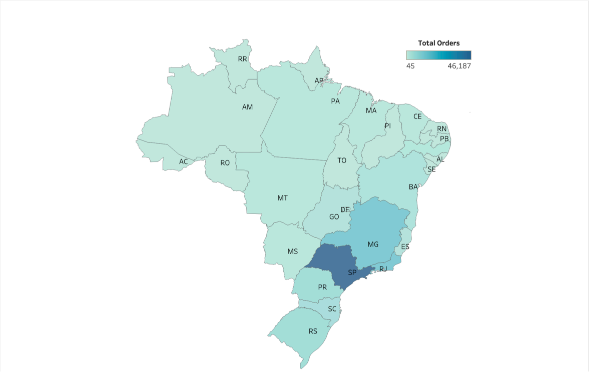
Exploratory Data Analysis(EDA)

The EDA commenced with a look at the ***customers table*** to see where most of the orders are coming from,the goal here is to show the highest source of the most orders.

It can be seen from the chart below that:

1. Most of the orders originate from the states of São paulo (~41% of all orders)and Rio de Janeiro (~13% of all orders),both coincidentally cities too.
2. Customers are spread across the country, however most customers located along the shoreline

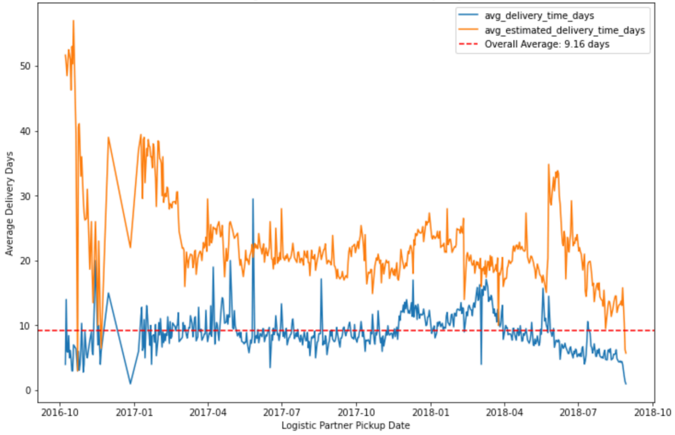
Total Orders per State



The above patterns can also be noticed in the ***seller's table***, São paulo and Rio de Janeiro take the lead in the states and cities with the most sellers. This is no surprise as both these cities are the biggest cities in Brazil. With regards to the ***orders tables***, there are a lot of features with respect to time and dates of e.g customer orders, order approval date, order carries date, estimated delivery date and actual delivery date. From these features the *delivery_time(days), estimated_delivery_time(days), approval_time(days)* etc was calculated.

Next, the actual delivery times were critically looked at, as it is the target variable here. We compared it with estimated timeline that was calculated to see, the performance of Olist team.

Average Estimated and Actual Delivery Days



From the figure Olist almost always overestimates delivery times the magnitude of over estimation can be seen in the figure below. Also on the average actual delivery time takes about ~9days, although it seems stable at this level a closer look shows a lot of instability in the delivery time.

It can be seen that Olist Overestimates delivery by about **11days** on the average, which means they predict **20days** on most deliveries but delivers on the 9th day on average, these are just average figures. **Deliveries above 20days are ~10% of the data** and actual delivery from EDA shows delivery can take up to **200days** after logistics partner pickup. This is a huge problem, as this is causing a huge imbalance in the data set which was addressed in feature engineering. **Consolidating the data tables** in the schema; *customers, sellers, orders, ordersItems and products, w*e end up with **~110k rows and 30 columns.**

Feature Engineering

Features such as delivery_time(days), estimated_delivery_time(days), approval_time(days) have been added to our dataset for various analyses in the EDA. This process continued with addition of the following features:

- Distance
- Holidays
- Time based features (day of week, week of year)
- State category- This was due to imbalance in our state data
- Same state delivery
- Product weight and size

Features such as product_weight, price, freight value and product_volume were log transformed to balance the features which were right skewed. It is known from the EDA that the target variable is imbalanced with delivery timeline up to ~200days. From research last mile delivery companies such as Olist that run amazon kind of model, have delivery windows for different types of orders based on different conditions. The best way to sort this imbalance is to bin the target column `delivery_time_days` into different delivery windows, for a more efficient experience since one does not control the third party delivery company or the sellers.

Logically the deliveries were broken to 3 delivery windows:

- Within 3 days of pickup
- Within 3 days - 1 week of pickup
- Over one week of pickup

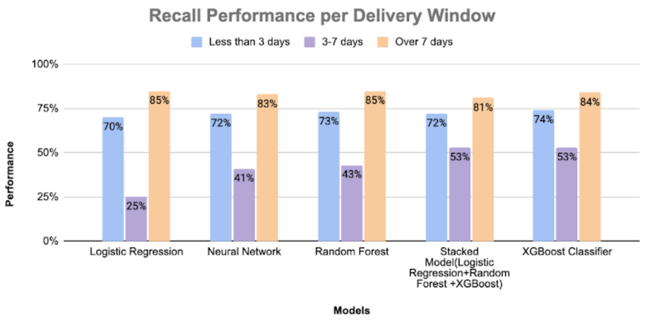
For our modeling the focus was on the aforementioned categories of features in our problem statement; *customer, orders and external factors*. **12 features** were finally selected for modelling.

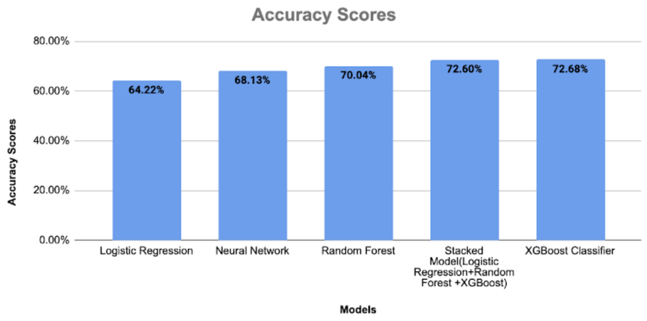
Modeling

The focus of our model is to predict the customer delivery time accurately, that is we want to be able to predict with certainty the time window of a customer delivery. This means we want to optimize the **recall.**The data was split into training, validation and test sets, then scaled. This is to prepare the data for modeling. In terms of the modeling, a logistic regression was used as base model, and Random Forest, XGBoost and Neural Network were used as secondary models.

For each of the above list models, the base accuracy was obtained and the model hyperparameters were tuned using on the validation sets and finally the tuned model was run on the test sets to get the accuracy scores and a classification report.

Findings





After running all the models the XGBoost model was the best in terms of accuracy scores, with an accuracy score of **72.68%.** The models are doing generally badly in the 3-7 days category, except the XGBoost and Stacked Model, which was set up for the purpose of correcting the mistakes made by other models. The accuracy and recall scores for the Neural network was very surprising as it is expected to be the strongest model. It was expected to recognize the underlying patterns in the data but after several iterations the accuracy was always **stuck at ~69% accuracy on validation set**.

Next Steps

The **business application** on this model is enormous as it can be used to help customers and logistics companies plan delivery times based on delivery windows as used in this project, thereby reducing cost associated with last mile delivery problems.

Although the model currently does a great job at predicting delivery time with overall accuracy of 72%, there are a few requirements and implementations that would improve the work already done on this project.

1. A more robust dataset is required that would give a good level of detail on the logistics partner operations to help with an overview of their fulfillment centres and work schedule. This will help predict the hour of the day of delivery to the nearest minute.

2. Explore and research more on how Neural network can help in improving accuracy of our model

3. Investigate further on why recall for the 3-7 days delivery window, performing badly.