Predicting Driver Supply for Loggi's Last-mile Delivery

Buenos Aires | Team #7 - Spencer's preferred team ;)

Ariel Kanelson, Ezequiel (Kecol) Bidart, Lucas Ansaldo, Pablo Schpilman

Business Problem

Loggi is a tech unicorn that offers next-day delivery to nearly anyone in Brazil. Through a network of distribution centers (aka agencies) across the country, including small warehouses in neighborhoods, the company leans on robotics, artificial intelligence and algorithms to find the most efficient ways to deliver packages. For last-mile delivery, Loggi relies on tens of thousands of partner drivers who are independent contractors, making it the most expensive and non-efficient portion of their supply chain. For Loggi is critical to predict driver supply in order to better allocate drivers to itineraries. We are interested in understanding the patterns in driver's motivation to accept a new itinerary, and ultimately, what's the capacity of an agency to allocate itineraries during the day.

Business Impact

With an optimal amount of itineraries by agency, we estimate that Loggi can reduce the amount of dropped itineraries up to 2x, and speed up the order delivery time by 30%. Therefore, understanding the patterns in driver's motivation to accept a new itinerary and predicting the optimal amount of deliveries that agencies can allocate during the day could have a substantial impact on the customer experience and the company's bottom line.

Data

The Itinerary dataset consists of ~21K records with the following columns:

- `itinerary_id (String)`: Unique identifier of the Itinerary
- `driver_id (String)`: Unique identifier of the Driver
- `created (Datetime)`: Itinerary creation datetime
- `accepted (Datetime)`: Driver's acceptance datetime
- `dropped (Datetime)`: Driver's dropped datetime
- `started (Datetime)`: Driver's started delivery datetime

finished (Datetime): Driver's finished delivery datetime
status (String): Status of the itinerary. Can choose from: finished, dropped, waiting pick up, cancelled, cancelled with charge, started, allocating, requires verification, awaiting completion
total_distance (Int): Distance traveled by driver from accepting the itinerary to delivering it
transport_type (String): Vehicle type used by the Driver. Can choose between moto and carro
product (String): Not sure. Useful?
product_version (String): Not sure. Useful?
distribution_center1 (Int): Distribution center (aka agency) the itinerary belongs to
packages (Int): Amount of items an itinerary contains
delivered_packages (int): Amount of items delivered to the client
checked_in_at (Datetime): Driver's checked in datetime into the agency
pickup_checkout_at (Datetime): Driver's checked out datetime from the agency with the package to deliver to the client
pickup_lat (Float): Driver's latitude picking up the packages (agency location)
pickup_lng (Float): Driver's longitud picking up the packages (agency location)
real_completion_time (Float): Total time allocated by the driver to fulfill the package delivery
pickup_distance (Float): Driver's distance traveled from the acceptance to arriving to the distribution center (agency location)
pickup_time (Float): Minutes the driver is in the agency picking up the packages
check_in_time (Float): Minutes the driver took from the acceptance to arriving to the distribution center (agency location)
waypoints (Int): Amount of records sent by Loggi's app during a driver delivery

This dataset is rich in understanding the client's volume of requests in a timeline and getting different statistics of each of the fields. The disadvantage of the dataset is that we don't have the workflow's states that each itinerary goes through and there are data inconsistencies across many datetime fields.

The Driver dataset consists of ~21M records with the following columns:

id (String): Unique identifier of the driver's availability location

`driver_id (String)`: Unique identifier of the Driver
`itinerary_id (String)`: Unique identifier of the Itinerary
`lat (float)`: Current latitude of the Driver
`lng (float)`: Current longitude of the Driver
`sent (Datetime)`: Driver's availability location datetime sent to the server
`transport_type (Int)`: Vehicle type used by the Driver. Can choose between 1, 3, or 4

This dataset is rich in knowing drivers positions every 5 minutes to better understand routes, speed, distance and other attributes for analyzing driver's logistic matters. No disadvantages spotted at this time.

The **Distribution Center**, **Traffic**, and **Holidays** datasets were not provided.

Methods

Visualizations
One key component of understanding drivers' motivation to accept itineraries is providing high-quality visualizations that determine the association with traffic and weather conditions, among other factors. Also understanding itinerary volumes over time. Here are some of the static and interactive visualizations we will provide:

- Correlation plot of drivers itinerary acceptance vs traffic and weather.
- Bar plot the volume of itineraries by weekday and by hour of the day
- Plot the average order delivery time by location with a heat map

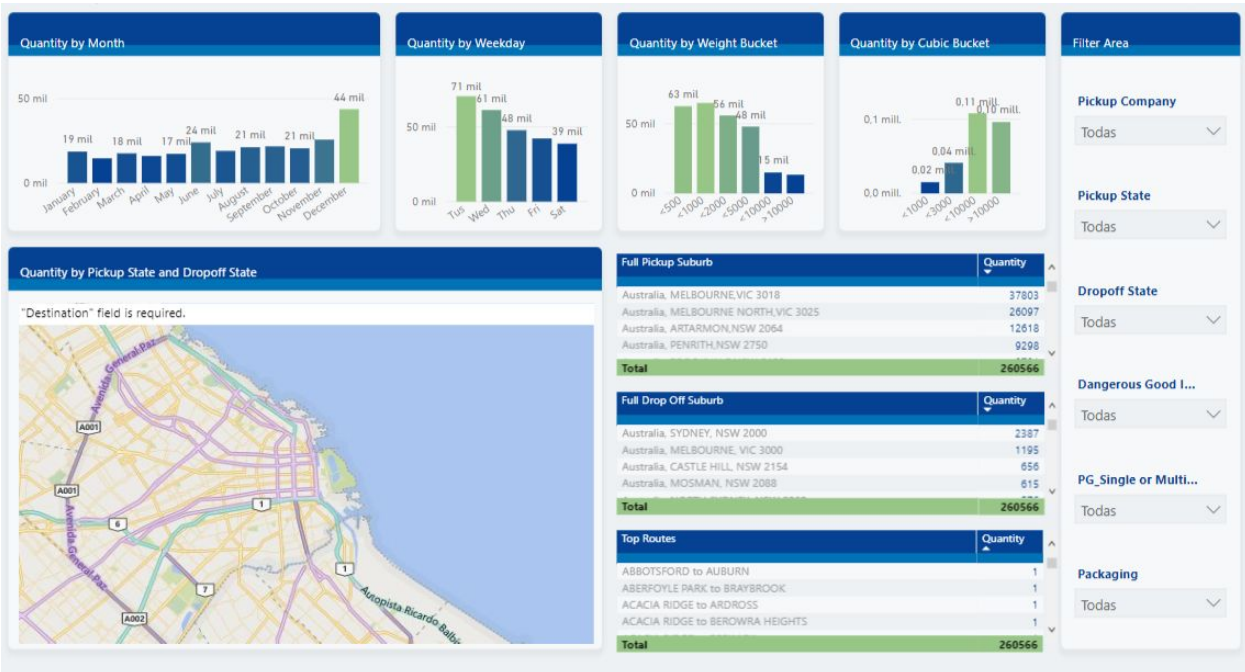
Models
Another key component of our project is predicting the optimal capacity of agencies to handle itineraries during a day. At this point, we don't have enough information to define what models we are going to use as they will be covered in the following weeks of the training.

Interface

The main stakeholder and end user of the interface will be the Operations team at Loggi. However, it will also help inform resource allocation and investment decisions, so other teams such as Strategy & Ops and Marketing can benefit from it as well.

The final front-end product will feature two landing pages: an Analytics dashboard with visualizations of the historical data, and a Recommendation page, where the results are summarized with a recommended amount of deliveries that agencies can allocate during the day. The interface will allow for interactive visualizations of the historical data, and users will be able to drill down on a particular agency and see the results for a day, or all outcomes over time. If time permits, we will also try to include a feature in the Recommendation section where a user can see what the impact is predicted to be if Loggi allocates X% more or less itineraries to an agency than what's recommended.

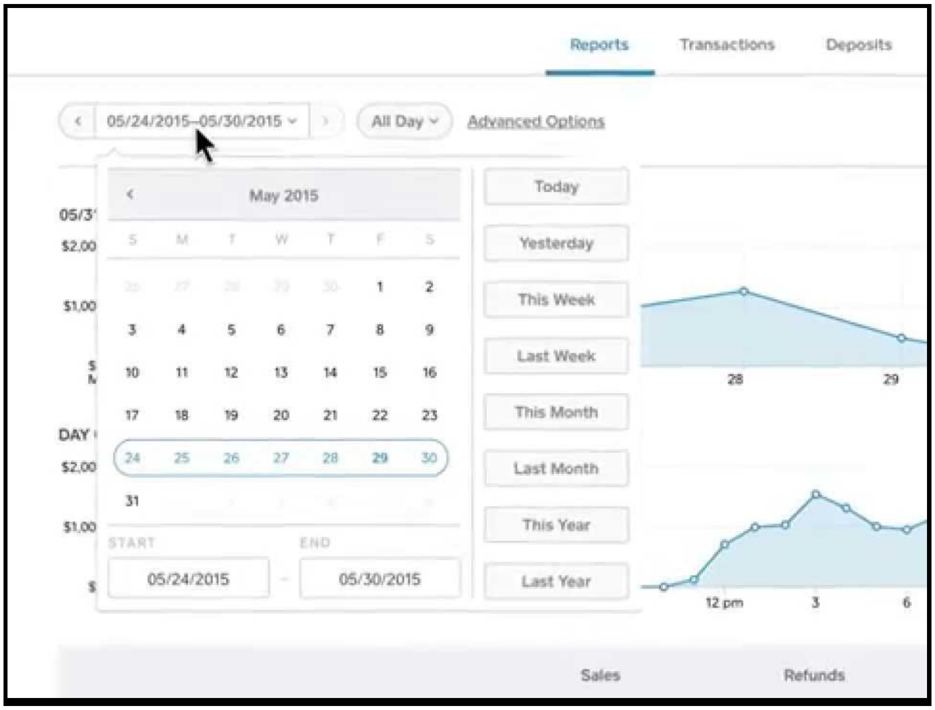
Analytics dashboard:



This is a mock reference of what our Analytics dashboard could look like: (1) bar plots with volume of itineraries (accepted/dropped and assigned/fulfilled) by weekday and by hour of the day, (2) a heat map with a time slicer with the average order delivery time by location, (3) a filter section with State, City, and Agency.

Recommendation page:

[Scroll down menu with Agencies]



This is a mock reference of what our Recommendation page could look like: a scroll down menu where the user selects the Agency to receive a table with its optimal capacity of itineraries by day and hour of the day.

Milestones

In this section, we provide details on the milestones we intend to achieve in our project. In particular, we have outlined 4 different versions:

- **Version 0**: Build and configure basic tools and deployment procedure on the cloud! AWS EC2 + RDS, GitHub, initialization scripts, and so on.
- **Version 1**: Build an MVP of the Analytics Dashboard with bar plots of volume of itineraries by weekday and hour of the day.
- **Version 2**: Complete our Analytics Dashboard. Build a heat map with a time slicer featuring the average order delivery time by location and a filter section with State, City, and Agency.

- **Version 3**: Set up the Recommendation Page. Build a simple predictor functionality to estimate the optimal capacity of itineraries that an agency can handle during a day.
- **Version 4**: Add complex predictor functionality to estimate next hours and days delivery capacity based on other external features/datasets.

Timeline

Date	Deliverable	Details
Week 1	Team formation Environment setup	
Week 2	Project selection Workflow setup	*This includes getting data access, e.g. NDAs.*
Week 3	Scoping document Data access	*The scoping document is the skeleton of the Final Report.*
Week 4	Data cleaning Initial data exploration	*Update Final Report to include EDA results.*
Week 5	Continue data exploration	*Get initial code reviewed by TA as well.*
Week 6	Advanced data exploration Initial modeling	*Begin front-end visualization.*
Week 7	Continue modeling Application on cloud	*Update Final Report to include initial modeling results.*
Week 8	Front-end complete Advanced modeling	*Update Final Report to include final modeling results.*
Week 9	Fine-tune modeling Fine-tune application	*Write Conclusions section of Final Report.*
Week 10	Finalize presentation, report, and application	

correlation one

Concerns

The primary concerns with our project are (1) that three of the five promised data sets were not yet provided, (2) time constraints given a demanding program schedule, and (3) that some of our team members have no previous Data analysis, Python or SQL knowledge.