Case 3: Improving last mile logistics with Machine Learning

After your last engagement with Charles, you have pretty much become the go-to service provider for advanced quantitative methods. Congratulations!

You have been called again to help a different manager within Beanie Limited: Estefania Pelaez. Estefania is the city manager for Barcelona. She is in charge of all commercial and logistic operations that happen in the city.

One of the operations that Beanie Limited runs in Barcelona is their own last-mile coffee delivery service. The company runs a small fleet of vans and trucks that delivers small quantities of roasted coffee beans (typically, around 10-100kg of coffee per delivery) to restaurants, cafes, hotels and other businesses in the city.

The efficiency of the deliveries is important to keep margins profitable for Beanie Limited. A sloppy management can make the company lose money. Hence, Estefania is always working on ways to make the operations as smooth as possible.

Currently, Beanie Limited has rented space in two warehouses: one located in Zona Franca and another one in Baro de Viver. Complementing that, the company has a small fleet of combi vans, regular sized vans and one truck, which are used by Beanie Limited own drivers to deliver the coffee beans from the warehouses to the customer's facilities.

Orders placed by the customers are predictable and placed with time in advance, which allows Estefania and her team to plan the deliveries to minimize wasted effort by the fleet. Since they know which locations they will need to deliver to, they use a routing software that drafts the routes that each vehicle will cover each day.

Recently, Estefania realized something: deliveries are almost always taking place too early or too late. After researching with some data, Estefania found out that there was nothing wrong with the routing software time estimates: the driving time between locations predicted by the software is accurate. The real issue is related to what Estefania's team calls the "engine-off" time.

The engine-off time is the time a driver spends actually dropping off goods in a client location. It's called engine-off because the clock starts ticking when the driver takes the keys off the van and stops when the driver starts driving again.

Currently, Estefania and her team assume an engine-off time of 3 minutes for all deliveries when building the delivery routes and schedules. But it seems that this not realistic at all and is causing a lot of trouble with the schedules. Clients are not happy with delivery times not being respected, some driver routes end up too early (which means that the same driver could have covered more clients) and some others run for too long (which means they have to go back to the warehouse without delivering all the goods requested by the clients).

If Estefania could know beforehand what would be the engine-off time of different deliveries, she could improve the route planning to fix all of these issues. She has been told that Machine Learning could help with this issue and is expecting you to find out if and how it can be applied to this problem.

Detailed Task Definition

Below you will find four levels of questions. Levels 1 to 3 are compulsory. Level 4 is optional. The special section is also optional.

You need to write a report document where you answer the questions of the different levels. This report should be directed towards Estefania. It's important for you to reflect your methodology to back your proposals.

- Each level is worth 2 points out of a total of 10. The 2 missing points will grade the clarity and structure of your report and code.

You need to use a Python notebook to solve all levels. Please attach a notebook that shows your solution/proposal/analysis. Your notebook should be runnable "as-is". That means that anyone should be able to run it from beginning to end without any additional instructions or action required (except for uploading data from a CSV in the Google Colab environment. That requires someone to upload the file with a few clicks and it's fine).

- Include your team number, names and student IDs in all your deliverables.

Data

By joining the customer database together with past deliveries details, Estefania has built a dataset of executed deliveries. The table contains 9,000 examples of past deliveries and their engine-off times. The exact field meanings are explained below:

- client_name: the name of the client.
- truck_size: what type of truck was being used. Can be one of Combi, Van or Truck.
- truck_origin_warehouse: from which Beanie Limited warehouse did the route start.
- delivery_timestamp: at what date and time was the delivery done (defined as the moment the engine-off time starts).
- total_weight: total weight of the goods delivery.

brand_1_coffee_proportion: what percentage of the delivered goods was of Beanie's brand #1.

brand_2_coffee_proportion: what percentage of the delivered goods was of Beanie's brand #2.

brand_3_coffee_proportion: what percentage of the delivered goods was of Beanie's brand #3.

- driver_id: the ID of the driver that was driving the route.
- is_fresh_client: whether the client was fresh at the date of the delivery. Fresh clients are clients that have been doing business with Beanie for less than 30 days.
- postcode: the postcode of the client location.
- business_category: whether the client is a hotel, a cafe or restaurant or a coffee retailer.

- floor: the physical position of the client location.
- partnership_level: indicates the partnership level with Beanie. Key Account are important clients for Beanie Limited. Diamond clients are the top priority clients for the company.
- box_count: how many distinct boxes were delivered to the client. The coffee beans bags are grouped into boxes for delivery.
- final_time: the engine-off time, measured in seconds.

Notebook

Case 3 comes with no helping notebook: this time, you will have to code things from scratch yourselves. Remember that you are still suposed to write and deliver a notebook (see the "Detailed Task Definition" section).

A few comments on your notebook:

- I'm a going to constraint you to use scikit-learn as a ML library. You can of course use other useful Python libraries such as pandas, numpy, etc. But for ML modeling, please go with scikit-learn.
- Below you can find some useful materials which relate to what you need to do as part of the case:
  - A simple, guided EDA on the Titanic Dataset
  - A guide on regression performance metrics and some material from scikit-learn on the same topic
  - An introduction to cross-validation
  - A thorough review on why we need to use baselines in ML
  - A simple introduction to linear regression with scikit-learn .

Levels

Level 1

- Assess for Estefania if ML is a good choice for her problem and explain why.
- Perform Exploratory Data Analysis on the given data. Is it clean? Which variables could be useful to explain the engine-off time? Are there any other interesting things you can draw from the dataset?

Level 2

- Present how are you going to measure performance for this problem and how you will use the available data for testing it.
- Develop a baseline algorithm and evaluate its performance.

Level 3

- Develop the best model you can make to predict engine-off time.
- Explain your methodology and report on performance.

- Compare your performance to the baseline algorithm. Reflect on what is the cause of whatever differences can be observed between both.

Level 4

After presenting your model and results, Estefania has two different questions:

- Estefania would like to learn from the ML algorithm. What are the most relevant features that define the engine-off time? Can you somehow quantify how important each is or which are most useful?
- Estefania is interested in learning about next steps. What can be done to improve even more the model performance and achieve better results?

SPECIAL

For this case, we are going to run a little competition. There will be a surprise gift on the last lecture for the team that wins.

The competition consists of getting the best performant model of the course. I have a hidden part of Estefania's dataset. If you present a notebook with a working model before 13/06 - 23:59 (note this is earlier than the case delivery time). I will use your model to predict engine-off times on the hidden data. The team with the lowest error will win.

To enter the competition, write a function at the end of your notebook called `predict_to_compete`. The function should take as its only input a dataframe with the same format as the shared dataset. The function should return a numpy array with the predicted drop-off times.