Module 3: TDD

Technical Description Document - Push Notifications DS

Context

We are building a new product to enable the sales team to target customers to be sent a push notification to nudge them to buy a certain product that has been previousle selected by them.

This tool is fractioned in 3 parts:

- Frontend: being carried out by the FE team.
- Backend engineering: all the processes for having the data available and serve it to the frontend. Engineering team is on charge of this.
- Data science: development of a predictive model that suits the sales team need. This is covered below.

Goal

Developing a machine learning model that, given a user and a product, predicts if the user would purchase it if they were buying with us at that point in time.

We should only be focusing on purchases of at least 5 items, since it is a requirement coming from the sales team. They expect from us to ship a Proof Of Concept (POC) in a week.

Data

We will be using our groceries dataset *feature_frame_20210304.csv*. This is a well known dataset by now since we have being working on it over the latest several weeks. For this reason, this project skips the Exploratory Data Analysis (EDA) phase as it is should already be available in previous reports.

The only tweak required in the dataset is filtering orders to keep only those with at least 5 items, since they are our target.

Approach

Milestone 1: exploration phase

Given that we have a clear understanding of our data, we just jump directly on building the predictive model. Firstly, we filter the data to only those orders with 5 items or more to build a dataset to work with.

Secondly, since we want to have a POC in a very short timeframe, we will limit ourselves to using linear models which we will evaluate using train/validation/test split.

The outcome is expected to be a report/notebook/documentation on what worked and what did not and whys Most importantly, we need to have a final model selected to move to milestone 2.

Milestone 2: MVP code

Using the outcomes of milestone 1 MVP ready code must be generated to share with the engineering team for deployment. For now, as this is just the MVP, we just want to make sure that we have production-ready code for our pipeline.

Some of the steps that should be present in such a pipeline are:

1. Data loading: loads data and applies validations if required.
2. Pre-processing: responsible of any pre-processing step that may be required.
3. Model training/selection: If applicable train model with different parameters and evaluate their performance to choose the best performing one. Trains a final model and save it to disk to be later used for inference. Since we are not using any ML Engineering Framework, think what is a good standard to save the trained models so we can keep track of the history of the models.