Data science coding project

26.03.2020

Goals

With this project we will assess your capacity to solve problems on your own and present a fully working solution to a data science problem mirroring the ones we are working on daily at Wiremind.

You will be given a chance to show your skills in data science: code, design, autonomy, rigor, etc.

At Wiremind, we leverage machine learning to deduce the willingness-to-pay of train travelers. The models are trained on structured, cleaned and preprocessed dataset that are extracted from raw, unformatted data sources such SQL tables. The goal of this project is to train such a model based on a few extractions of relevant tables. The data transformation, preprocessing, training and validation will be handled by your code. The data and its underlying trends is close to real data of our customers.

Specifications

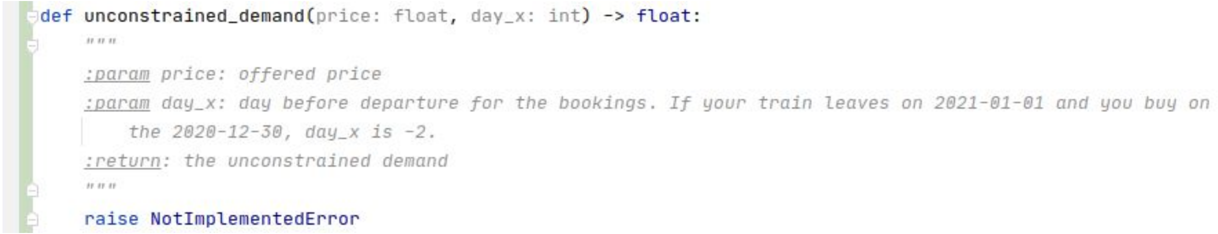
With this instruction notice, you have been given a .tgz folder containing a CSV file.

passengers.csv is a list of all the bookings made on a fictional O&D (origin – destination) A-B for the past 3 years.

Column	Description
departure_datetime	Train departure datetime in isoformat
station_origin	Station at which the passenger is boarding
station_destination	Station at which the passenger is disembarking
sale_datetime	Datetime at which the booking was made
cancel_datetime	Datetime at which the booking was canceled. Will be empty if the booking has never been canceled

price	Price paid for this booking

You are asked to write a clear pipeline composed of three components: features extraction, training and validation to deduce from the raw data the daily unconstrained demand. We define the unconstrained demand as the amount of people ready to pay P to buy a ticket for A-B for a given Day-X before departure. Thus your model (or composition of models) should be a function with the following signature:



```python
def unconstrained_demand(price: float, day_x: int) -> float:
    """
    :param price: offered price
    :param day_x: day before departure for the bookings. If your train leaves on 2021-01-01 and you buy on
        the 2020-12-30, day_x is -2.
    :return: the unconstrained demand
    """
    raise NotImplementedError
```

In reality, the information of the offered price during a day before departure would come from another data source but, for this test, it can be deduced from the sales in the CSV. Day where the price changed will have tickets with different prices while day without any booking will have no lines associated with it in the CSV.

Moreover, and this may be of importance in your design, the model will be used on circulation not yet departed but for which we already have seen some bookings (for instance, we will use the model to predict DAY - 0 / -1 / -2 given that we know all booking made up to DAY-3). Your model can leverage all the available information from previous sales and/or events (trend features).

The programming language is **python**, you are free to organize your code the way you want with the tools/frameworks you feel comfortable with (tensorflow, scikit, pandas, pytorch, jupyter notebook…). The three parts of your pipeline should be easily identified and separated in your code in a manner that would allow it to be deployed on a DAG supported framework such as Airflow or Kubeflow.

We intentionally are not giving any strict instructions on how the feature extraction, training or validation should be performed to let you explore and think about each problem. Given the time you have, you may not be able to implement every of your ideas, do the most important ones and write a readme.md with your eventual follow ups. In addition your package should at least contain a quickstart.md file allowing us to easily run your pipeline with fresh data along a dockerfile to launch it.

For this test, more than the precision or architecture of your model, what we want to assess is the quality of your code in terms of maintainability, readability, organization and ease to extend. The second goal of this test is to see how well you understood the problems at play with the given data and what you have done or would do to solve them.

Project presentation

We expect you to send us a link to a public repository (github, gitlab, bitbucket...) before the agreed limit date. We will then review your code and, if this first internal review is positive, will ask you to come present the results of your work to us.

The presentation will allow you to explain your technical decisions, and the problems you encountered.

Good luck!