Take Home assignment

The assignment is divided into 2 parts:

1. solving the data science challenge described below
2. presenting your solution during the interview with notebooks and, if you want, slides

The scope of the assignment is to evaluate the candidate on multiple criteria:

- data science skills
- creativity and problem solving
- presentation skills

Challenge

In the car insurance sector, an insurer can sell its policies to customers through different distribution channels. The channel we are talking about in this assignment is a price comparison website (PCW) where the customer has the opportunity to compare prices for similar insurance products from different companies.

To give some context, a quote (or quotation) is when a potential customer declares a set of personal information required by the site and receives 1 price maximum for each company on the PCW panel.

Imagine that you act as one of the insurance companies operating in such PCW, and you have to provide a price that is in line with the rest of the market using all the declarative information provided by the potential customers.

Rules and evaluation metric

The rules are the following:

1. if you are cheaper than the cheapest price of all your competitors (`competitor_lowest_price`), you sell the policy
2. you can decide to not provide a price to certain customers, but you won't sell anything
3. you must sell at least 30% of the quotes in the test set
4. you have to minimize the difference between the cheapest price of the competitors and your proposed price. More precisely, minimize the average absolute difference calculated only on the policies you sell.
5. write clean code and have fun!

To give an example of the evaluation metrics that will be evaluated on the test set (named as `test` in the following example using `pandas`):

```
has_sold = test["competitor_lowest_price"] > test["proposed_price"]
sold_policies = test[has_sold]
# Minimize average loss on sold policies
avg_loss = (sold_policies["competitor_lowest_price"] -
            sold_policies["proposed_price"]).mean()
# Reach at least 30% market share
market_share = has_sold.mean()
print(avg_loss, market_share)
```

The output you have to deliver **MUST** follow these standards, otherwise the evaluation of the assignment will be invalid:

- **prediction file**: provide a csv/xlsx derived from test.xlsx with only 2 columns:
  - quote_id: you can find this column already in the file
  - proposed_price: column of float numbers with your proposal of prices for each quote_id
- **assignment code**: provide the code that generated the prediction file

Remember that the candidate will have to present the result of the assignment in a live session. The quality and the clarity of the code and of the oral exposition will be an important part of the overall evaluation. Any visual support (notebook, ppt, google slides, etc.) is allowed.

💡**Suggestion**:
Prioritize exploring how to minimize the average loss rather than spending too much focus on basic feature engineering and hyperparameter tuning.
As a reference, a straightforward ML model prediction can lead you to an average loss in the range of 55-60€. We expect you to tweak your pricing strategy to reach a better score than this baseline while maintaining the market share of at least 30%!

Data

The data provided contains the following information:

```
'driver_birth_date':
    Driver birth date.
'driver_driving_license_ym':
    Date (year-month format) when the driving license was obtained by the driver.
'driver_other_vehicles':
    Whether the driver owns other vehicles. If missing, the driver doesn't own any
other vehicle.
'driver_insured_years':
    Number of years the driver has been insured in the past.
'occasional_driver_birth_date':
    Occasional driver birth date. The occasional driver is optionally declared by
the driver.
'occasional_driver_license_attainment_age':
    Age at which the occasional driver obtained the driving license.
'policyholder_age':
    Policyholder age. The policyholder is declared when the driver and the
policyholder don't match.
'policyholder_license_attainment_age':
    Age at which the policyholder obtained the driving license.
'vehicle_acquisition_state':
    Whether the vehicle to be insured is currently owned by the policyholder, or
recently bought or will be bought in the future.
'vehicle_buy_ym':
```

```
    On which date (year-month format) the vehicle is bought.
'vehicle_registration_ym':
    In which date (year-month format) the vehicle has been registered.
'vehicle_engine_power':
    Vehicle engine power.
'vehicle_number_of_doors':
    Vehicle number of doors.
'vehicle_use':
    Whether the vehicle is used frequently or not, privately or for commercial
reasons.
'driver_claims_last_1_year':
    Number of claims occurred in the last year
'driver_claims_from_year_1_to_2':
    Number of claims occurred between 1 and 2 years
'driver_claims_from_year_2_to_3':
    Number of claims occurred between 2 and 3 years
'driver_claims_from_year_3_to_4':
    Number of claims occurred between 3 and 4 years
'driver_claims_from_year_4_to_5':
    Number of claims occurred before 4 years
'timestamp':
    Timestamp when the quotation was finalized.
'quote_id':
    Unique quote id.
'competitor_lowest_price':
    Target variable of the task. Minimum price offered by the competitors for a
quote.
'number_of_competitors':
    Number of competitors proposing a price for a quote. Not available in the test
set.
```