




Numerical Methods. Project

📅 Dates	@May 5, 2021 → May 9, 2021
☑ Finished	<input type="checkbox"/>
# Module	3
# Time spent, hours	
📌 Type	 Project

Project description

Rusty Bargain used car sales service is developing an app to attract new customers. In that app, you can quickly find out the market value of your car. You have access to historical data: technical specifications, trim versions, and prices. You need to build the model to determine the value.

Rusty Bargain is interested in:

- the quality of the prediction
- the speed of the prediction
- the time required for training

Project instructions

1. Download and look at the data.
2. Train different models with various hyperparameters (You should make at least two different models, but more is better. Remember, various implementations of gradient boosting don't count as different models.) The main point of this step is to compare gradient boosting methods with random forest, decision tree, and linear regression.

3. Analyze the speed and quality of the models.

Notes:

- Use the *RMSE* metric to evaluate the models.
- Linear regression is not very good for hyperparameter tuning, but it is perfect for doing a sanity check of other methods. If gradient boosting performs worse than linear regression, something definitely went wrong.
- On your own, work with the LightGBM library and use its tools to build gradient boosting models.
- Ideally, your project should include linear regression for a sanity check, a tree-based algorithm with hyperparameter tuning (preferably, random forest), LightGBM with hyperparameter tuning (try a couple of sets), and CatBoost and XGBoost with hyperparameter tuning (optional).
- Take note of the encoding of categorical features for simple algorithms. LightGBM and CatBoost have their implementation, but XGBoost requires OHE.
- You can use a special command to find the cell code runtime in Jupyter Notebook. Find that command.
- Since the training of a gradient boosting model can take a long time, change only a few model parameters.
- If Jupyter Notebook stops working, delete the excessive variables by using the `del` operator:

```
Copy codePYTHON
del features_train
```

Data description

The dataset is stored in file `/datasets/car_data.csv`. [download dataset](#).

Features

- *DateCrawled* — date profile was downloaded from the database
- *VehicleType* — vehicle body type

- *RegistrationYear* — vehicle registration year
- *Gearbox* — gearbox type
- *Power* — power (hp)
- *Model* — vehicle model
- *Mileage* — mileage (measured in km due to dataset's regional specifics)
- *RegistrationMonth* — vehicle registration month
- *FuelType* — fuel type
- *Brand* — vehicle brand
- *NotRepaired* — vehicle repaired or not
- *DateCreated* — date of profile creation
- *NumberOfPictures* — number of vehicle pictures
- *PostalCode* — postal code of profile owner (user)
- *LastSeen* — date of the last activity of the user

Target

Price — price (Euro)

Project evaluation

We've put together the evaluation criteria for the project. Read this carefully before moving on to the task.

Here's what the reviewers will look at when reviewing your project:

- Have you followed all the steps of the instructions?
- How did you prepare the data?
- What models and hyperparameters have you considered?
- Have you managed to avoid code duplication?
- What are your findings?
- Have you kept to the project structure?

- Have you kept the code neat?

You have your takeaway sheets and chapter summaries, so you are ready to proceed to the project.

Good luck!

Let's go!