Task 2: Predictive model

The model used for this task was a Light GBM model. This is a strong tree ensemble based model that has very high performance with tabular data. This model is highly optimized, uses the scickit learn API for easier integration with deployment standards. Its training is parallelizable making it efficient and scalable to higher volumes of data and also is very fast at inference.

Feature engineering:

From the EDA done for task 1, I selected for the first iteration a specific small set of variables that were highly descriptive of the dependent variable. The first experiment included the following features:

Numerical features**:** reservation_lead_time, trip_duration.

Categorical features**:** car_parking_address_postcode, weekday, month, day_nr, weekend, 7_to_12hs.

The initial performance was below expectations giving a RMSE on test data of 40 RMSE. To improve the quality of the information, the next step was to include information on the availability of the fleet, suspecting that the set of rules had this aspect into consideration. In other words, if we have low capacity available, the prices for new reservations in that specific area should be higher.

The new set of variables tried to give past utilization information to the model considering that there is seasonal behavior in the demand of the service as shown in the EDA. This means that if we can model how the demand works for each zip code, segmented by month, we should probably improve the quality of our predictions. Again, this is based on a supposition that the set of rules consider availability as an input for its dynamic pricing strategy.

The new variables combine a long memory information (hist_* features) with short memory (prev_* features) information. Long memory features describe the historical usage of the fleet at that zipcode at that hour of the week. Short memory features describe the previous hour usage of the fleet showing more flexibility for latest changes in demand. This short memory information will be more demanding on the model pipeline as very fresh information will have to be sent to the inference pipeline about the latest hour of utilization and journeys, so the positive impact on model performance has to be high enough to include this kind of information features.

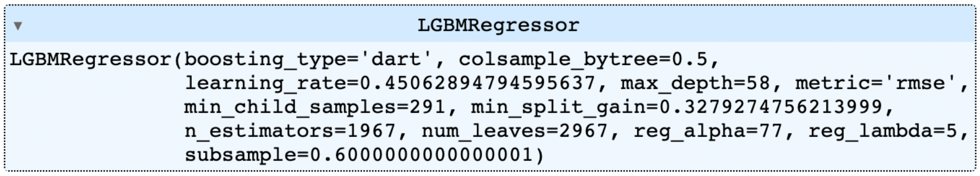
2nd iteration**:**
hist_usage_q10, hist_usage_median, prev_hour_usage_q10, Prev_hour_usage_median, car_id_hash. The median is useful to avoid impact of outliers and describes the general use of the fleet. As we also want to know if there is specific free capacity available we also get the 10th quantile as the low end of the distribution describing some cars free that are ready to be used. Also we encoded with catboost encoder the car hash id supposing that cars have different price categories. This is a strong encoding strategy for high cardinality categorical features that cannot be encoded with traditional encoding techniques like one hot encoding.

3rd iteration:
Included the number of cars with zero utilization (hist_free_cars and prev_free_cars) and removed short memory variables (Prev_hour_usage_median, prev_hour_usage_q10, prev_free_cars) as no increase in performance was detected.

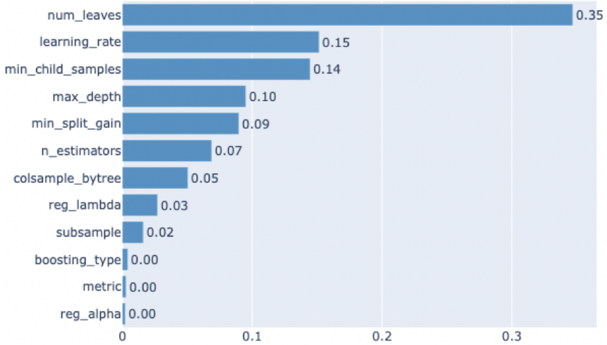
Model optimization:

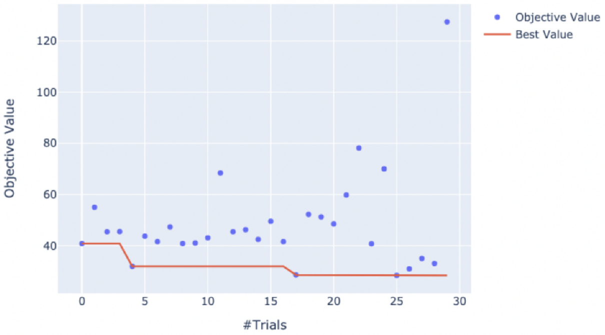
The method used to find the best set of hyperparameters selected was cross validation with 5 folds and Optuna. The cost metric to minimize at optimization selected where RMSE and MAPE, the last is more resilient to outliers than RMSE). The latest experiment consisted in 30 trials with 15 start up trials (random exploration of the surrogate cost function) running for approximately 2 hours.

Final Model Results: experiment “2022-06-15_150931”



```
▾                              LGBMRegressor
LGBMRegressor(boosting_type='dart', colsample_bytree=0.5,
              learning_rate=0.45062894794595637, max_depth=58, metric='rmse',
              min_child_samples=291, min_split_gain=0.3279274756213999,
              n_estimators=1967, num_leaves=2967, reg_alpha=77, reg_lambda=5,
              subsample=0.6000000000000001)
```





Performance metrics - 10% test set model “2022-06-15_150931”:
R2: score 0.9287 - RMSE: 14.39

Feature importance: (fit on full data)

22347	hist_free_cars
21379	reservation_lead_time
20272	car_parking_address_postcode
19154	weekend
17928	7_to_12hs
17302	car_id_hash
13732	trip_duration
2218	hist_usage_median
17	hist_usage_q10

Conclusions and next steps:

The model performance is good and according to expectations. More hyperparámeter optimization runs could result in significant gains as 70 trials is recommended, now for testing purposes we stopped at 30. External data could greatly improve performance depending on how the current rules works. If the rules consider weather ,marketing campaigns (discount codes for example) or competition in the área, this information would be relevant to better mimic these rules.