**Cars Prices Forecasting**

**Motivation**

Understanding of the principles of cars prices formation is important for customers, dealers and intermediaries. If you are buying a brand new car from an authorized dealer, you face sticky prices. However, if you want to buy a used car from an individual or unauthorized dealer, prices can vary sharply. If you want to sell your car, you probably do not know the exact price of your vehicle on aftermarket. Car price obviously depends on broad variety of different parameters and it is very hard to keep all in mind if you are not a car reseller and selling cars is not your business. Ability to forecast the car price based on its characteristics could save a lot of money for customers and increase profits of dealers and intermediaries.

In addition, I have personal motivation in this project because I plan to buy a car in the nearest future.

**Project stages**

1. **Data searching.** First of all, I limited variety of cars for analysis to most popular D-segment cars in Russia[[1]](#footnote-1). <https://moscow.auto.ru/> – one of the most visited automotive sites in the Russian internet and it was selected as data source for analysis. This website has most number of cars selling advertisements of new and used cars. One should remember that this would be supply prices. Without loss of generality we can make an assumption that supply price exceeds actual price at a certain percentage because it is very common to have 5%-10% discount from car seller (used car usually has a number of shortcomings). All materials for this project can be found at <https://github.com/BiXiC/auto_ru>
2. **Data collection.** To scrap webpages with cars advertisements, I used Python 3 and BeautifulSoup library and wrote the script [*AutoRu\_parser.ipynb*](http://localhost:8889/notebooks/AutoRu_parser.ipynb) that parses all advertisements from Moscow and +200 km area around it and saves data to pickle files (auto\_ru/data/D\_class).
3. **Data preparation and features engineering.** For preparation of features matrix, I wrote *features\_calculation.ipynb*that cleans data, fills missing values, extracts some useful features from scrapped data and saves features matrix (auto\_ru/processed data/)
4. **Exploratory analysis.** You can find in *Explonatory\_analysis.ipynb*
5. **Linear regression forecast.** To get forecasting benchmarks I fit two simple linear models (OLS) (*Linear\_model.ipynb*) based on results of exploratory analysis. First mode uses short list of basic car features. Second has a bit more extended list. I used MAPE to estimate error. On test data (20% of data, stratified to car name + generation variable) first model got MAPE 10.6% and second – 9.91%. This is not bad result for simple linear regression and can be obtained very fast.
6. **ML forecast.**
7. *Gradient\_boosting.ipynb* script contains Gradient Boosting Regressor model on same train/test split got MAPE 9.14% on test data. (3000 trees, max\_depth = 3)
8. *XGboost.ipynb*

Obviously cross validation should be done for more precise estimation of out of sample error on validation set but due to the lack of time it was left for future analysis.

1. **Further development.** On the next step of this project accurate and neat ML model should be developed. Cars models list should be extended and to all automotive segments. Other car-selling platform can be added. After that advisory website can be created and it can work as a service for customers for predicting optimal price for their vehicle. Another option for this model is integration into automotive website.

1. Short list of models for analysis: Kia Optima, Honda Accord, Hyundai i40, Hyundai Sonata, Toyota Camry, Mazda 6, BMW 5er, Audi A6, Ford Mondeo, Infiniti G35, Nissan Teana, Opel Insignia, Volvo S60 [↑](#footnote-ref-1)