Used Car Prices

The project is an attempt to automate process of estimation of used car price based on set of its features. It is supposed to be used by a company that specialized in selling used cars.

I was dealing with a regression model for price prediction. Mean squared error (MSE) and coefficient of determination (R2 score) were used as the performance metrics. The dataset was provided by SuperDataScience platform and was available on GitHub. It contained features such as manufacturer name, year when car was produced, odometer state, capacity of the engine, etc.

As part of the exploratory data analysis phase, I removed extreme outliers using bivariate analysis and scatter plots. Duplicates were considered as real world cases and were used for modeling. Also I checked liner regression assumptions, and decided to use nonlinear regression models because only few features had strong linear association with target variable. Categorical features were one hot encoded. Numerical features were transformed using PowerTransformer, QuantileTransformer, and scaled using StandardScaler. Those transformations were implemented as steps of a preprocessing pipeline. All the analyses were performing using scikit-learn libraries.

For the baseline model, I chose Linear Regression model with features, which were strong linear associated with price. The model had MSE error equal to 0.26 and R2 score of 73%. According to QQ plot, residuals were not normally distributed, so I preceded with nonlinear regression models.

For the model selection, I checked few algorithms such as SVR, GradientBoostingRegressor, and RandomForestRegressor. SVR gave the best cross-validation scores among them. Then I performed hyperparameter tuning using GridSearch CV and was able to achieve an MSE error equal to 0.13 and R2 score of 91% on the test set.

I used Python's pickle module to serialize the object of the final model. The trained machine learning algorithm was saved into a file, and was loaded and used for prediction.

Some challenges were skewed target variable with many outliers which reflected real spread of car price and should be considered in modeling.

I tried to reduce number of outliers based on car price but from some point that didn’t help much in improving performance. In addition, the data set was quite small it could lead to underfeeding. However, feature engineering and feature reduction allowed to improve performance metrics.

Moreover, an implemented pipeline with customer feature transformer allowed scaling the model to new data and features.

Data source: https://github.com/edis/sds_challenges/tree/master/challenge_2/data.