Report - 02.01.2018

The challenge provided by Haensel-AMS comprised a spreadsheet with 10,000 items and 8 features, one of which was the target (price). The task was to create a model to accurately predict the price of items previously not seen by the model.

Abstract

A model was created using 75% of the supplied dataset for training purposes. The evaluation on the remaining data resulted in an accuracy of 97% explaining approximately 67% of the observed variance (R^2-score – 0.6746). The model seems to generalize well but could still be improved if desired.

Categorical features

The regression problem was tackled with data cleaning as the first step. Three of the features were identified as categorical (loc1&2 and dow). Feature loc2 contained two digits/letters where the former was identical to the value of loc1 for the corresponding row. It was decided to reduce ambiguity by removing the first character of the loc2 feature.

Most of the rows contained numerical values, however few were letters – These could be errors/wrongly labeled data –2 digit encoded categories are quite restricting in current commerce settings, hence it was decided to keep the rows in the dataset and finally use one hot encoding to enable the model to work with letters.

One feature was identified as day of the week (dow) which was only one hot encoded similar to loc1/2.

These operations yielded a total of 34 columns plus the numeral / target features.

Numerical features

Concerning these features a variety of transformations was applied (x/1, x**2, x**3, sqrt(x), log10(x)) and compared to the non-treated distribution. The goal was to identify transformations that eased the distribution into something closely resembling a Gaussian distribution, making it easier for the regression algorithm to model the target dependency of the individual features.

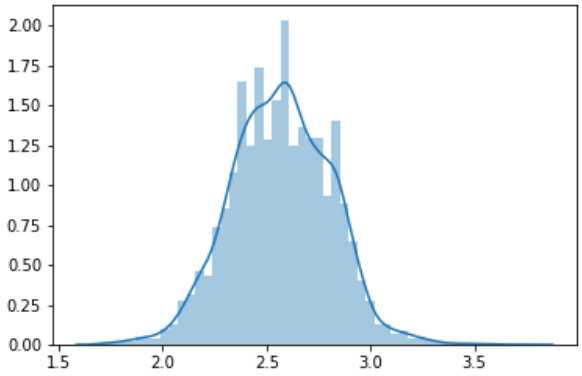

Figure 1: : log10 transformation of the target column (price)

Ultimately, only one transformation was found to do exactly what was expected. The application of the decadic logarithm to the target column resulted in a distribution with a mean of ~2.56 and a standard deviation of approximately 0.24 (Figure 1). As a result, the model will predict the magnitude of the price rather than the actual price, which might not be always desirable but works incredibly well in the current scenario.

Modeling

Three models were chosen for analysis in the challenge – LinearRegression (LR), Ridge and cross gradient boosting (XGB). While both the LR as well as Ridge provided similar prediction accuracy, use of the XGB-regressor increased the **testing accuracy** to about **97%** (Figure 2).

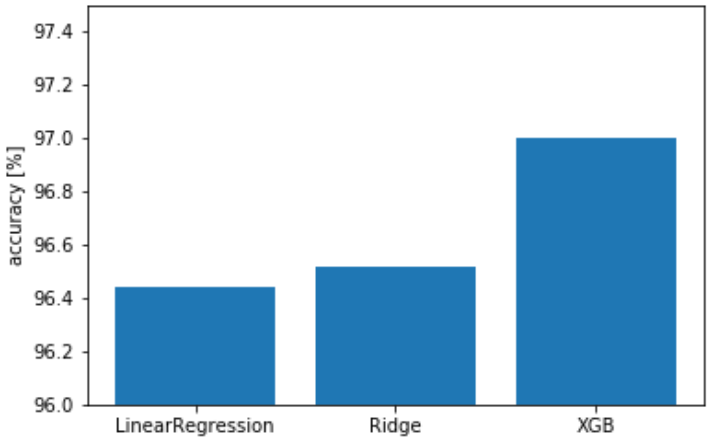

Figure 2: Accuracy of tested models in percent.
XGB – cross gradient boosting regressor

Outlook

Further enhancement could be provided by use of additional hyperparameters as well as their respective tuning. If that does not yield satisfactory results – the rather new lightGBM (Microsoft) might be sufficient for further experiments. However, to be noted -

it is recommended to only use the algorithm with more than 10,000 observations. The use of Kfold and shuffle split cross validation might also have considerable impact.
Furthermore, the loc1/2 and para1 features contain low frequency values which could be grouped depending on the column, but these values could also be excluded from analysis if desirable.
Concerning the numerical features scaling the values using one of scikit's many scalers could improve the model accuracy. The correlation matrix (0.55 para2-price), as well as the feature analysis (Figure 3) provide evidence for a strong correlation of para2 and price. The experimental investigation of the interaction could potentially lead to more insight as well.

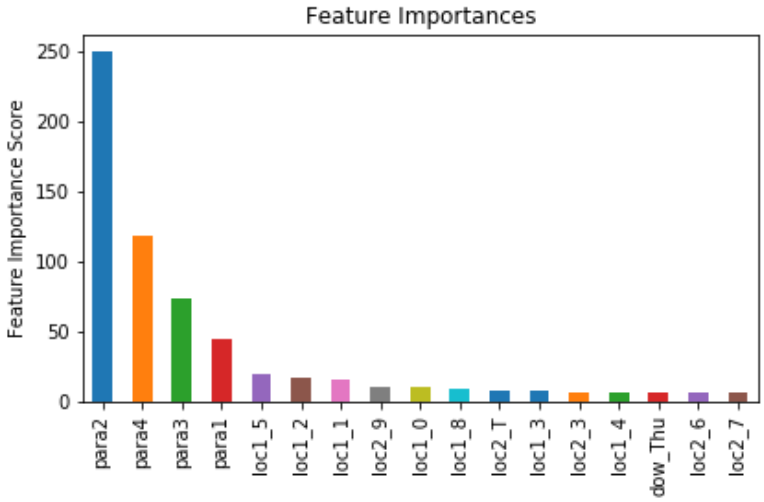

Figure 3: Excerpt of feature importance for the XGB-regressor prediction