Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

WYDZIAŁ MATEMATYKI STOSOWANEJ

Statistical Learning in Practice

Final Project

Baltazar Augustynek, Mateusz Mglej

Kraków, 19.01.2025

1. Dataset and Problem Statement

The *100,000 UK Used Car Dataset* is a comprehensive collection of information about the used car market in the United Kingdom. It contains 100,000 records covering vehicles from nine different manufacturers. The data was collected for the purpose of market analysis and includes the following features:

Feature	**Description**
model	Vehicle model name, e.g., Octavia, Fabia
year	Year the vehicle was manufactured
price	Vehicle price in British pounds
transmission	Type of gearbox, e.g., manual, automatic
mileage	Total mileage of the vehicle in miles
fuelType	Type of fuel used by the vehicle, e.g., petrol, diesel
tax	Annual road tax for the vehicle
mpg (miles per gallon)	Vehicle efficiency expressed in miles per gallon
engineSize	Engine displacement in liters

The dataset enables analysis of the used car market and key factors influencing vehicle value. It is available for download at the following address: Kaggle link. We will focus on a subset of the dataset that includes Škoda brand vehicles.

Problem Statement

The aim of this analysis is to develop a statistical model that allows for accurate estimation of the price of a used Škoda vehicle based on its available attributes. From a business perspective, such a model can bring tangible benefits to:

- **buyers**: by enabling assessment of a fair market price for a vehicle,
- **sellers and dealers**: by supporting the vehicle valuation process,
- **market researchers**: by assisting in the analysis of trends and consumer preferences.

By applying machine learning techniques and statistical analysis, it will be possible to build a precise tool for pricing Škoda vehicles in the UK market.

2. Data Exploration and Preparation

The process of data exploration and preparation involves several steps aimed at cleaning the dataset and eliminating potential issues that could impact the quality of analysis and modeling. The following actions were taken:

Removing Observations with Engine Size Equal to 0

In the dataset, we observe records where `engineSize` is equal to `0`. Initially, it may seem that such cases refer to electric vehicles, but analysis of the `fuelType` variable does not confirm this hypothesis. Therefore, all observations with `engineSize` equal to `0` are removed from the dataset.

Identifying Incorrect Prices

During price analysis, we notice a single observation with an extremely high value: `91874`. This value significantly deviates from the rest and most likely results from a data entry error. It is therefore removed.

Creating a Vehicle Age Variable

To better understand the impact of vehicle age on its price, we create a new variable `age`, which represents the vehicle's age in years since 2020. The calculation is performed as follows: $\texttt{age} = 2020 - \texttt{year}$.

Preparing Data for Model Building

The next step is to prepare the data for building predictive models. To do this:

- We remove white spaces from text values in the `model`, `transmission`, and `fuelType` columns to avoid errors during further analysis.
- We create binary (dummy) variables for categorical features:
  - Vehicle models such as `Octavia`, `Fabia`, `Karoq`, etc.
  - Transmission types: `Manual`, `Automatic`, `Semi-Auto`, `Other`.
  - Fuel types: `Petrol`, `Diesel`, `Hybrid`, `Other`.
- We also create binary variables for engine sizes such as `2.0`, `1.6`, `1.0`, etc.
- We remove columns that are not relevant for further analysis (categorical labels).
- We randomly split the dataset into a training set (75%) and a test set (25%) to evaluate the accuracy of predictive models.

3. Linear Regression

The goal of this stage is to construct and compare various linear regression models. The models will be evaluated using 10-fold cross-validation (10-CV), bootstrap analysis, variable subset selection methods, and performance on the test set.

Building Regression Models Using the `glm()` Function

We define the following six models:

Model	**Description**
Model 1	Intercept-only model (no predictors)
Model 2	Price as a function of age
Model 3	Price as a function of age and mileage
Model 4	Price as a function of all available variables
Model 5	Result of backward selection
Model 6	Result of forward selection

Each model is evaluated using 10-CV.

Model	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
MSE	39,680,045	24,805,003	24,652,933	3,116,636	3,130,760	3,113,791

Models 4–6 yield similar results, but we choose Model 6 (forward regression), as it uses fewer variables (25 compared to 34 in the full model).

Evaluation of the Selected Model on the Test Set

The selected forward selection model is tested on the test set, yielding the following results:

- Mean Squared Error (MSE): 2,778,571
- Mean Absolute Error (MAE): 1,250.03

Bootstrap Analysis

To assess the stability of the selected model, we apply bootstrap analysis with 1000 iterations. The classical coefficient estimates are compared with those obtained via bootstrap. Example differences between classical and bootstrap estimators are small, e.g., for age:

$$\textit{age}\text{: model } 6 = -983.5, \quad \text{bootstrap} = -981.7, \quad \text{difference} = 1.8.$$

The stability of the results suggests that the selected model is robust to sample variation.

Variable Subset Selection

To identify key variables, we use the variable subset selection algorithm (`regsubsets`). The algorithm evaluates models of increasing size based on several criteria, including R^2, adjusted R^2, Mallows' C_p, and the Bayesian Information Criterion (BIC). For our two selected models, we choose variables based on BIC and 10-fold cross-validation.

Model with the Lowest BIC

The lowest BIC is achieved for the model with 20 variables. Test set results:

$$\text{MSE}_{\text{SUB BIC}}: \quad 4{,}318{,}514 \qquad \text{MAE}_{\text{SUB BIC}}: \quad 1{,}492.92$$

Model with Best 10-CV Performance

The model with 27 variables achieves the lowest error during 10-fold cross-validation. On the test set, this model yields the following results:

$$\text{MSE}_{\text{SUB CV}}: \quad 2{,}783{,}338 \qquad \text{MAE}_{\text{SUB CV}}: \quad 1{,}252.26$$

4. Ridge and LASSO Regression

The aim of this section is to apply two regularization methods for linear regression: ridge regression and LASSO (Least Absolute Shrinkage and Selection Operator). We use both methods to improve model performance and reduce the problem of overfitting.

Ridge Regression

Ridge regression is performed using the `glmnet()` function (with parameter $\alpha = 0$, which corresponds to ridge regression). The optimal regularization parameter λ is selected using 10-fold cross-validation, resulting in: $\lambda_{\text{best}} = 385.91$. The model based on λ_{best} is evaluated on the test set, yielding the following results:

$$\text{MSE}_{\text{RIDGE}}: \quad 2{,}862{,}187 \qquad \text{MAE}_{\text{RIDGE}}: \quad 1{,}254.38$$

We observe that all variables remain in the model, with their coefficients simply being shrunk.

LASSO Regression

LASSO regression is also performed using the `glmnet()` function (this time with parameter $\alpha = 1$). The optimal λ value is: $\lambda_{\text{best}} = 6.90$. The constructed model is tested as follows:

LASSO regression shrunk 4 coefficients to zero, effectively eliminating less important variables, confirming its ability for feature selection.

$$\text{MSE}_{\text{LASSO}}:\quad 2{,}778{,}904 \qquad \text{MAE}_{\text{LASSO}}:\quad 1{,}250.13$$

5. Dimensionality Reduction Methods

Principal Component Regression (PCR)

As part of the principal component regression (PCR) analysis, the model is fitted using 10-fold cross-validation (10-CV) for various numbers of principal components. The results indicate that the best fit is achieved with 28 principal components, with test errors as follows:

$$\text{MSE}_{\text{PCR}}:\quad 2{,}776{,}254 \qquad \text{MAE}_{\text{PCR}}:\quad 1{,}251.05$$

The model with 28 components explains 92.25% of the price variance, which indicates high model quality. Further increasing the number of components results in a sharp increase in error, suggesting that more components may lead to model overfitting.

Partial Least Squares Regression (PLS)

In the partial least squares (PLS) regression analysis, the model is fitted using 10-fold cross-validation (10-CV) for various numbers of components. The optimal fit is achieved with 22 components, where the test errors are as follows:

$$\text{MSE}_{\text{PLS}}:\quad 2{,}769{,}480 \qquad \text{MAE}_{\text{PLS}}:\quad 1{,}249.77$$

The PLS model with 22 components explains 92.26% of the price variance, suggesting good model fit. Furthermore, increasing the number of components beyond this point does not lead to a significant improvement in model performance.

6. Tree-Based Methods

Regression Trees

The first model involves regression using a decision tree. The regression tree is fitted to the data using default parameters. Test results are as follows:

$$\text{MSE}_{\text{TREE}}: \quad 8{,}903{,}689 \qquad \text{MAE}_{\text{TREE}}: \quad 2{,}317.99$$

The constructed tree has 10 leaves and uses 6 variables. Cross-validation is performed to evaluate potential tree pruning, returning an optimal number of leaves: 10.

Bagging

The model based on *bagging*, i.e., an ensemble of trees, is built using the `randomForest` function with the default number of trees (500). The model yields the following results:

$$\text{MSE}_{\text{BAGGING}}: \quad 1{,}614{,}615 \qquad \text{MAE}_{\text{BAGGING}}: \quad 904.37$$

Testing the model with different values of *ntree* shows that increasing or decreasing the number of trees does not significantly improve performance, so 500 trees remain the optimal choice.

Random Forest

Next, we build a random forest model with the appropriate `mtry` parameter set to $\frac{p}{3}$, where p is the number of predictor variables. This model achieves the best test performance compared to the previous regression techniques:

$$\text{MSE}_{\text{RAND FOREST}}: \quad 1{,}537{,}323 \qquad \text{MAE}_{\text{RAND FOREST}}: \quad 897.14$$

Boosting

We apply *boosting* for regression trees using the `gbm` function. Initially, we use default parameters, obtaining an MSE of 1,710,982 and MAE of 934.54.

Next, we tune the model by searching for the best hyperparameters. The selected values are: *number of trees* = 1700, *depth* = 3, and *learning rate* = 0.1. The model based on these settings gives the following test results:

$$\text{MSE}_{\text{BOOST}}: \quad 1{,}586{,}667 \qquad \text{MAE}_{\text{BOOST}}: \quad 924.81$$

$$\text{MSE}_{\text{BART}}: \quad 1{,}656{,}678 \qquad \text{MAE}_{\text{BART}}: \quad 933.44$$

Bayesian Additive Regression Trees (BART)

After training the BART model with default parameters on the training data, we perform prediction on the test set. The results are as follows:

The BART model allows for assessing how often specific variables were used in the constructed trees. The most frequently used variables in this model are:

- **mpg** (35.90)
- **age** (18.56)
- **mileage** (14.16)
- **model_superb** (13.99)
- **model_karoq** (13.41)

These values indicate which features had the greatest impact on the model's predictions.

XGBoost

To achieve the best possible results, we optimize hyperparameters, in particular the `max.depth` parameter, which determines the maximum tree depth in the model. After analyzing various models for this parameter, the best value turned out to be 7. In addition, the optimized XGBoost model is sufficiently trained after 25 iterations ($\text{RMSE}_{\text{best}} = 1,274.59$).

After building and training the XGBoost model (with parameters *max.depth* = 7, *nrounds* = 25), the following results were obtained on the test set:

$$\text{MSE}_{\text{XGBoost}}: \quad 1{,}624{,}576 \qquad \text{MAE}_{\text{XGBoost}}: \quad 926.67$$

7. Summary

We present the results of a comparison of various regression models used to predict the price of a Škoda vehicle based on its features. The aim of the analysis was to select the best model based on the Mean Squared Error (MSE) and Mean Absolute Error (MAE) values.

Model Results

After building all models based on the training data, MSE and MAE errors were calculated on the test set. The results are shown in the table below, which includes model names and their corresponding error values.

Model	MSE	MAE
Random Forest	1,537,323	897.14
Boosting	1,586,667	924.81
Bagging	1,614,615	904.37
XGBoost	1,624,576	926.67
BART	1,656,678	933.44
PLS	2,769,480	1,249.76
PRC	2,776,254	1,251.05
GLM Forward	2,778,571	1,250.03
Lasso	2,778,904	1,250.13
GLM Full Model	2,780,182	1,251.55
Best Subset: CV	2,783,338	1,252.26
Ridge	2,862,187	1,254.38
Best Subset: BIC	4,318,514	1,492.92
Tree	8,903,689	2,317.98

Random Forest **Boosting** **Bagging**
XGBoost **BART**

Commentary

From the table above, we see that the best-performing models are:

These models show minimal differences in performance; however, depending on computational complexity, one may be more suitable depending on analytical requirements. On the other hand, traditional regression models such as the **GLM Full Model**, variable selection methods (**Best Subset: BIC** and **Best Subset: CV**), and regularized regressions (**Ridge** and **Lasso**) achieved weaker results.

Conclusions

Based on the results obtained, the best-performing models are ensemble techniques such as **Random Forest**, **Boosting**, and **Bagging**, which demonstrated the highest predictive power for car prices. Although the performance differences between them are small, if there is a need to choose the most efficient model, it is worth considering the computational complexity and training time, especially for very large datasets.