UK Car Market: EDA & Prediction

1. Problem statement

The United Kingdom's automotive market has been experiencing fluctuations in car prices due to various factors such as economic conditions, government policies, and technological advancements. Accurate car price prediction is essential for buyers, sellers, manufacturers, and other stakeholders in the industry to make informed decisions. This study aims to develop a data-driven model that can accurately predict car prices in the UK automotive market, taking into account variables such as make, model, age, mileage, fuel type, transmission type, and other relevant factors.

2. Solution Approach

To address the problem of predicting car prices in the UK automotive market, we propose the following solution approach using Bayesian optimization in machine learning models:

a. Data Collection and Preprocessing.
b. Feature Engineering.
c. Model Selection.
d. Hyperparameter Tuning with Bayesian Optimization.
e. Model Training and Validation.
f. Model Evaluation

The complete project code can be accessed here.

3. Stakeholder and benefits

Several stakeholders would benefit from accurate and reliable predictions in the context of UK car price prediction. These stakeholders and their associated benefits include:

a. *Car Buyers*: Accurate car price predictions empower buyers to make informed decisions when purchasing a vehicle. They can evaluate whether a car is pretty priced, avoiding overpaying or falling victim to fraudulent deals.
b. *Car Sellers (Individuals and Dealerships)*: Sellers benefit from a data-driven price estimation, ensuring they list their vehicles at competitive market prices. This fact helps them attract potential buyers and facilitates quicker sales transactions.
c. *Car Manufacturers*: Accurate predictions also enable them to optimize inventory levels and better anticipate future demands.

4. Data

The data was obtained from Kaggle. The data is divided into a train and test dataset, with 19237 and 8245 instances, respectively. The features include Levy, Manufacturer, Model, Prod. year, Category, Leather interior, Fuel type, Engine volume, Mileage, Cylinders, Gear box type, Drive wheels, Doors, Wheel, Color and Airbags.

5. Tools

Data processing, Pipeline, Bayesian Optimization, Python, Data Visualization, Scikit-learn, XGB regressor, Stacking model.

6. Results

6.1. Data Processing

Only Price, Prod. year, Cylinders, and Airbags are numerical values in the raw data. No missing values were found according to the check function. The NA values of price and Levy were replaced by 0. The feature Doors showed an inconsistent data format, resulting the in the following value counts: “04-May”: 14855, “02-Mar”: 746, and >5:124. The corresponding values were corrected and transformed into numeric values (2,4, or 5 doors). Engine and Mileage feature values were transformed into float and integer data types, previous string format correction (split and strim techniques).

6.2. Cardinality reduction

Cardinality reduction is a process where the number of unique categories (cardinality) in a categorical feature is reduced. High cardinality in categorical variables can lead to increased computational complexity, memory usage, and longer training times. By reducing the number of categories, we can simplify models, improve generalization, and reduce overfitting.

Fig 1 shows the number of unique values for different features. Manufacturer (67) and model (> 1500) show high cardinality. Only the top 15 values remained for all categorical features, and the others were grouped into “Other” values.

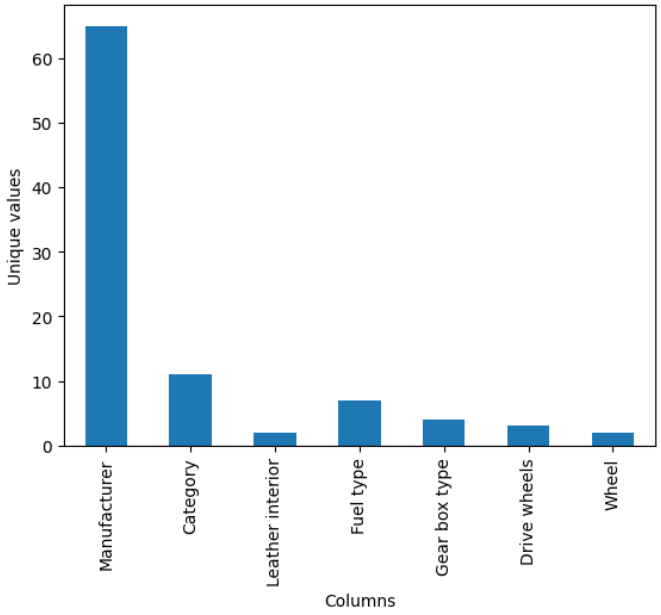

Fig 1. Unique values for multiple features.

6.3. Encoding and quantile processing

Label Encoder and Quantile Processing are two preprocessing techniques used in machine learning and data analysis. Label encoding is a methodology used to convert categorical variables into integer values. LabelEncoder module was employed for this purpose.

Quantile processing, also known as quantile transformation, is a technique used to transform the distribution of a continuous variable into a uniform or normal distribution. The main idea behind this transformation is to map the original data points to their corresponding quantiles in the target distribution. It can mitigate the impact of outliers,

as extreme values are mapped to the tails of the target distribution. QuantileTransformer was used for the transformation.

6.4. Feature engineering

Feature engineering is the process of transforming raw data into a format that is more suitable for machine learning algorithms. This process often involves creating or modifying new features to enhance the model's performance. Preprocessing is a crucial feature engineering step that helps clean, organize, and structure the data before it is fed into a machine learning algorithm.

One common preprocessing technique is scaling, which adjusts the range of feature values so that they are on a similar scale. This characteristic is essential because some machine learning algorithms are sensitive to the scale of input features and may perform poorly if the elements are on different scales. MinMaxScaler module was used. MinMaxScaler is a popular scaling method that typically transforms features by scaling them to a specific range [0, 1].

Once the previous transformations were applied, the mutual information scores were determined and plotted (Fig 2). Mutual Information (MI) measures the amount of information that knowing the value of one variable provides about another. It quantifies the "mutual dependence" between two variables. MI can capture both linear and non-linear relationships between variables. This makes MI more versatile and capable of identifying relationships that Pearson correlation would miss. MI is invariant to monotonic transformations of variables (e.g., logarithmic or exponential transformation). In addition, a heatmap using Pearson correlation is shown in Fig 3.

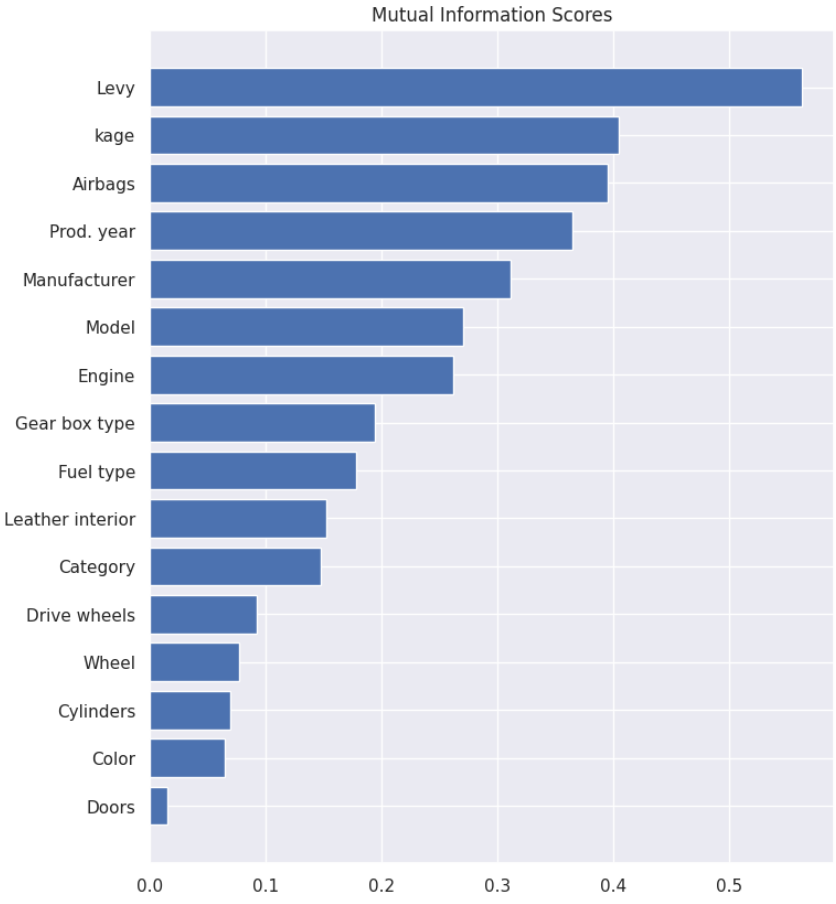

Fig 2. Mutual information scores with car price.

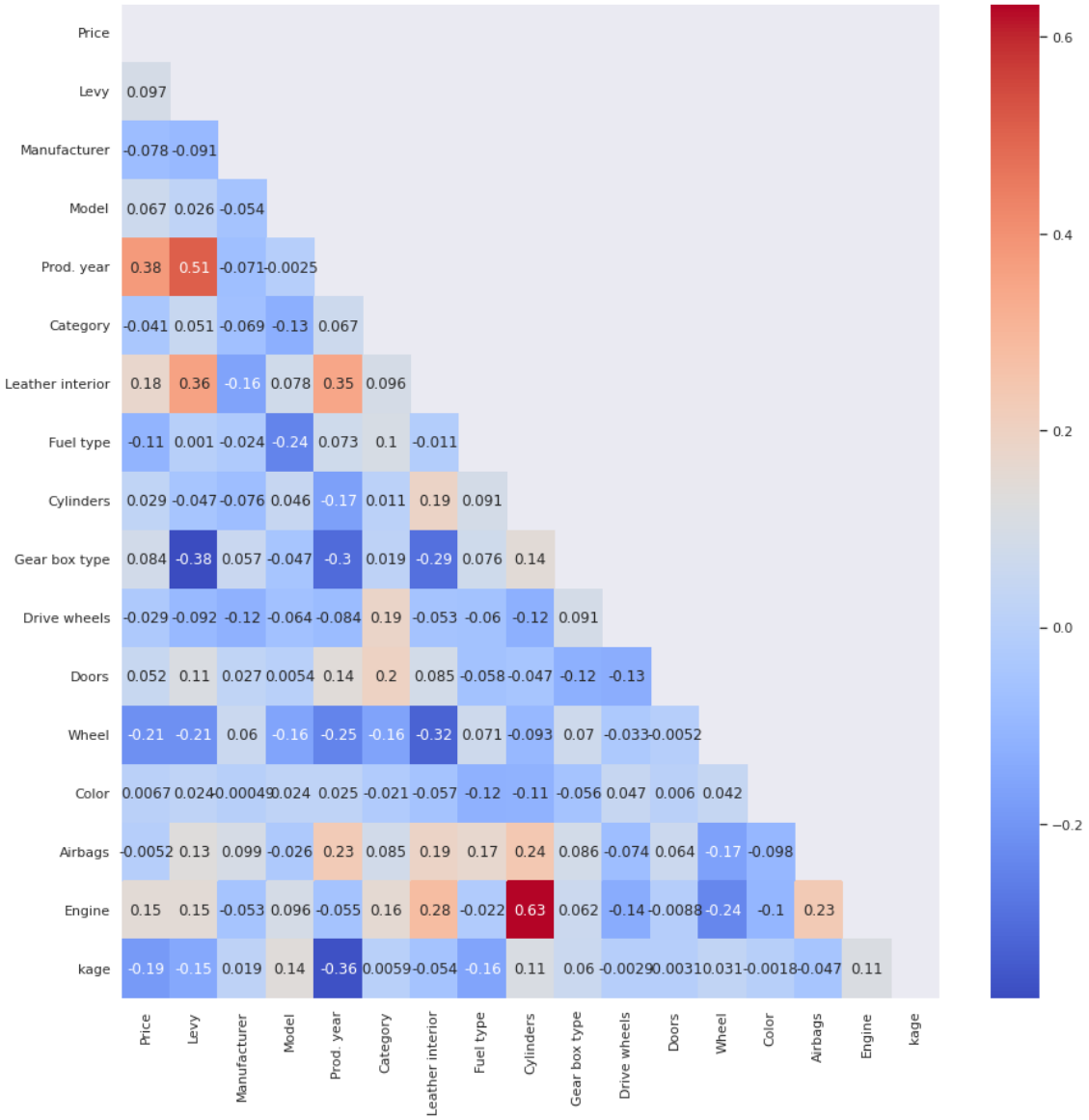

Fig 3. Heatmap of features (Pearson Correlation).

The top five features according to MI scores were Levy (0.56), Kage (0.46), Airbags (0.39), Prod. year (0.36), and Manufacturer (0.31). In contrast, the top 5 features according to Pearson correlation were Prod. Year (0.38), Wheel (-0.21), Kage (-0.19), Engine (0.15), and Fuel type (-0.11). As shown, the scores differ for each variable. Pearson's score was selected as the selector in this case according to the high proportion of numeric and continuous variables. The features with lower Pearson correlation coefficients were removed. Fig 4 shows the Pearson heatmap for the remained variables (Prod. year, wheel, Kage, Engine, Fuel type, and Leather interior).

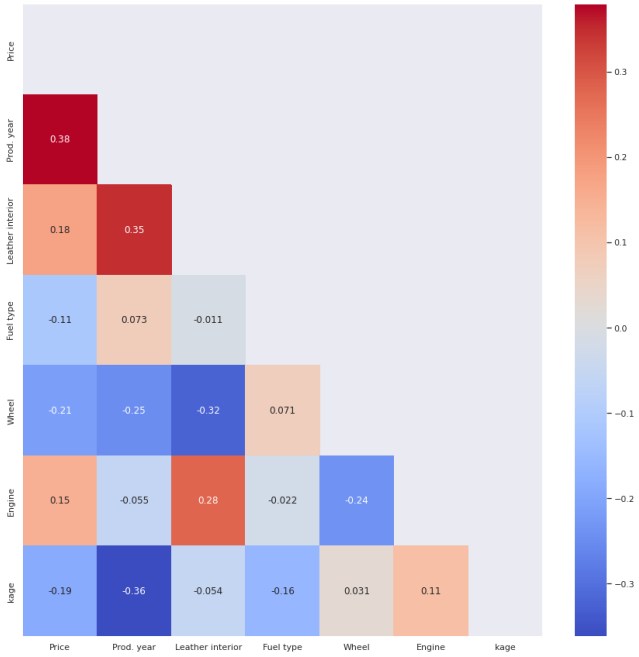

Fig 4. Heatmap after feature selection.

6.5. Data modeling

First, multiple machine-learning models were tested by evaluating their cross-validation scores without tuning. Then, the Bayesian optimization method was adopted to tune the top five models, producing the final result by averaging the result. K-Fold Cross-Validation is a resampling method used to evaluate the performance of a machine learning model. It helps to address the overfitting issue and provides a more reliable estimate of the model's performance on unseen data. In K-Fold Cross-Validation, the dataset is divided into 'K' equal-sized subsets (or folds), and the model is trained and tested K times, with each fold being used as the test set exactly once. Table 1 shows the metrics (R^2 and RMSE) obtained for the initial model testing.

Table 1. Initial model testing.

Model	R^2	RMSE
XGB Regressor	0.375	0.228
LGBM Regressor	0.427	0.218
Gradient Boosting Regressor	0.374	0.228
Ada Boost Regressor	0.248	0.249
K Neighbors Regressor	0.346	0.233
Random Forest Regressor	0.380	0.227
Bayesian Ridge	0.201	0.258
Ridge*	0.201	0.258
Lasso*	0.201	0.258
Elastic Net*	0.201	0.258
Support Vector Regression	0.333	0.235

*alpha=0.0001.

The top five models with the best metrics were: LGBM regressor, XGBRegressor, Gradient Boosting Regressor, KNeighborsRegressor and RandomForestRegressor

6.6. Bayesian optimization

Bayesian optimization was applied to multiple models. A particular grid search was defined for each model, considering their specific hyperparameters. Table 2 shows the tunned hyperparameters and the performance metrics obtained using cross-validation.

Table 2. Metrics and tunned hyperparameters using Bayesian optimization.

Model	Tunned hyperparameters	R^2	RMSE
Gradient Boosting Regressor	learning_rate=0.01, max_depth=8, max_features=0.1, n_estimators=1442	0.442	0.215
XGB Regressor	colsample_bylevel=0.325, colsample_bynode=0.1, colsample_bytree=1.0, gamma=4.071e-05, learning_rate=0.01, max_depth=15, min_child_weight=10, n_estimators=3000, objective='reg:squarederror', reg_alpha=0.0171, reg_lambda=0.004646, subsample=1.0	0.443	0.215
LGBM Regressor	colsample_bytree=0.988, learning_rate=0.01, max_bin=166, max_depth=9, min_data_in_bin=2, num_iterations=1082, num_leaves=198, subsample=0.7305	0.433	0.217
Elastic Net	alpha=0.000540, l1_ratio=0.001, max_iter=10000	0.201	0.258

After Bayesian optimization, XGB Regressor was the best model. Stacking Model was performed using XGB Regressor, Gradient Boosting Regressor, and LGBM Regressor. Stacking is an ensemble learning technique that uses multiple regression models to make a prediction and then uses another model, often referred to as a meta-learner or second-level learner, to make a final prediction based on the initial values. After stacking, the ensemble model with tunned hyperparameters was again trained using cross-validation. The best model shows the following metrics: R^2 = 0.449, RMSE = 0.214, and σ = 0.00166.

Feature importance was measured through the F-score, a method used to quantify the significance of individual predictors in a model. The F-score represents the ratio of the mean squared error of the model with the predictor to the mean squared error of the model without the predictor. A larger F-score indicates a more significant predictor. Fig 5 exhibits the feature importance for model prediction.

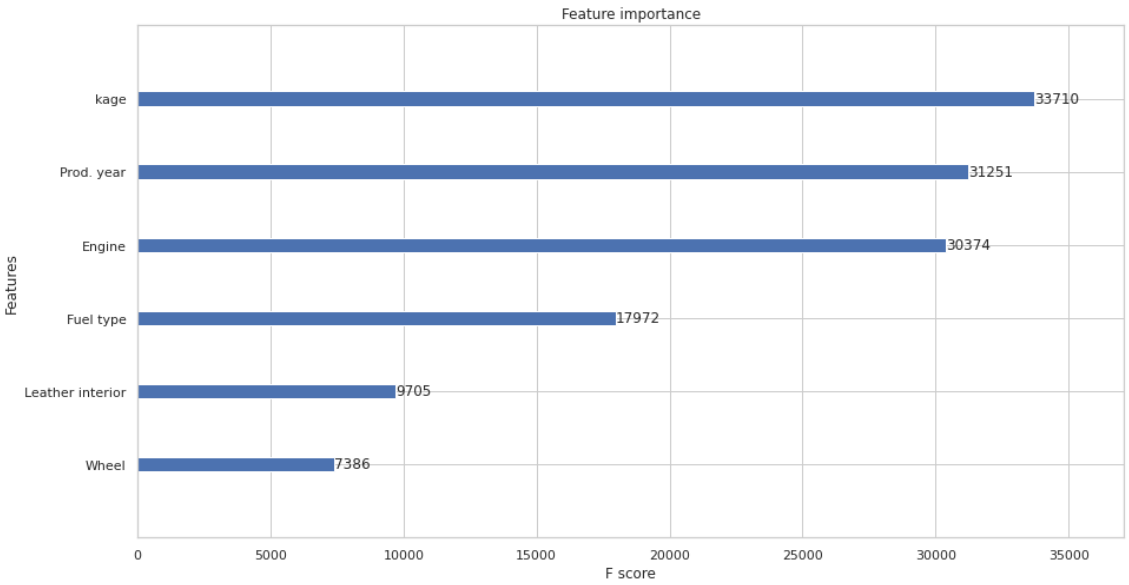

Fig 5. Feature importance in model prediction (F-score).

As shown in Fig 5, In the UK car market, the essential feature is the kilometer age of the vehicles, as well as their production year.

7. Conclusions

a. Kilometer age is the essential feature in the UK car market, followed by the year of production (F - score).
b. Multiple data processing methodologies were applied: Cardinality reduction, encoding, quantile processing, and min-max scaling.
c. Mutual information (MI) scores and Pearson correlation coefficients differ.
d. XGB Regressor shows the best metrics models after Bayesian optimization. The ensemble model was again trained, reaching the best indicators: R^2 = 0.449, RMSE= 0.214, and σ = 0.00166.

8. Further works

a. Use of MI scores as an indicator for feature engineering selection.
b. Incorporate extra variables into machine learning modeling to obtain better performance metrics.
c. Test other model regressors.