Machine Learning CS412 Term Project

Car Price Prediction

Group 11

İlker Gül	26352
Görkem Güzeler	26841
Oğuz Kağan Çakır	25328
Hakan Büyüktopçu	26338
Serra Yılmaz	26969

Google Colab Link:
https://colab.research.google.com/drive/18vD2St6YDztwcggZJoHv3S7xSXytO4rk?usp=sharing

Problem Definition

In this project, we try to understand how car prices change due to various variables such as the year of the car, the tax, the mileage, the fuel type, etc. Using different types of Machine Learning techniques, we try to detect the price of a car with given attributes in the dataset.

Data Exploration

After the training dataset is loaded, the types, statistics of the variables, and correlations among them are observed. Our training dataset consists of 60,000 observations and 12 variables. The target variable is the price of the car. The variables in the training dataset are as follows: ID, brand, model, year, transmission, mileage, fuel type, mpg, engine size, tax, tax in £, and price. Our test dataset consists of 25,555 observations and 11 variables.

The types of variables:

integer/float and quantitative variables: ID, year, mileage, mpg, engineSize, tax, tax (£), price

string and categorical variables: brand, model, transmission, fuelType

The counts of missing values in the training dataset in each variable:

ID: 0, brand: 66, model: 60, year: 59, transmission: 52, mileage: 66, fuelType: 70, mpg: 64, engineSize: 52, tax: 3372, tax (£): 56628, price: 0

The correlation values of the quantitative variables with price in decreasing order:

engineSize, year, tax, tax (£), mpg, mileage (mpg & mileage are negatively correlated)

The types of brands in the training dataset in decreasing order of their counts:

Ford, Volkswagen, Mercedes, Audi, Bmw, Toyota, Skoda, Hyundai

Preprocessing

In the preprocessing section, the missing numerical values are handled by replacing them with the mean of the related variable. The categorical variables are encoded into numerical forms in the end. Other than these general claims, each attribute is handled in a way that suits it the best, and many different features derived from existing ones are added. These changes & additions are as follows:[1]

- brand: Unknown brands are assigned based on the model of the cars, since a model can belong to only one brand.
- Year: Replaced by **age** (by subtracting it from the current year) to give better insight about the model. Then the attribute **age_class** that shows the age interval of the cars with values 1 to 4 is created.
- tax: tax (£) entries are converted to Dollar currency using a fixed rate of Sterling/Dollar (1.30) and inserted into the null parts of tax, then the tax (£) column is discarded.[2]
- taxorNot: This binary attribute was added to emphasize whether the car has any tax or not.
- mileage: Classified as unknown, low, average, high by given values 0 to 3. These results are then added as a new attribute called **mileage_class.**
- engineSize: Classified based on the size and given values 1 to 4. Then a new attribute called **engineClass** is added which infers to engine size**.**
- fuelType: A new binary attribute called **Eco-friendly** is created that maps if the car is eco-friendly or not. Electric and Hybrid fuel types are considered eco-friendly while other fuel types are not.

[1] Added attributes are shown with **bold characters.**
[2] Here, it was assumed that the tax column was in Dollar currency.

- mpg: Classified with values 1 to 3 that gives the interval of the mpg. Then a new attribute **mpg_class** is added using these results.
- transmission: Categorical attributes are realized by the values 1 to 4. 4 for automatic, 3 for semi-automatic, 2 for manual and 1 for other types.
- **popularity**: Another attribute is added which ranks models based on their mean in terms of price. For the models that are not in the training set, a number is assigned according to their brand. Also, if a brand is not known, then the average of all models is assigned.
- **segment**: This new attribute classifies the car brands with values 1 to 3 based on their quality perceived by the world.
- model: A specific number is assigned for each model and brand. For the cars of which models are not given, the average model of that specific brand is given.
- **life**: This new attribute demonstrates the possible remaining lifespan of the vehicle based on the mileage. It takes the values 1 to 4.

Model

In the model generating process, we have tried 4 models: Gradient Boosting Regressor, Random Forest Regressor, Linear Regression and Voting Regressor. We splitted our train data into 0.8 train set (48000 instances) and 0.2 validation set (12000 instances). We compared our results based on the Mean squared error and R2 squared.

Gradient Boosting:	MSE= 7940129.608157305	r2= 0.92
Random Forest Regressor:	MSE= 4891123.290037597	r2= 0.95
Linear Regression:	MSE= 21272980.533136427	r2= 0.78
Voting Regressor:	MSE= 7722369.627910673	r2= 0.92

Since Random Forest outperformed other models with the highest r2 score and lowest mean squared error, we used Random Forest regressor. We tested [100, 250, 400, 550] for n_estimators parameter, [2, 4, 6, 8, 10] for max_features parameter to pick the best hyper-parameters through the grid-search method. We decided to train our model with 400 estimators, and we set the max_features as 8. Lastly, the same preprocessing steps are applied to the test dataset, and the generated model is used to predict the prices of the cars in the test dataset. We obtained a 0.95 r2 score in the given public test dataset. Then, we wanted to train the model with more instances. Thus, we changed the train-test split to 0.98-0.02. After that, we received 0.95541 in the given public test set.

Appendix

	ID	year	mileage	mpg	engineSize	tax	tax(£)	price
ID	1.000000	-0.000191	-0.000517	0.000728	0.005202	-0.002048	0.021099	0.004285
year	-0.000191	1.000000	-0.744589	-0.137330	-0.035766	0.209365	0.241393	0.503388
mileage	-0.000517	-0.744589	1.000000	0.184649	0.105415	-0.226317	-0.188928	-0.430598
mpg	0.000728	-0.137330	0.184649	1.000000	-0.285945	-0.438297	-0.479663	-0.340773
engineSize	0.005202	-0.035766	0.105415	-0.285945	1.000000	0.315618	0.230743	0.638214
tax	-0.002048	0.209365	-0.226317	-0.438297	0.315618	1.000000	nan	0.349616
tax(£)	0.021099	0.241393	-0.188928	-0.479663	0.230743	nan	1.000000	0.316598
price	0.004285	0.503388	-0.430598	-0.340773	0.638214	0.349616	0.316598	1.000000

Fig 1. Correlation matrix of training dataset.

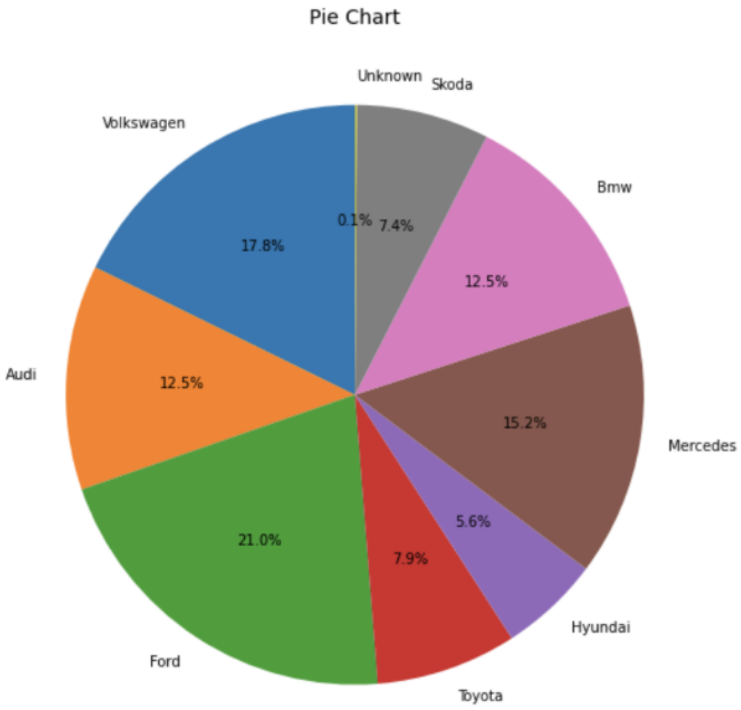

Fig 2. Pie chart of brand percentages.