Car Data Analysis and Price Prediction

Ömer Bera Dinç[1], Ece Yılmaz[1]
1. Control and Automation Engineering Department, Istanbul Technical University

***Abstract*— This study focuses on developing a machine learning model to predict the prices of new vehicle models to be launched based on a dataset comprising 25 different vehicle features and their corresponding prices. To achieve the most accurate model, various machine learning and deep learning algorithms were utilized. Initially, the dataset was transformed into a more meaningful dataset by applying feature engineering techniques. Using this refined dataset, linear regression, random forest, gradient boosting, and neural network models were trained, and their results were compared. Among these techniques, the random forest method demonstrated superior performance in terms of maximizing the R^2 score and the number of relevant features. This study is expected to help analyzing consumer demands in the automotive industry and improving pricing efficiency. The diversity of examples in the dataset enables the model to perform analyses in different markets globally.**

***Keywords*—machine learning, deep learning, feature engineering, car price prediction**

I. Introduction

Pricing is a science, relying on statistical and experimental models to establish a strong market profile for a brand and its products. Automotive industry is one of the world's largest industries by revenue, with pricing playing a pivotal role in its dynamics. Determining the right launch price for a new car model is crucial not only for gaining customer trust in the market but also for shaping the direction of the industry. (Mala & Sudhish, 2024, p.1-6)

Before entering the automotive market in a new country, conducting comprehensive market research specific to that country is a crucial step. Understanding the economic, cultural, and demographic characteristics of the target market is needed for developing a successful market entry strategy. This research helps companies to identify consumer demands and key factors affecting car selection, ensuring that their vehicle features and prices align with target audience expectations.

In developed countries, consumers prioritize technological innovations, brand reputation, safety features, and eco-friendly vehicles when selecting a car. The emphasis on sustainability and advanced features reflects the high income and environmental awareness common in these regions. Conversely, in developing countries, priorities tend to shift towards affordability, fuel efficiency, and durability. These criteria align with the economic realities and needs of consumers in such markets. Even within the same country, preferences can vary based on geographic and demographic factors. For instance, consumers living in urban areas may prefer hybrid or electric vehicles due to their energy efficiency and suitability for city driving. In contrast, consumers in rural areas often prefer SUVs or off-road vehicles, which are better equipped to handle rugged terrains and long-distance use. Cultural differences also play an important role in shaping car selection criteria. In countries like the United Arab Emirates, where show-off is common, there is a big demand for luxury sport cars.

To successfully enter a new market, automotive companies must introduce cars that meet the demand at competitive prices. Considering the production, logistics, and marketing costs involved, the critical role of this market analysis becomes even more evident. Our aim in this project is to analyze a market to determine how car features affects pricing and to create a predictive model that can estimate car prices based on their features for this market. To achieve this goal, a suitable dataset with 205 car samples available in the U.S. market has been selected. [2] The U.S. market offers an ideal setting for such a study, because it is one of the countries with the most diverse car owner base due to its multicultural population and various geographical and climatic conditions across its states. The chosen dataset contains 25 distinct features of cars, such as model, engine specifications, car body measurements, mileage, and corresponding prices.

At the end of this project, the key factors affecting car prices in the U.S. market will be identified. By using the predictive model developed during this study, it will be possible to predict a car's market price based on its features.

II. Literature Review

Numerous studies have been conducted to predict car prices; however, most of these works has focused on the price prediction for used cars. Predicting used car prices is a more complex problem due to the large number of features affecting these prices and the variability in selling prices for the same car models. Furthermore, inflated pricing of used cars often distorts market dynamics, creating additional challenges. Consequently, creating accurate predictive models for the used car market can address real-world issues and provide valuable solutions.

In a study by Monburinon et al. [3], models predicting car prices were developed using multiple linear regression, random forest regression, and gradient boosting regression methods. Among these all models, gradient boosting was selected as the best performing model, with a mean squared error (MSE) of 0.28.

Narayana et al. [4] studied the used car market in India, applying linear regression, decision tree, and random forest algorithms to create different machine learning models. Their findings revealed that the random forest model had the highest accuracy rate of 85%.

Similarly, Varshitha et al. [5] conducted research on a dataset from the Indian used car market, implementing deep neural network models with varying number of layers, linear regression, ridge regression, and random forest methods. The study concluded that the random forest model was the most accurate, with a mean absolute error of 0.746 and an R^2 error of 0.917.

In this study, aim is to predict the prices of newly launched cars. While the problem differs from the mentioned studies, the methods applied in those works will be a guide for our project.

III. Methodology

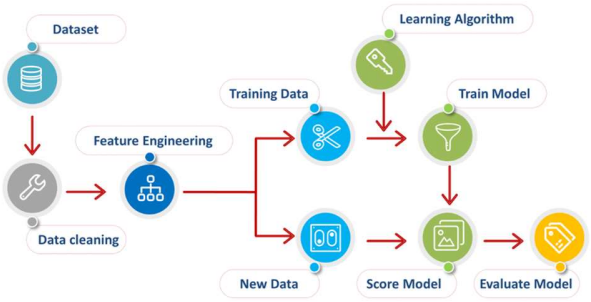

Fig. 1. Machine learning process diagram.

As it can be seen from Figure 1, dataset has been pre-processed. Cleaned data splitted into training and test sets to conduct training and evaluation phases.

A. Dataset

Kaggle is a big data science community with open-source resources, datasets and more. Dataset needed for car price prediction has been retrieved from Kaggle. [2] The dataset contains 26 different features of a car. These features can be seen in Table I.

While 16 features in the data set contain numerical data, the remaining 10 features contain categorical data. These features cover technical properties about engines, exterior body, and fuel consumption of cars.

B. Feature Analysis and Data Preprocessing

A systematic approach has been followed for the best feature selection process. At first, only numerical features taken into account and built a correlation matrix with price feature which means label for the task. As seen in Figure 2, there are 6 features whose correlation with price is below 0.12. We removed these 6 features from this data set: *carheight, stroke, compressionratio, symboling, peakrpm, car_ID.*

TABLE I
Data Features

Name	Description
car_ID	Unique id of each observation (Integer)
symboling	Assigned insurance risk rating, between +3, -3 (Integer)
CarName	Name of car company (Categorical)
fueltype	Car fuel type, gas or diesel (Categorical)
aspiration	Engine aspiration type (Categorical)
doornumber	Number of doors in the car (Categorical)
carbody	Body style, e.g., sedan (Categorical)
drivewheel	Type of drive wheel, e.g., 4wd (Categorical)
enginelocation	Location of car engine (Categorical)
wheelbase	Weelbase of car (Numeric)
carlength	Length of car (Numeric)
carwidth	Width of car (Numeric)
carheight	Height of car (Numeric)
curbweight	The weight of a car without occupants or baggage (Numeric)
enginetype	Type of engine e.g., dohc (Categorical)
cylindernumber	Cylinders placed in the car (Categorical)
enginesize	Engine size in cubic inch (Numeric)
fuelsystem	Type of fuel injection system (Categorical)
boreratio	Bore ratio of car engine (Numeric)
Stroke	The length piston travels in the cylinder (Numeric)
compressionratio	Compression ratio of engine (Numeric)
horsepower	A unit of measurement of power (Numeric)
peakrpm	Engine peak rpm (Numeric)
citympg	Distance that a car can travel in city using a particular amount of fuel (Numeric)
highwaympg	Distance that a car can travel in highway using a particular amount of fuel (Numeric)
price	Price of car (Numeric)

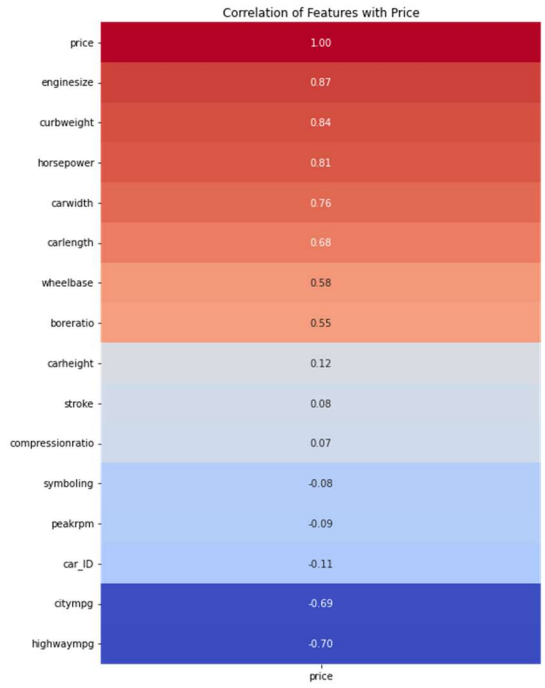

Fig. 2. Price correlation of numerical features.

Engine size, curb weight, horsepower, carwidth, carlength, wheelbase, boreratio were positively related with the price. However, citympg, and highwaympg has a negative effect on price. These features are related with efficient fuel consumption. If a car's citympg or highwaympg is high, then their motor size is being small. Since small sized motored cars are cheap, these features affect the price negatively. Therefore, all of the numerical features in the dataset show the expected relationships with car prices.

In order to comment on the correlation of categorical features with price, conversion of these features to numerical versions of them was needed. Applied different encodings to decide which method is the most effective. One-hot encoding yields many features but having this amount of feature was not efficient in this case. So, label encoding has been used.

The only feature has not been converted to numeric data is the **CarName** feature, which is in the string type and contains the car brand and model. At first, instead of whole car model name, manufacturer of the car has been taken and label encoding has been applied. However, label encoding makes a value definition for each producer by queue. This is problematic. To give an example, if car manufacturer value being 0 for Porsche, and 3 for Toyota, and 7 for BMW; then there won't be healthy relation among this feature and price. Afterward one more analysis has been done by ranking manufacturers based on the correlations of them with price of vehicle. Noticed that less-known producers may exist on top while other luxury brands ranked lower than them. "Buick" which is a mid-upper segment car brand has been ranked first while there are brands such as Porsche. Dataset is biased to Buick because distribution is not normal. At this point, **CarName** feature has been removed from the dataset and look at the correlation matrix of all the features with the price again. As seen in Figure 3, there are 5 features whose correlation with price is nearly 0. We removed these 5 features from this data set: *fueltype, enginetype, doornumber, cylindernumber, carbody.*

Figure 4 illustrates the final dataset features with whole correlation matrix. As citympg, highwaympg relation with engine size has been mentioned before; there are several features that affect each other. This situation leads to multicollinearity. Multicollinearity affects regression models negatively because decision making of models being affected by mis-understanding of features due to high correlation of independent variables. Management of multicollinearity has been considered after reaching model training phase. Reason of this will be explained in the fourth section as well.

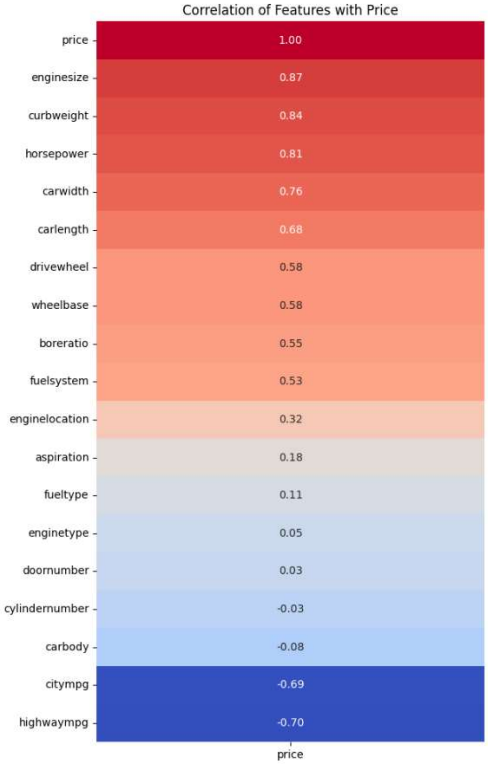

Fig. 3. Price correlation of cleaned dataset, categorical features are included.

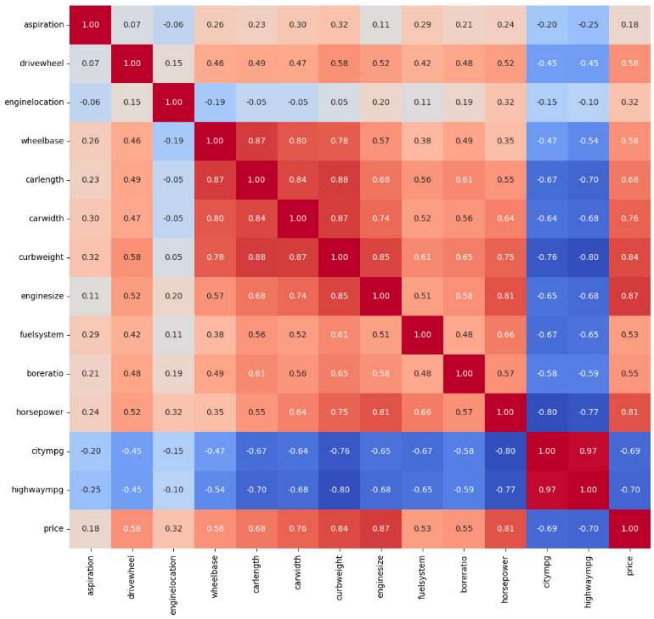

Fig. 4. Correlation matrix of totally cleaned dataset.

IV. Price Prediction

A. Regression Analysis

Using the cleaned dataset, regression has been performed with several beneficial models. These models are as follows:

- **Linear Regression:** Linear modelling method, provides relation of target value with one or more independent variables.
- **Lasso:** Regression method that performs feature selection and L1 regularization. Efficient for multicollinearity situations.
- **Ridge:** Another regression method that decreases effects of the multicollinearity by using L2 regularization. However it can not totally solve relation of high correlated features.
- **Decision Tree Regressor:** This regression method has a tree structure. Decision tree makes feature selection without regarding correlation between features. Only focuses on target label and features. Therefore decision trees are not being affected from multicollinearity.
- **Random Forest Regressor:** Ensemble model for regression tasks. Consists of multiple decision trees, thus it is called Random Forest.
- **Gradient Boosting Regressor:** Machine learning method to manage regression tasks. Ensembles weak learners to create a strong predictive model by optimization during iterations. This weak learner is generally decision tree. Differs from random forest because in gradient boosting, trees are built iteratively, each tree learns from the errors of the previous tree. However in random forest, trees are trained independently in parallel.
- **Forward Neural Network (FNN):** Deep learning technique that can be used for regression. Two hidden layered neural network has been tried to solve the task.

Also, log transformation has been used to improve linear relationships in the target variable since normal distribution or linear relationships in linear regression models may improve the performance. Trials has been done in Linear Regression method and better metrics have been received with using log transformation.

```
y_train_log = np.log(y_train+1)
 y_test_log = np.log(y_test+1)
```

Fig. 5. Code snippet for log transformation using Numpy.

B. Results of Regression

Each model gives insights about regression task and their working principle with their metrics. R^2 is a scoring that is being used for regression tasks. This metric has been considered during model training and hyperparameter tuning phases.

To be able to select best hyperparameters, hyperparameter tuning has been done by using **GridSearch** and **RandomSearchCV**. These methods have been used for all models except Forward Neural Network. In FNN, hyperparameters have been selected with considering the task, inspecting loss, preventing overfitting with applying early stopping.

Linear Regression gives better metrics than expected. Normally, unsuccesful model is expected since dataset has a multicollinearity problem. Even without using any methods to manage the multicollinearity, reached R^2 scores are 0.8772, 0.8146 for training and test sets, respectively.

Our aim was showing that linear regression is being affected from multicollinearity more than other specific models. Therefore, no methods have been applied to dataset to handle this effect. However, feature selection may be done with analyzing results of different methods such as Variance Inflation Factor (VIF), Lasso (feature importance ranking), Principal Component Analysis (PCA) (Chan et al., 2022).

In Linear Regression, *y_train*, *y_test* and also *y_train_log, y_test_log* has been used to fit two different linear models. Compared these two models to show that log transformation improves metrics. R^2 scores for training, test, training (with log transformation), test (with log transformation) are 0.8772, 0.8146, 0.8909, 0.8761 respectively. Therefore log transformation yields slightly better metrics.

By applying L1 regularization, **Lasso** effectively reduced the impact of less important features, potentially addressing multicollinearity to some extent. R^2 scores for training, and test sets are 0.8334, and 0.8961 respectively. GridSearchCV has been used to find optimum hyperparameters for Lasso. Hyperparameter tuned Lasso gives R^2 scores of 0.8909, and 0.8763 for the training and test sets respectively.

Also by applying L2 regularization, **Ridge** regression demonstrated high metrics as well. Hyperparameter tuned Ridge yields 0.8909, and 0.8767 R^2 score for training and test sets which shows that Ridge regression can generalize and learn the dataset.

Decision Tree Regressor was essential to use in this case since there is a multicollinearity in the dataset and also tree-based algorithms are not being affected from this phenomena like the other algorithms. Thus, decision tree regressor showed its potential with giving 0.9326, and 0.8722 R^2 scores in the training and test sets. RandomizedSearchCV has been used to tune hyperparameters. However, metrics did not show any increase and 0.8982 for the training set, and for the test set 0.8744 R^2 score has been achieved.

Random Forest Regressor is one of the most suitable algorithms for the case since it is a tree-based algorithm and also it is reliable regarding several decision tree. Random Forest Regressor succesfully met high expectations with R^2 score of 0.9486 for the training set, and 0.9304 R^2 score for the test set.

Gradient Boosting Regressor is one of the most famous algorithm that is used by the data science community to deal with regression problems. It is famous because it gives incredible results for almost each task. Among each model that has been tried to deal with this dataset, Gradient Boosting Regressor demonstrated exceptional performance. With an R^2 score of 0.9853 on the training set and 0.9380 on the test set, highlighting its capability to capture complex patterns and relationships in the data. This robustness and perfect ability to generalize makes itself one of the most suitable model for this prediction task.

Each model with their scores has been represented in Table II. It can be seen from the table, the most suitable model is Gradient Boosting Regressor regarding R^2 scores. However Random Forest Regressor makes big impact with its MSE (Mean Square Error). These metrics gave us insight about not log transforming prices and training all of the models again.

Although it is consistent to compare R^2 scores of models trained with log transformated and non-transformated data, this is not the case when considering MSE. Therefore inverse transformation of predicted and real values has been realized, as it can be seen from Figure 6, to compute MSE. Afterward these MSEs have been compared.

```
y_pred_train_original = np.exp(y_pred_train)- 1
 y_pred_test_original = np.exp(y_pred_test)- 1
```

Fig. 6. Code snippet for inverse log transformation using Numpy.

Trials showed that non-transformed data suits better than log-transformed data just for the random forest. Thus, an additional Random Forest Regressor with its metrics has been added to the second row of the Table II.

Mean price for the dataset is 13276.71 dollars while Random Forest with R^2 score of 0.9454 has an RMSE around 2075.1979. This is about 15.6% of the mean price. Thus, it is acceptable for general market analysis.

V. Conclusion

In this project, a machine learning (ML) solution capable of predicting the price of a newly released car was intended to be developed.

To achieve this, we began by analyzing the dataset and eliminating unnecessary variables, while doing that the phenomenon of multicollinearity has been observed among the features. The data was divided into two distinct subsets for training and testing purposes. Using these two sets, various regression models and a deep learning model were trained and tested.

TABLE II
Comparison

Additional	Model	MSE Score	R^2 Score
Without Log Transformation	Linear Regression	14.634.839,17	0.8146
	Random Forest	**4.306.446,56**	**0.9454**
Without Hyperparameter Tuning	Lasso	11.785.870,12	0.8961
	Ridge	12.349.494,56	0.8961
	Decision Tree	10.301.558,62	0.8722
With Log Transformation	Random Forest	**4.587.603,75**	0.9304
	Gradient Boosting	6.821.985,55	**0.9380**
	Linear Regression	13.113.161,05	0.8761
With Log Transformation Hyperparameters Tuned by GridSearchCV or RandomSearchCV	Lasso	13.091.974,99	0.8763
	Ridge	12.985.302,92	0.8767
	Decision Tree	10.902.571,67	0.8744
	Random Forest	5.880.285,95	0.9142
	Gradient Boosting	6.821.985,55	**0.9380**
Deep Learning	FNN	16.788.380,07	0.7873

The test results demonstrated that the Random Forest and Gradient Boosting models show the best performances for the task. MSE and R^2 scores are consistent with each other for all models. To improve the accuracy of all models, additional optimization techniques were applied, and the best results that could be obtained were compared.

This study shows the potential of machine learning and deep learning models in predicting car prices, offering a valuable tool for the automotive industry in pricing strategies and market entry planning.

VI. Future scope

The model developed in this project has the potential to be expanded and form a basis for more complex problems, such as predicting used car prices. By leveraging larger and more comprehensive datasets for training, it is possible to develop more accurate ML models.

Moreover, the applicability of this model extends beyond car price prediction; it can also be utilized for pricing other motor vehicles with similar feature types. This adaptability shows the model's potential to contribute to various fields within the automotive and transportation industries.

REFERENCES

[1] D. J. Mala and V. Sudhish, "Machine Learning-Based New Model Release Car Price Prediction," *2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE)*, Rourkela, India, 2024, pp. 1-6, doi: 10.1109/ICSPCRE62303.2024.10674864.

[2] J. Schlimmer, 1987, "Automobile," 1985 Ward's Automotive Yearbook. [Online]. Available: https://www.kaggle.com/datasets/hellbuoy/car-price-prediction.

[3] N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," *2018 5th International Conference on Business and Industrial Research (ICBIR)*, Bangkok, Thailand, 2018, pp. 115-119, doi: 10.1109/ICBIR.2018.8391177.

[4] C. V. Narayana, C. L. Likhitha, S. Bademiya and K. Kusumanjali, "Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business," *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2021, pp. 1680-1687, doi: 10.1109/ICESC51422.2021.9532845.

[5] J. Varshitha, K. Jahnavi and C. Lakshmi, "Prediction Of Used Car Prices Using Artificial Neural Networks And Machine Learning," *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2022, pp. 1-4, doi: 10.1109/ICCCI54379.2022.9740817.

[6] Chan, J. Y. L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z. W., & Chen, Y. L. (2022). Mitigating the multicollinearity problem and its machine learning approach: a review. *Mathematics*, *10*(8), 1