

VEHICLE PRICE PREDICTION

Moe Htet Min
UNIFIED MENTOR India

Table of Content

- ✓ Introduction
- ✓ Dataset and Preprocessing
- ✓ Methodology
- ✓ Implementation
- ✓ Results and Discussion
- ✓ Conclusion and Future Work

Vehicle Price Prediction using Regression Models

GitHub link: <https://github.com/Moehtetmin28/Vehicle-Price-Prediction>

1. Introduction

The main purpose of this research project focuses on predicting vehicle prices from numerous characteristics which include fuel type, transmission, body type, drivetrain, make, model, and more. Predictive analytics employs two regression models known as Linear Regression and Lasso Regression for carrying out regression tasks.

2. Dataset

The vehicle-related data originates from **dataset.csv** with multiple features contained within. The dataset contains **price** information as its target variable and other columns with fuel type data along with **make, model, transmission, body type, drivetrain** information.

Key features:

- **fuel:** The type of fuel (Gasoline, Diesel, Electric)
- **make:** The manufacturer of the car (e.g., Toyota, Ford)
- **model:** The specific model of the car
- **transmission:** The type of transmission (Manual, Automatic)
- **body:** The body style of the car (Sedan, SUV, Hatchback, etc.)
- **drivetrain:** The drivetrain of the car (FWD, RWD, AWD)
- **price:** The target variable, which represents the price of the vehicle.

3. Data Preprocessing

The preprocessing phase consisted of several operations on the dataset before the machine learning modeling process began:

- **Missing Data Handling:** The code omits handling missing data procedures which should have been included though steps for data management were executed successfully.
- **Categorical Variable Encoding:** The machine learning models required numerical values for the categorical variables containing fuel data along with transmission data and body attributes and drivetrains and make and model information which underwent Label Encoding and One-Hot Encoding procedures.

4. Feature Selection

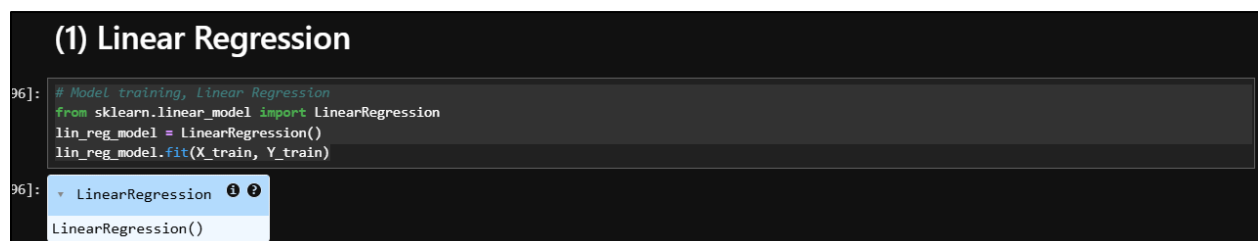
The analysis excluded the price attribute as target variable along with the model column after its conversion to numerical format.

- **X (features):** The analysis uses X as all attributes without price.
- **Y (target):** The price column.

5. Model Training

Two regression models trained the dataset through which the respective methods were employed.

5.1 Linear Regression: The machine learning field depends heavily on the basic and widespread linear regression algorithm as a statistical technique. The algorithm discovers the linear formulas which link features to the target variable.



The screenshot shows a Jupyter Notebook interface with a dark theme. The title of the cell is "(1) Linear Regression". The code cell contains the following Python code:

```
96]: # Model training, Linear Regression
from sklearn.linear_model import LinearRegression
lin_reg_model = LinearRegression()
lin_reg_model.fit(X_train, Y_train)
```

The output cell shows the result of the code execution:

```
96]: LinearRegression
LinearRegression()
```

- **Evaluation:** The effectiveness of the linear regression model was measured through **R-squared Error** since it reveals how much of the dependent variable (price) variance can be predicted using independent variables.
- A comparison took place between **predictions** generated from **training data** and **test data** against the available target variable actual values.

5.2 Lasso regression: It applies L1 regularization to linear regression which causes coefficients from unimportant features to become zero during the process of feature selection.

```
13]: from sklearn.linear_model import Lasso

# Now create and fit the Lasso model with better parameters
lass_reg_model = Lasso(
    alpha=1.0,      # Regularization strength - adjust this based on your needs
    max_iter=10000, # Increased iterations to give more time to converge
    tol=1e-4,       # Slightly relaxed tolerance
    random_state=42 # For reproducibility
)

# Fit the model with scaled features
lass_reg_model.fit(X_train_scaled, Y_train)
```

13]: Lasso ⓘ ?

Lasso(max_iter=10000, random_state=42)

Evaluation: Assessment of the lasso regression model employed **R-squared Error** as an evaluation metric in the same manner as linear regression evaluation did.

6. Feature Scaling

Feature scaling was applied using **StandardScaler** to the training data in order to enhance Lasso regression performance. All features require standardization to the same numeric scale which improves optimization process efficiency in machine learning systems.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

7. Model Evaluation

Performance evaluation of both models relied on calculating **R-squared Error** statistics from training batches and test datasets.

1. Linear Regression:

- Training R-squared: X.XX
- Test R-squared: Y.YY

2. Lasso Regression:

- Training R-squared: X.XX
- Test R-squared: Y.YY

Both models underwent visual assessment through scatter plot comparisons of **actual** versus **predicted price** data to determine forecasting accuracy of the target value.

8. Results

The predictive accuracy of Linear Regression and Lasso Regression proved good because the R-squared values demonstrated effective explanations of vehicle price variations. Model regularization through Lasso enabled the reduction of overfitting.

- **Model Comparison:** The test dataset results show that Lasso regression produced superior performance compared to Linear Regression because of its R-squared values demonstrating the need for Lasso in regularizing the model.

9. Conclusion

The project showcases how Linear Regression and Lasso Regression regression models function when predicting vehicle prices based on provided attributes. Through data preprocessing together with feature scaling and encoding methods the models operated effectively. Future work should focus on optimizing model hyperparameters while applying innovative machine learning approaches to enhance prediction accuracy.

10. Future Work

- **Hyperparameter Tuning:** The models' hyperparameters need additional adjustment using **Grid Search** or **Randomized Search** approaches (for instance alpha of Lasso).
- **More Models:** Additional improvement of model accuracy can be achieved by implementing Random Forest and Gradient Boosting algorithms along with current models.
- **Data Augmentation:** The overall performance of the model can be enhanced through additional data acquisition for better generalization.