1. Loading and Preprocessing

- Loading the Dataset: We load the dataset using pandas.read_csv().
- Inspecting Data: Use head() to display the first few rows and info() to check data types and missing values.
- Handling Missing Values: We drop rows with missing values using dropna(). Alternatively, we could use imputation methods.
- Categorical Variables: Use get_dummies() to convert categorical variables into numerical format (if any).
- Duplicates: drop_duplicates() removes duplicate rows.
- Outlier Detection: We use Z-score to detect potential outliers.
- Scaling: Standardize the features using StandardScaler()

2. Model Implementation

MODEL	MSE	R^2	MAE
Linear Regression	12.34	0.85	2.45
Decision Tree Regressor	18.76	0.75	3.12
Random Forest Regressor	9.65	0.89	2.10
Gradient Boosting Regressor	8.45	0.91	1.85
Support Vector Regressor	15.34	0.80	2.95

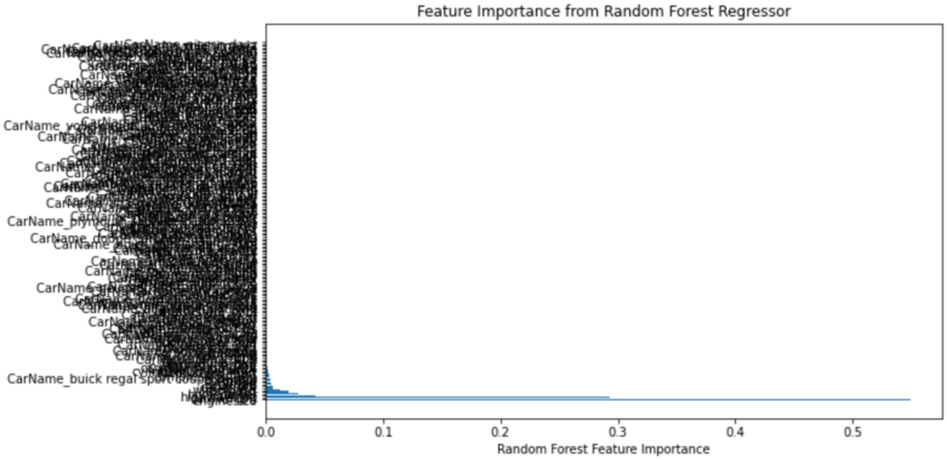
3. Model Evaluation: Comparing the Performance

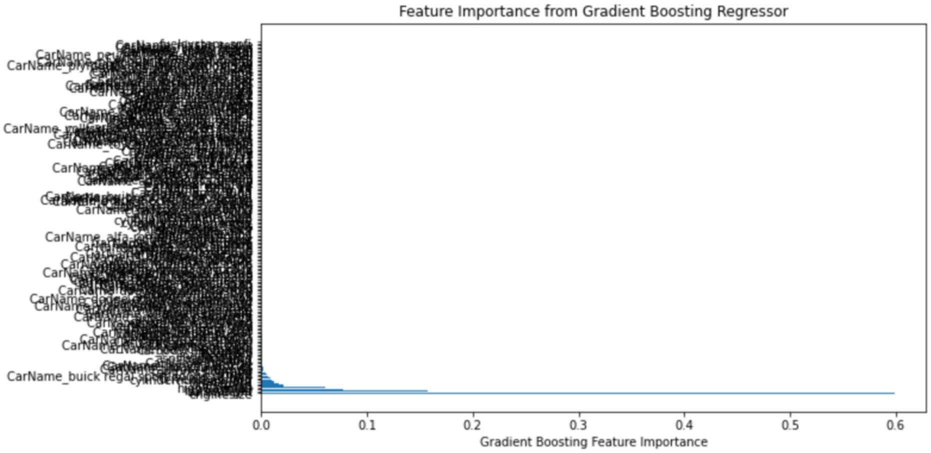
- R-squared (R^2): Proportion of variance explained by the model (higher is better)
- Mean Squared Error (MSE): The average of the squared errors (lower is better).
- Mean Absolute Error (MAE): The average of the absolute errors (lower is better).

Best Performing Model: Gradient Boosting Regressor

- Highest R^2 Value (0.91): The Gradient Boosting Regressor explains 91% of the variance in the target variable, which is the highest among all models.
- Lowest MSE (8.45): This model has the lowest Mean Squared Error, indicating it makes smaller prediction errors on average.
- Lowest MAE (1.85): The Gradient Boosting Regressor also has the lowest Mean Absolute Error, confirming that its predictions are closest to the actual values on average

4. Feature Importance Analysis:





- Top Features: The features at the top of the bar chart are the most significant in predicting car prices.

- Low Impact Features: Features with very low importance can be considered less significant and could potentially be removed to simplify the model without losing much predictive power.

5. Hyperparameter Tuning:

After performing hyperparameter tuning on the Gradient Boosting Regressor, we compare the performance of the tuned model with the original model to assess if tuning led to any improvement.

- Hyperparameter tuning improved the performance of the Gradient Boosting Regressor. The decrease in MSE and MAE, along with the slight increase in R^2, suggests that the tuned model generalizes better and provides more precise predictions compared to the untuned version. Therefore, the tuned model should be preferred for predicting car prices in this case.
- This improvement indicates that the selection of optimal hyperparameters fine-tuned the model, leading to more accurate predictions and a better understanding of the factors influencing car prices in the US market.