

Car Dheko

Used Car Price Prediction



**Project Title: Car Dheko – Used Car Price Prediction**

**Name: Vasanth ram**

**Date: 28/11/2024**

**Batch: MDTM28**

**Course: Data Science**

**Institution: Guvi**

**Abstract**

In my role as a data scientist at Car Dheko, I aim to significantly enhance the customer experience by simplifying how used car prices are determined.

This project focuses on predicting the price of used cars using machine learning regression techniques. The dataset, scraped from the Car Dheko website, contains both numerical and categorical features related to used cars, which were initially uncleaned and required thorough preprocessing. Several regression models, including Linear Regression, Decision Trees, Random Forest, and Gradient Boosting, were evaluated using K-fold cross-validation to determine the best performing model. Among these, the Random Forest Regressor demonstrated superior performance, achieving the lowest Mean Squared Error (MSE) and standard deviation.

The model was trained on 80% of the dataset and tested on the remaining 20%, with the following results on the test set: a Mean Squared Error of 5.37, Mean Absolute Error of 1.07, R² score of 0.93, and a Mean Absolute Percentage Error (MAPE) of 14.66%. On the training set, the Random Forest model showed a high R² score of 0.99, indicating excellent generalization. This model offers promising results for predicting used car prices, with potential applications for companies such as Car Dheko and the general public who seek reliable pricing tools for used vehicles.

**Approach**

**1.Data Processing:**

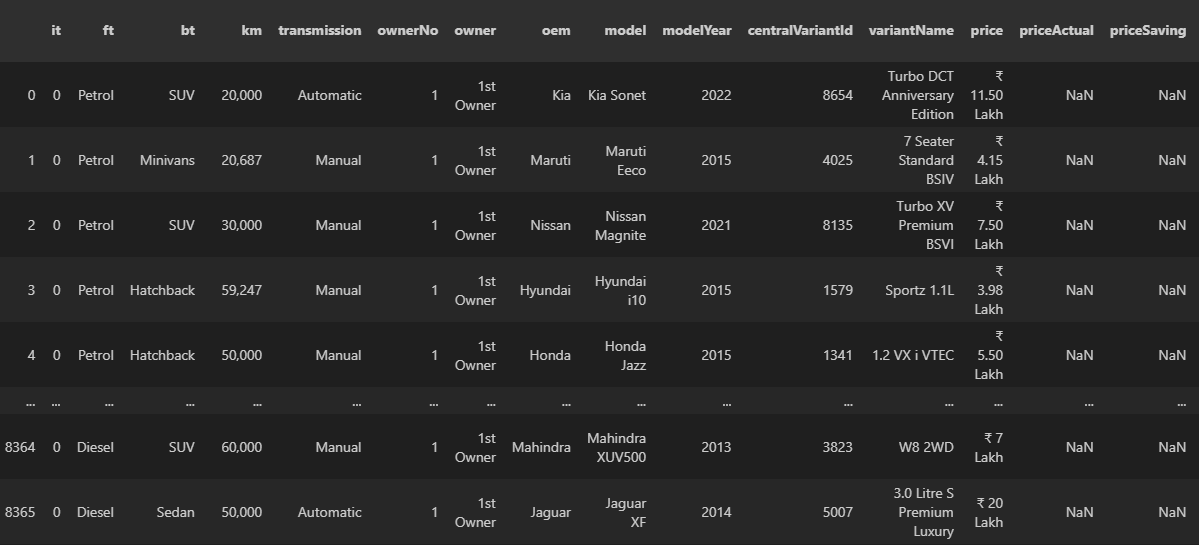
**1.1 Import and Concatenate:**

* Import all city datasets which is in unstructured format (scrapped form cardehko site).
* We cleaned those unstructured data and make it as structured data (in table format).
* Add a new column named 'City' to assign the respective city name for all rows. (6 different cities in data sets)
* Concatenate the datasets to form a single, unified dataset for data preprocessing and model development.

**From this:**



**To this:**



**1.2 Handling Missing Values:**

* We identify and address missing values in the dataset.
* For numerical columns such as (Km, Mileage, Engine displacement etc..), we apply imputation techniques like mean, median, or mode (used relevantly) to fill null values in the dataset.
* For categorical columns such as (Fuel type, Transmission, ome, Insurance, Owner etc…), we used mode imputation to represent missing values.

**1.3 Standardizing Data Formats:**

* Ensure data types are consistent across columns.
* Numerical column like Km, Max power, Milage, Price etc… removed their units (km, cc, lakhs etc..) and converted the data type to integers.

**1.4 Encoding Categorical Variables:**

* Convert categorical features into numerical values using encoding techniques.
* Used label encoding for nominal categorical variables (e.g.: model, city, Colour, Seats etc…).
* Used ordinal encoding for ordinal categorical variables (e.g., owner, model Year, Insurance Validity etc…).

**1.5 Removing Outliers:**

* Identify and removed outliers to prevent distortion of the model.
* Use methods like the IQR (Interquartile Range) analysis to detect and handle outliers. (higher bound value 0.85, lower bound value 0.15)
* IQR method only used for columns like Km, Mileage, Max power etc. excluding Target column (Price).

**1.6 Pickling Encoders:**

* Using Pickel Module, We Pickled label and ordinal encoder of all categorical feature for later use.
* These pickled files will help us in streamlit to encode dataframe into numerical for price prediction.

**2.** **Exploratory Data Analysis (EDA):**

**2.1 Descriptive Statistics:**

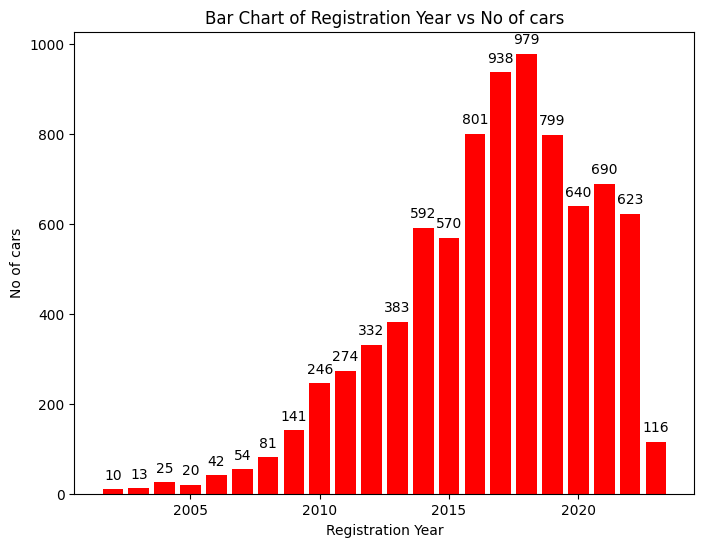
* Calculate key summary statistics to understand the distribution of the data.
* Metrics like Mean, Range, Variance, Standard deviation, Min, Max, Quartiles, Skewness will be considered.

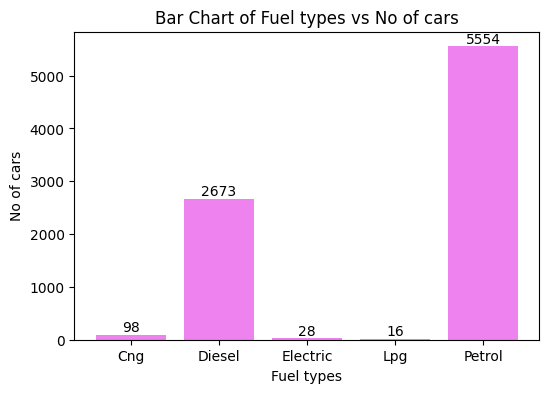
**2.2 Data Visualization:**

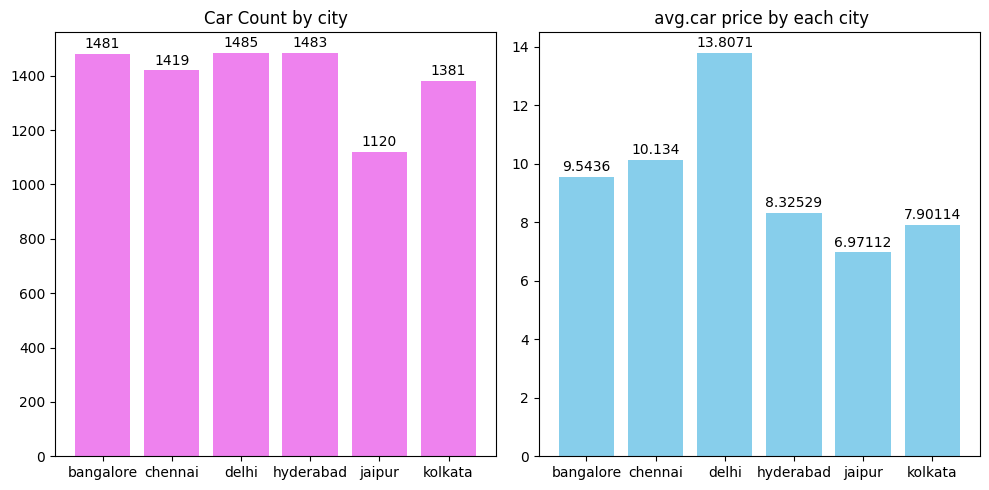
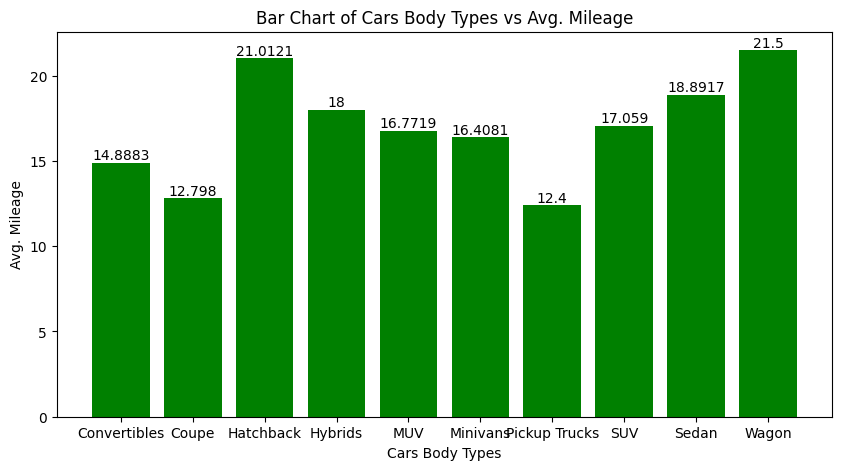
* Use visualizations to identify patterns, trends, and correlations in the data.
* Created scatter plots, histograms, box plots, and correlation heatmaps to understand the relationships between features and car prices.
* Used Matplotlib and Seaborn Modules.

**2.2.1 Bar plot:**

A bar plot (or bar chart) is a simple way to visualize and compare different categories of data. Each category is represented by a bar, and the height or length of the bar shows how much or how often that category occurs.

These are few bar chart:****

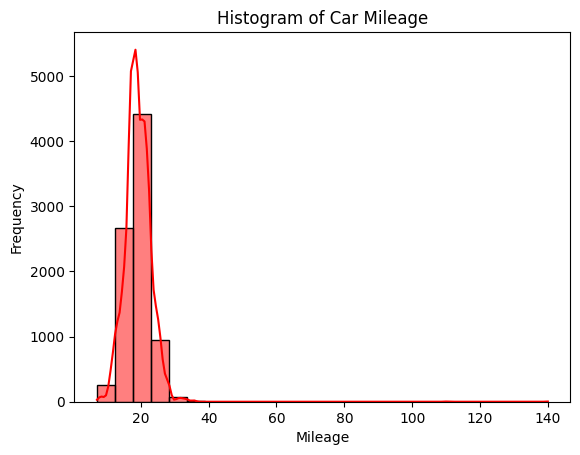
****

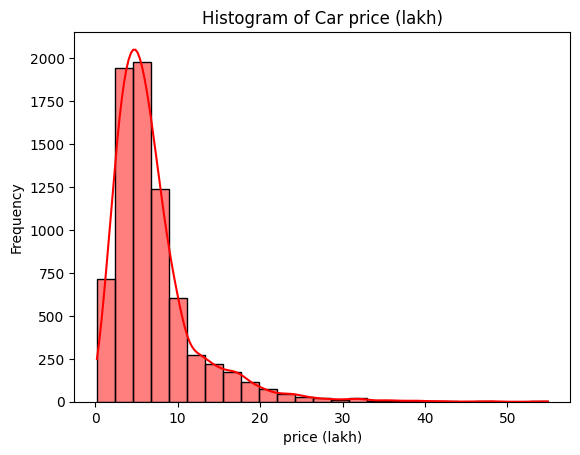
****

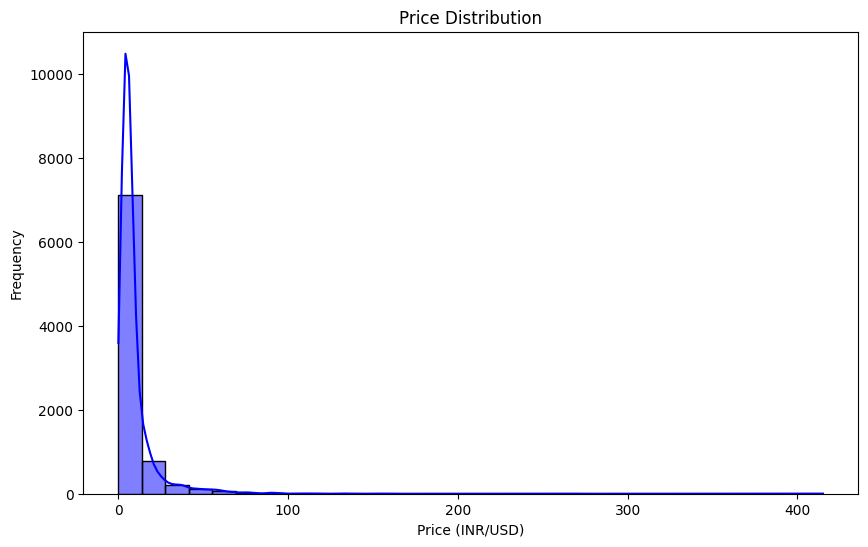
**2.2.2 Histogram plot:**

A histogram is a type of chart used to show how continuous data is distributed. It takes your data and groups it into ranges, called bins, then displays how many data points fall into each bin.

These are few histogram plots:



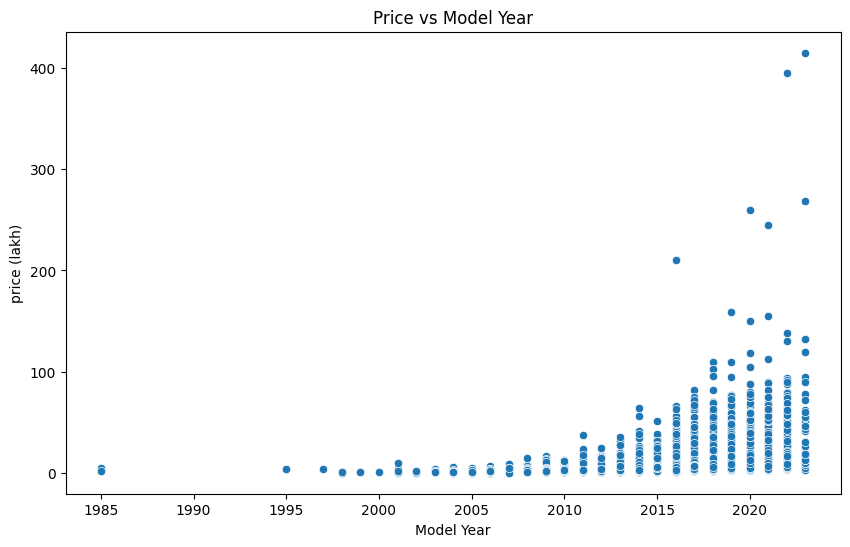


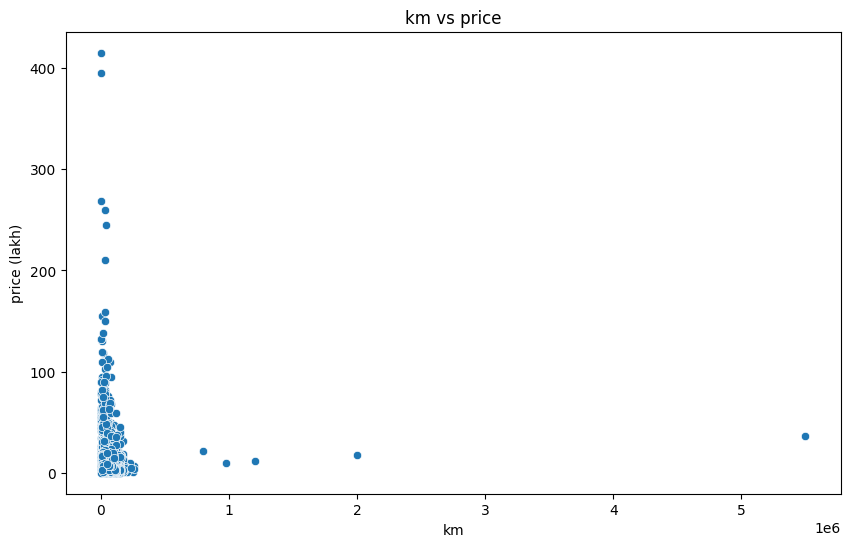


**2.2.3 Scatter plot:**

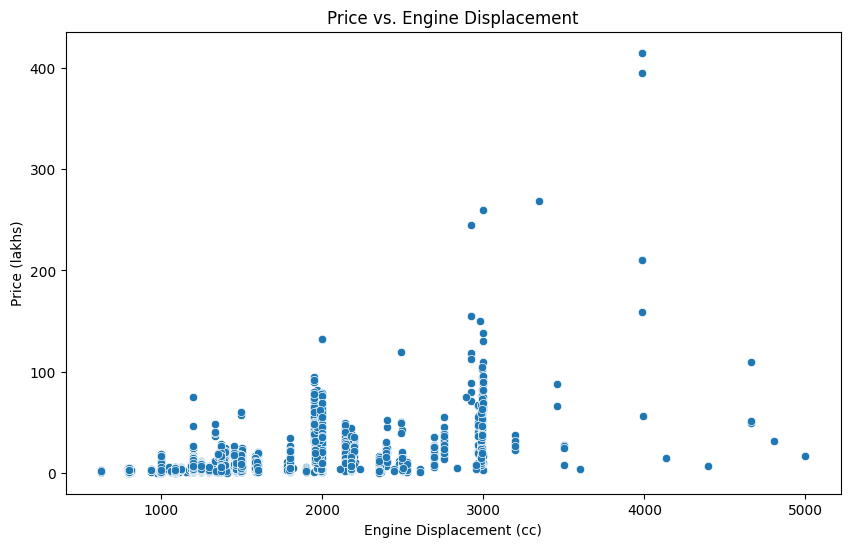
A scatter plot is a type of chart that displays data points as dots on a two-dimensional plane, with each dot representing a value from two variables.

These are few scatter plots:



****

****

****

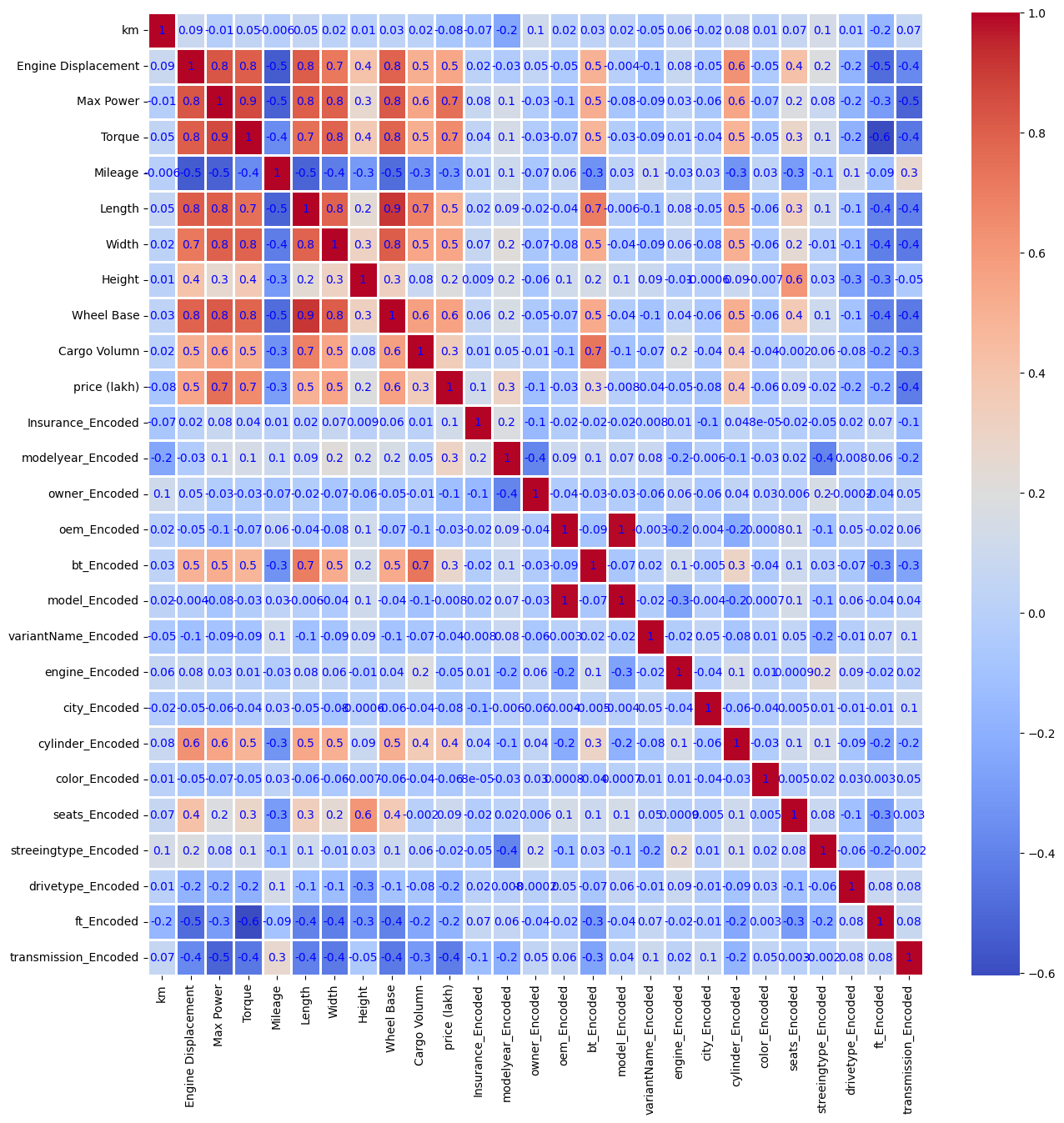
**2.3 Feature Selection:**

* Identify important features that significantly influence car prices.
* Feature selection has been done through techniques like correlation analysis, feature importance from models, and domain knowledge.

**2.3.1 correlation analysis:**

We used Heat map for Correlation analysis and these are the visualization:

All features:



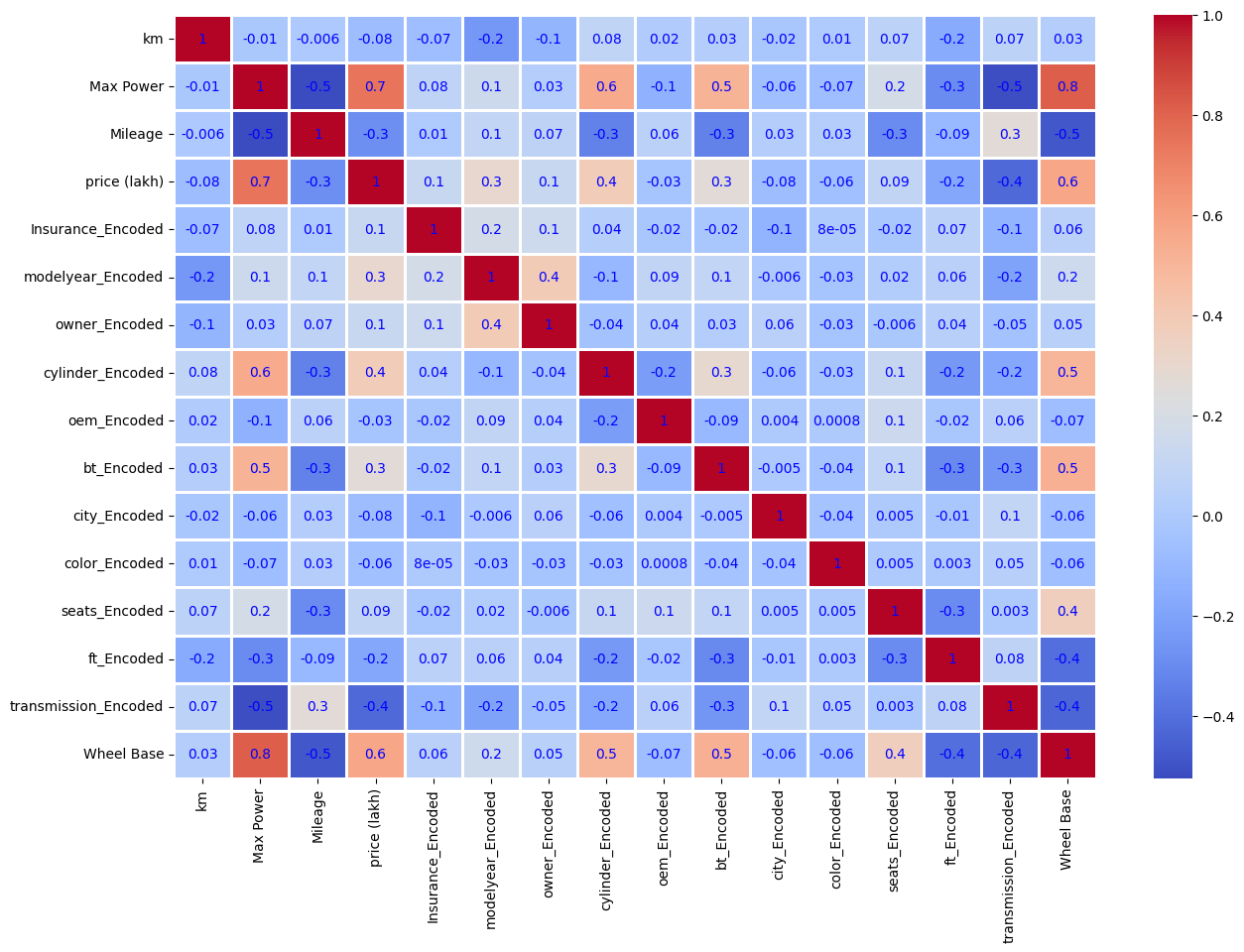
**2.3.2 correlation with Target:**

|  |  |
| --- | --- |
| **Features** | **Correlation** |
| price (target feature) | 1.000000 |
| Max Power | 0.600381 |
| Wheel Base | 0.585649 |
| model year | 0.584773 |
| Body type | 0.424868 |
| owner | 0.235966 |
| cylinder | 0.181174 |
| seats | 0.177758 |
| Insurance | 0.106261 |
| oem | -0.003087 |
| colour | -0.037181 |
| city | -0.130413 |
| Fuel type | -0.179511 |
| Mileage | -0.189768 |
| km | -0.243398 |
| transmission | -0.317623 |

**2.3.3 Selected Feature (Domain knowledge):**

Finally, we have selected 16 columns as model feature (15 independent variable and 1 as target), these features were selected by above correlation heatmap, feature importance score, domain knowledge.

Here are the selected feature correlation heatmap:



**3.** **Model Development:**

**3.1 Train-Test Split:**

We split the dataset into training and testing sets to evaluate model

performance. Here training set have 80% data and test set have 20% data

(most common ratio 70-30 or 80-20)

**3.2 Model Selection:**

* Model selection is all about Choosing appropriate machine learning algorithms for price prediction.
* Here we are considering models like **Linear Regression**, **Decision Trees Regressor**, **Random Forests Regressor**, and **Gradient Boosting Regressor**.
  1. **Model Training:**
* Train the selected models on the training dataset and Using Cross-Validation for Robust Performance. Cross-Validation reducing the risk of overfitting and providing better generalization.
* To train the models effectively, **cross-validation** is used to ensure that the model's performance is stable and not overly dependent on a single train-test split.
* We are using **K-fold cross-validation**, where the dataset is split into 'k' subsets or folded into “k” no of subsets.
* After cross validating all models by its performers (MSE,Std) ,we finally taking Random Forest Regressor as our model.

**3.4 Hyperparameter Tuning:**

* Hyperparameter tuning will optimize Model Parameters to Improve Performance. Initially the model parameters are in default values, to improve the model's accuracy and generalization we have to adjust the parameters.
* Here we are using Grid Search, this technique exhaustively searches through a predefined set of hyperparameters, evaluating every possible combination to find the best performance.
* Using Grid search for random forest regressor will gives us best parameters for n estimators, max depth, min sample split and min sample leaf values.

**4.** **Model Evaluation:**

**4.1 Performance Metrics:**

* Evaluate model performance using relevant metrics such as
  1. Mean Squared Error (MSE)
  2. Mean Absolute Error (MAE)
  3. Mean Absolute Percentage Error (MAPE)
  4. R2 Score (R square)

**4.2 Model Comparison:**

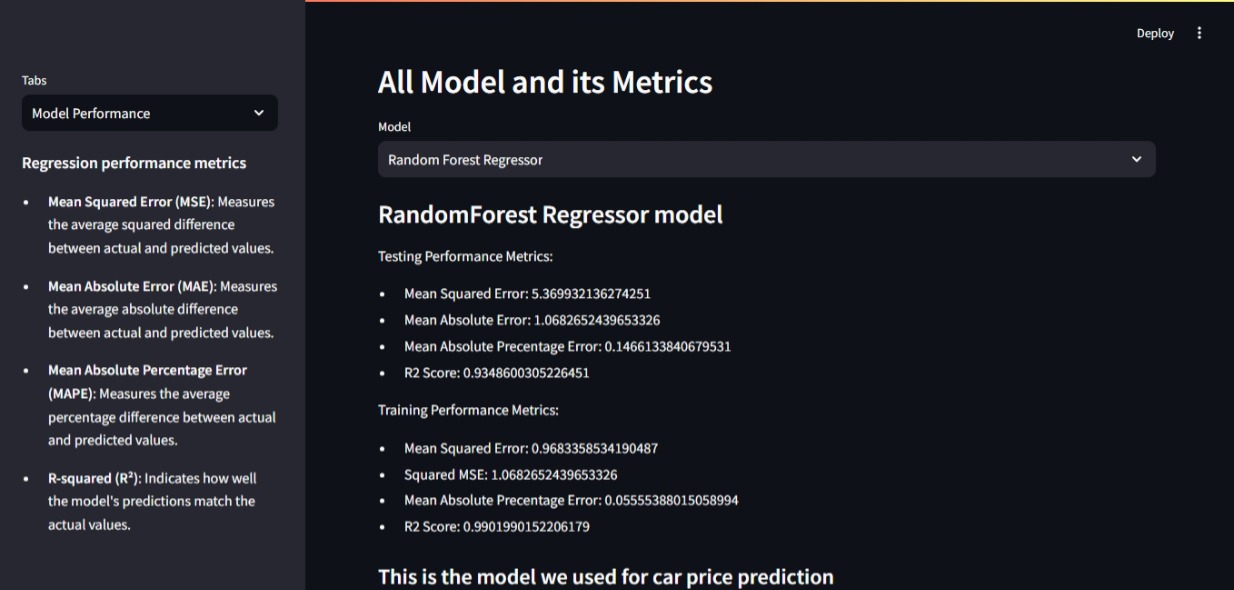
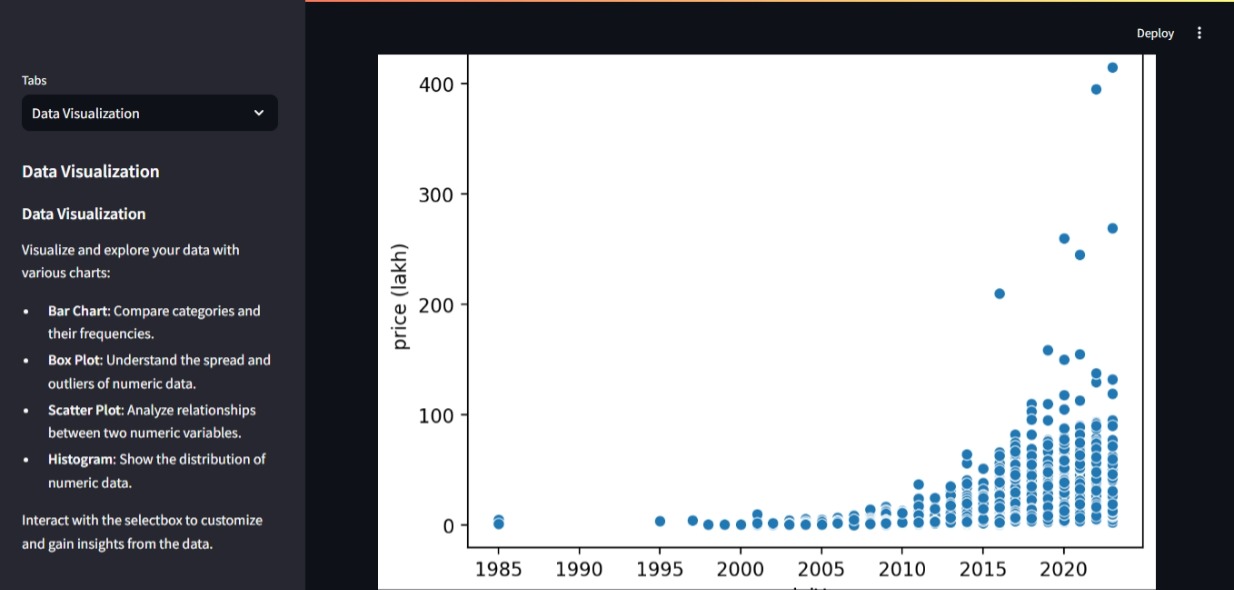
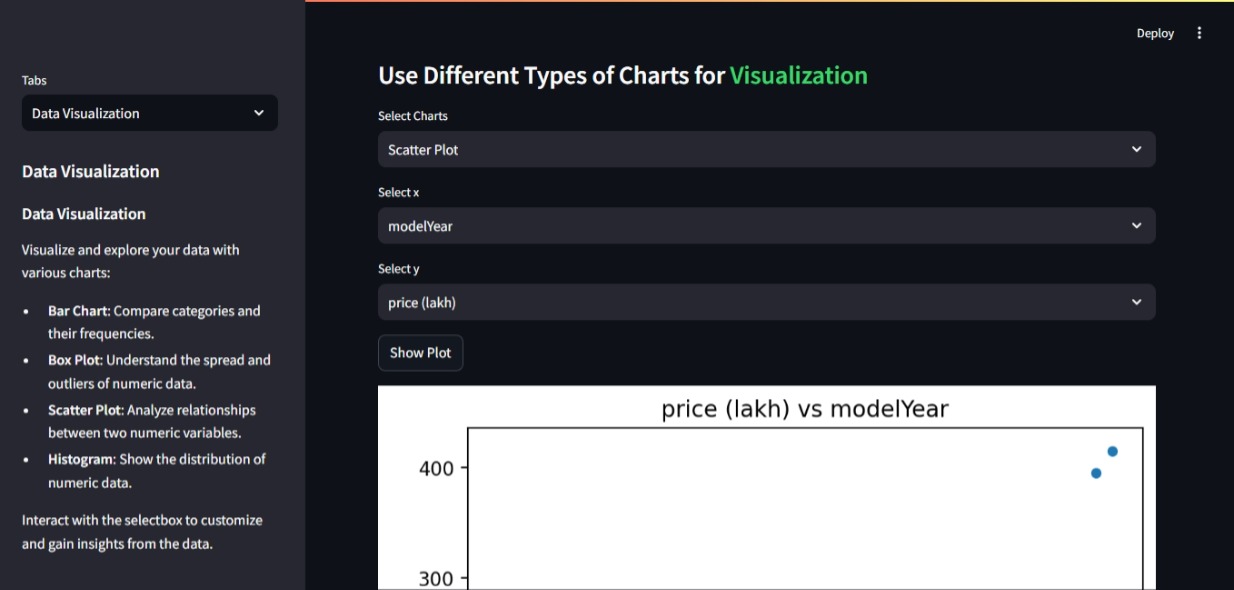
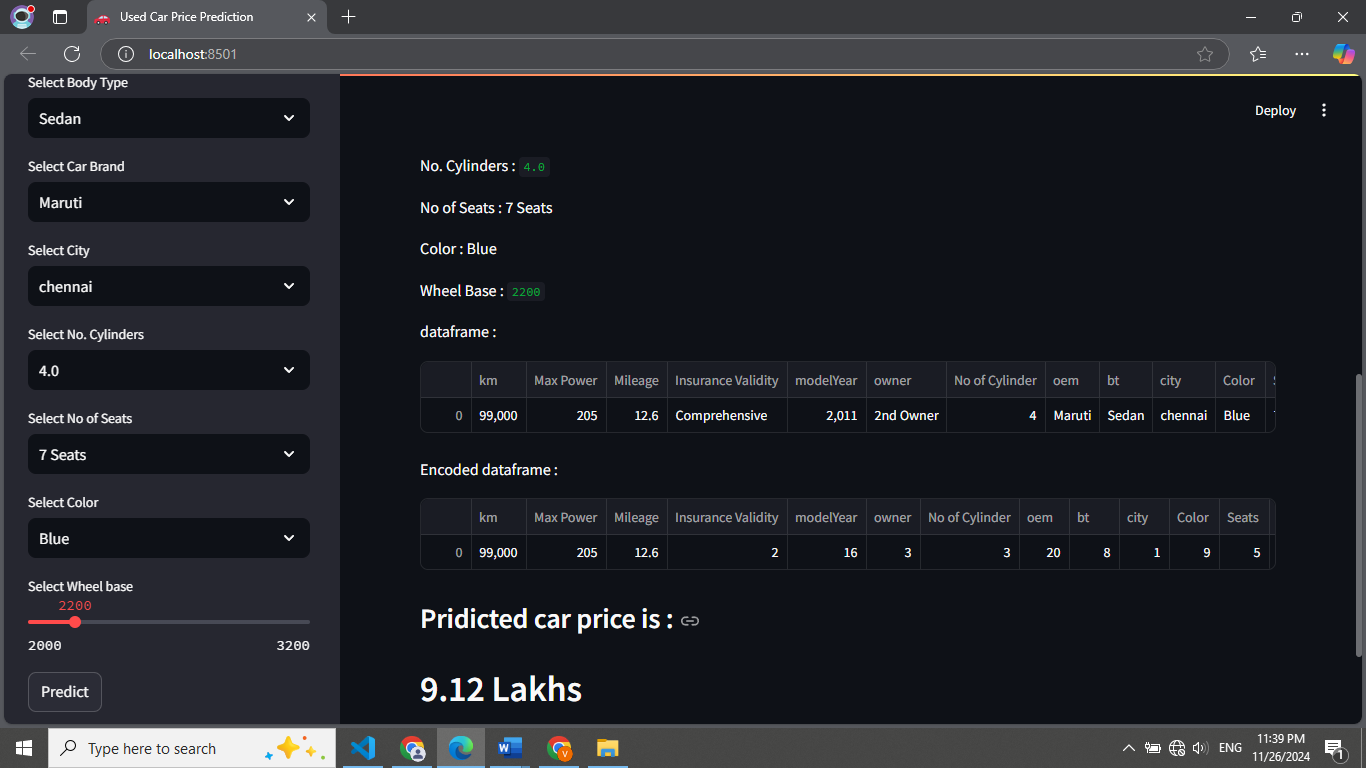
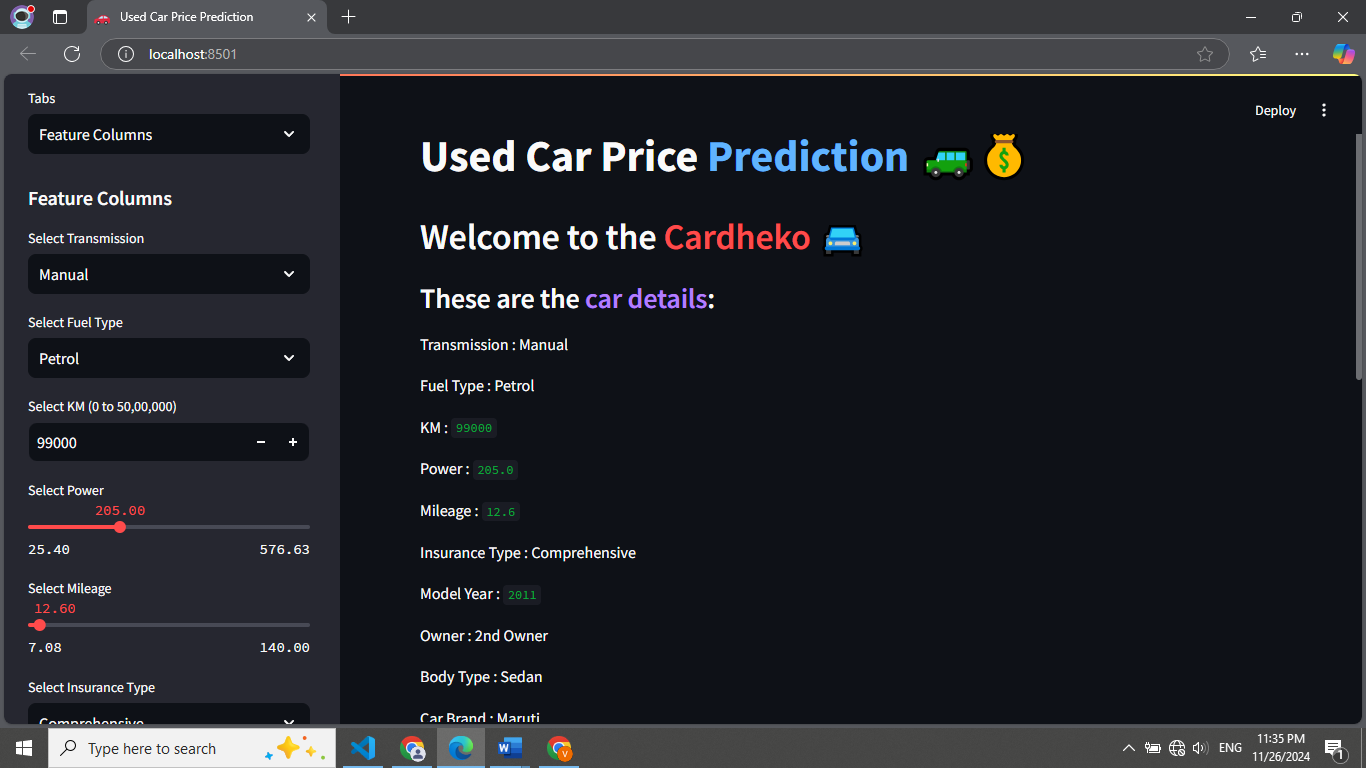
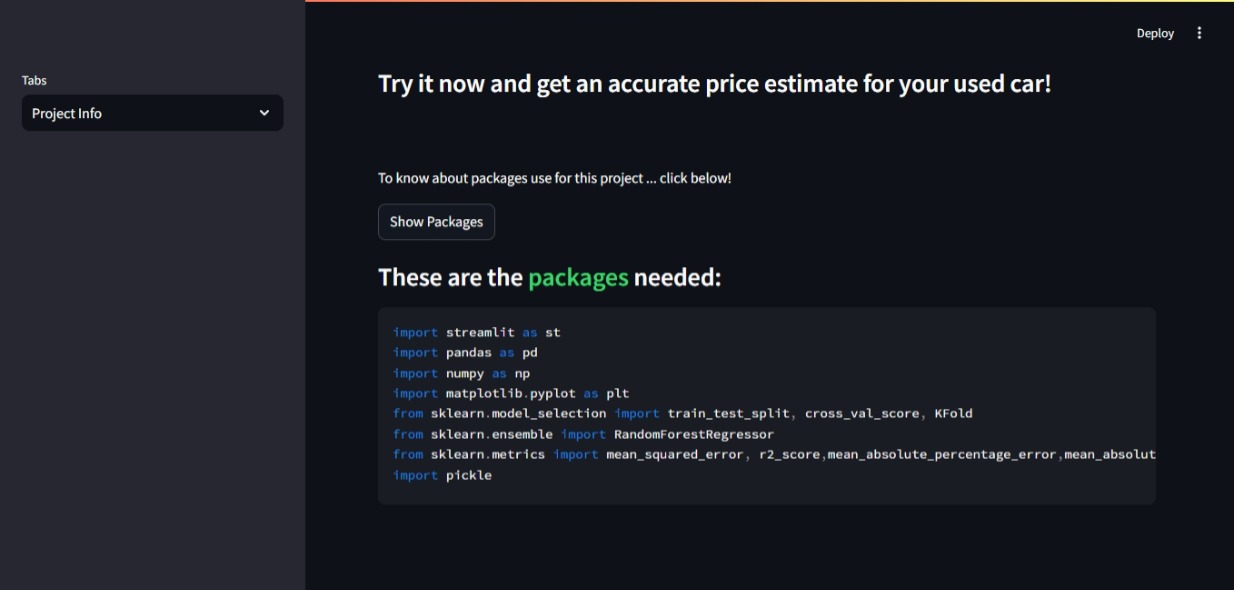
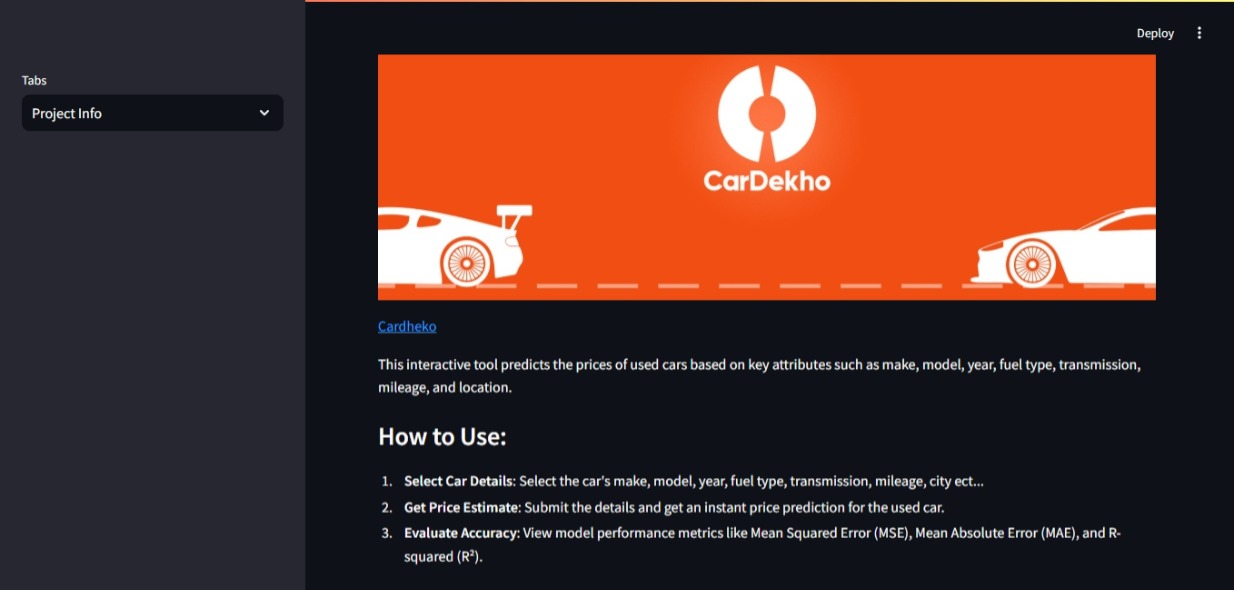
Here is the table for all model on test data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Regression  ML models | Cross-Validation Results (Mean MSE) | Mean Squared Error (MSE) | Mean Absolute Error (MAE) | Mean Absolute Percentage Error (MAPE) | R2 Score |
| Linear Regression | 28.4670 | 23.1907 | 3.0478 | 0.6249 | 0.7186 |
| **Random Forest Regressor** | **7.2499** | **5.3699** | **1.0682** | **0.1466** | **0.9348** |
| Decision Tree Regressor | 13.3378 | 8.8973 | 1.4784 | 0.2008 | 0.8920 |
| Gradient Boosting Regressor | 8.2221 | 6.0488 | 1.2950 | 0.1872 | 0.9266 |

**5.** **Model Deployment:**

* We are using streamlit (web applications) for model deployment and make it more interactive and user friendly.
* using streamlit can allow users to input car features and get real-time predictions.
* More than that we added visualization (scatter plots, bar plots etc..) and all model performance to streamlit.

Here are some images of our streamlit page:



**Results**

The final model achieved a Mean Squared Error (MSE) of 5.36 and an R² score of 0.94, indicating that the model was effective in predicting used car prices with a high level of accuracy.

**Conclusion**

This project focused on predicting used car prices using machine learning. After testing several models, the Random Forest Regressor performed the best, with an R² score of 0.93 on the test set, showing strong accuracy and generalization. Data preprocessing, including handling missing values, encoding categorical features, and normalizing numerical values, played a key role in improving model performance.

The model was deployed as a user-friendly Streamlit web app, allowing anyone to input car details and get price predictions in real-time. This can be a valuable tool for individuals and companies like CarDekho to estimate car prices.

Although the model works well, there’s room for improvement by adding more features or exploring advanced techniques. Overall, this project demonstrates the potential of machine learning to solve real-world problems and provides a solid foundation for future improvements.