Car Dekho –

Used Car Price Prediction

Project Report

Submitted By:

Vikneswari.M

Table of contents	
1.	Summary
2.	Introduction
	Problem Statement
	Objective
	Project scope
3.	Data preprocessing
	Data overview
	Data cleaning and preprocessing
4.	Exploratory Data Analysis (EDA)
	Objective of EDA
	Key Insights
5.	Model Development
	Methodology
	Algorithms i. Linear Regression ii. Decision Tree Regressor iii. Random Forest Regressor iv. XGBoost Regressor (GBR) v. Hyper Parameter Tuning
6.	Model Deployment (streamlit application)
	Overview of streamlit application

1. Summary:

At Car Dekho, we developed a machine learning model to predict used car prices, aimed at enhancing customer experience and improving pricing accuracy. The project involved data preprocessing, exploratory data analysis (EDA), and the application of advanced machine learning algorithms to build a highly accurate model. The result is a Streamlit-based web application that provides users with an intuitive platform for car price predictions, streamlining decision-making and transactions in the used car market.

2. Introduction:

2.1. Problem Statement

Determining the accurate value of used cars is a complex task due to the wide range of variables that influence pricing, such as age, condition, and market demand. Car Dekho aims to overcome this challenge by creating a machine learning model capable of delivering precise price predictions. By integrating this model into an intuitive web application, the solution will simplify the pricing process for both customers and sales teams, improving decision-making and overall efficiency.

2.2. Objective

The main goal is to develop and deploy a machine learning model that can predict used car prices based on various input features, including make, model, year, fuel type, transmission, kilometres driven, seats, engine displacement and more. This model will be integrated into a Streamlit application, offering users quick and accurate price estimates at their fingertips.

2.3. Project scope

- Creation of a machine learning model to predict used car prices.
- Integration of the model into a Streamlit web application for deployment.
- Design of an intuitive interface for easy use by both customers and sales teams.

3. Data preprocessing

3.1. Data overview:

The dataset for this project was sourced from Car Dekho and includes detailed records of used car prices. It features attributes such as make, model, year, fuel type, transmission, kilometres driven, ownership, city, and more. This dataset, originally in an unstructured format, has been converted into a structured format for analysis.

3.2. Data Cleaning and Preprocessing:

- Drop null values and duplicates.

3.2.1 Price Conversion:

- Converted price values from various formats (e.g., "₹ 4 Lakh" and "₹ 1,50,000") into a consistent numeric format by removing non-numeric characters and converting terms like "Lakh" and "Crore" into numeric values.

3.2.2 Feature Engineering:

- **Categorical Features:**
  - Checked and corrected spelling errors in categorical columns.
  - Applied one-hot encoding to certain categorical features like fuel type, body type, manufacturer, model, transmission type.
  - One hot encoding used to avoid Ordinal misinterpretation.
- **Numerical Features:**
  - Cleaned features like kilometre's driven, Owner no, engine displacement, mileage by replacing commas with nothing, removing terms such as "kg," "cc", "kmpl" and "rpm," and eliminating unnecessary characters like '@' and ','.
  - In max power column, convert all units to standard horse power(hp) unit.
  - Converted numerical features to integers where necessary.

3.2.3 Unrelated Columns:

- Dropped columns that were irrelevant to the analysis, such as image URLs, links, and key columns.
- Removed all duplicated columns.

4. Exploratory Data Analysis (EDA):

4.1. Objective of EDA

Exploratory Data Analysis (EDA) is a crucial step in data preprocessing aimed at understanding the underlying patterns, relationships, and structures in a dataset. The goal is to summarize the main characteristics of the data and identify any potential issues or insights that may influence further analysis or model development.

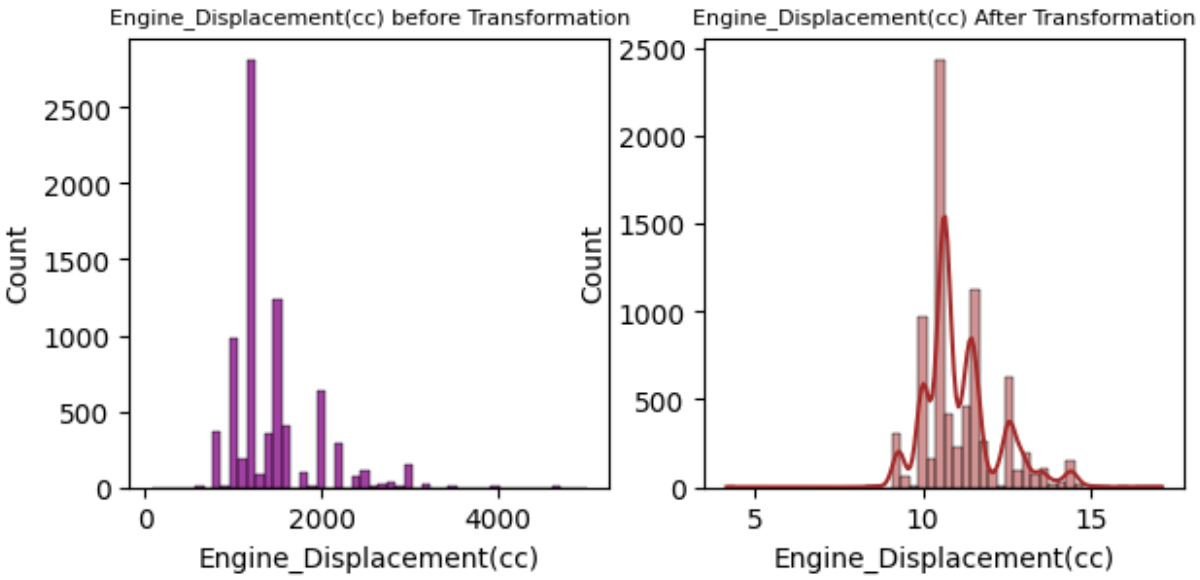
4.2. Key Insights

4.2.1 Check for Null Values:

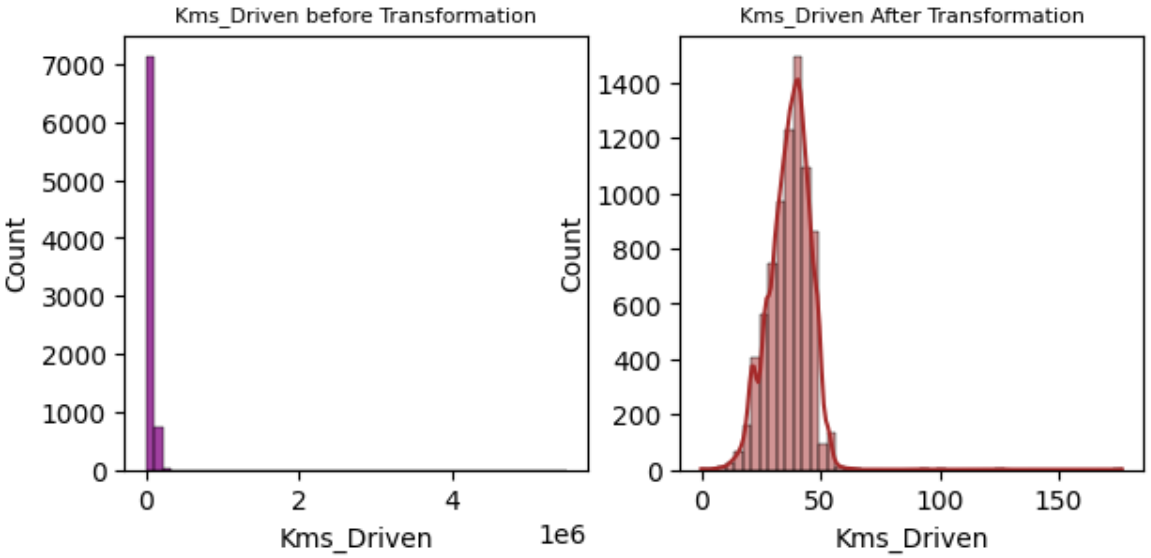
- **Objective:** Identify missing values in the dataset.
- **Methods:** Analyze the extent and patterns of missing data, and choose appropriate strategies for handling them, if the null values count is more than 80 percent drop the null values.

4.2.2. Remove Outliers:

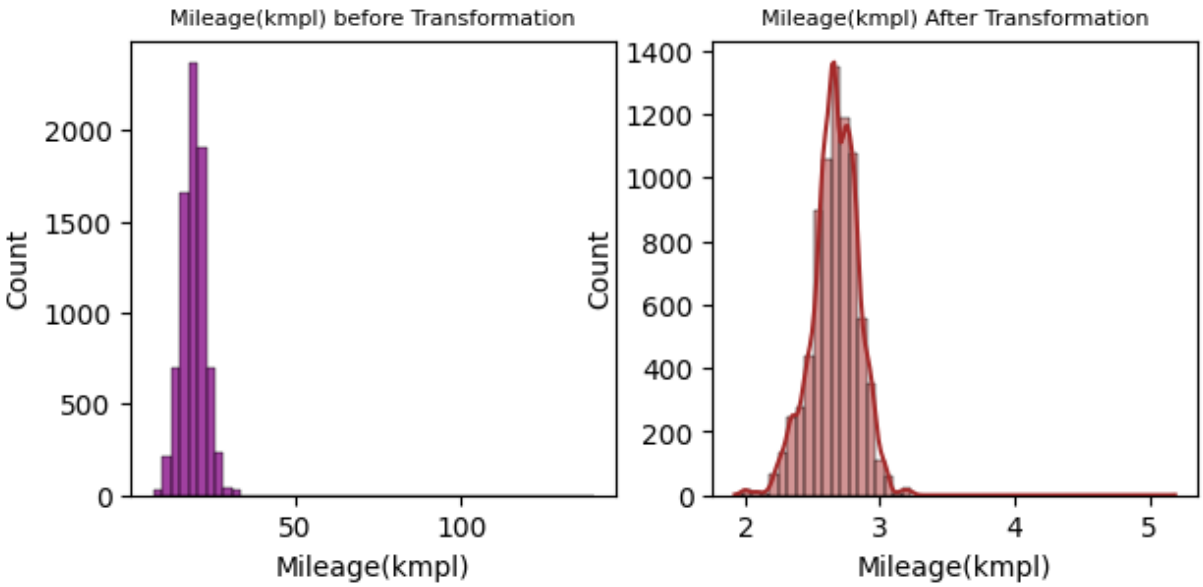
- **Objective:** Detect and handle outliers that may skew the analysis or affect model performance.
- **Methods:** Most of the features were positively skewed. Hence the **Cuberoot transformation** was used to fix the skewness in Data.
- Use statistical methods like the **Interquartile Range (IQR)** to identify and remove outliers Visualization.
- **Univariate Analysis:** Examine the distribution of individual features using histograms, box plots.
- **Visualizations:** using matplotlib and seaborn were used to further understand the data and some insights were derived.
- **Histogram** plots for before and after transformation.
- **Skewness** before and after Transformation



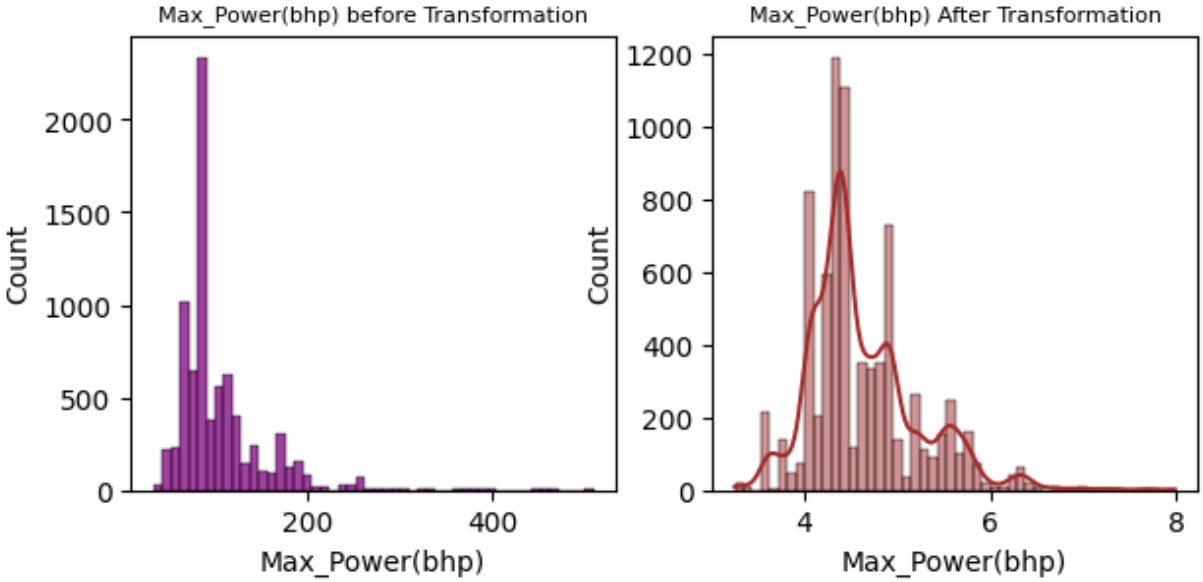
The skewness after transformation is : 1.0087054669261875



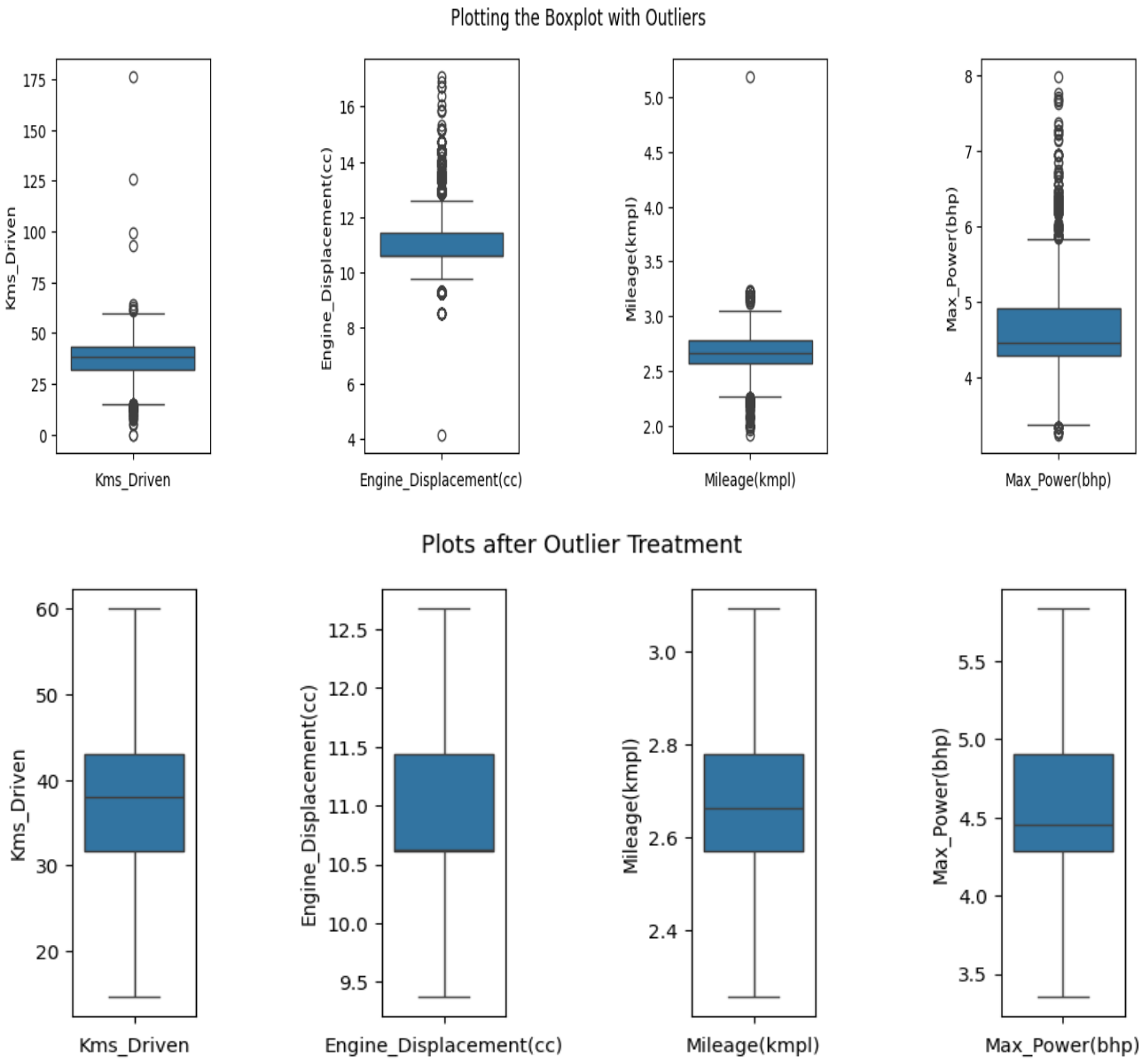
. The skewness after transformation is : 0.37742164180292104



The skewness after transformation is : 0.023430801926596887



The skewness after transformation is : 1.0628802135711266



5. Model Development:

5.1. Train Test Split:

- The data is then split with 20% allocated for testing data that will be used to evaluate the final model's performance after model selection and the rest of the data 80% us allocated for training data that is used to evaluate model performance during training.

5.2. Methodology:

- Different regression models were evaluated, including Linear Regression, Extra Tree Regressor, XG Boost, Decision Tree, and Random Forest, to determine the most accurate and dependable model for predicting used car prices.

5.3. Algorithms:

- In this project, the target variable price is a continuous variable. hence 5 different ML regression algorithms were used for building.

- **Linear Regression:**

  Linear Regression was utilized as the initial model due to its straightforward nature and interpretability.

- **Decision Tree Regressor:**

  Decision Trees were selected for their clear interpretability and ability to model intricate non-linear relationships.

- **Random Forest Regressor**:

  Random Forest, an ensemble technique, was adopted for its high accuracy and resilience.

- **Extra Tree Regressor:** This class implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset

- **XGBoost:** builds an ensemble of decision trees sequentially, where each new tree tries to correct the errors made by the previous ones.

5.4. The Models were Assessed Using The Following Metrics:

- **Mean Squared Error (MSE):** Evaluates the average of the squared differences between the actual and predicted values.
- **Mean Absolute Error (MAE):** Provides an average of the absolute differences between predicted and actual values, offering a straightforward measure of prediction accuracy.
- **R^2 Score:** Reflects the proportion of variance in the dependent

variable that is explained by the independent variables.

HyperParameter Tuning:

Comparitively the Xg boost Regressor Model gave a better R2 score. Hence after Hyper Parameter tuning the model using the Random Search CV, the model's r2 score increased and the MSE decreased to some extent. The model was then saved as a pickle file and was deployed using Streamlit Apllication to predict the Prices.

Results:

Hyperparameter Tuned Xgboost Regressor: Delivered the highest performance, with the top R^2 score and the lowest MSE and MAE, making it the selected model for deployment.

6. Model Deployment (Streamlit Application):

6.1. Overview of the Streamlit Application:

Streamlit is an open-source Python library designed for swiftly building custom web apps tailored for data science and machine learning. Its ease of use and versatility make it an excellent tool for deploying machine learning models as interactive web applications.

6.2. Features of the Application:

- **User Input Interface:** The app offers a user-friendly interface allowing users to enter car details, such as make, model, year, fuel type, transmission, mileage, number of owners. It utilizes drop-down menus and sliders to simplify the input process and minimize errors.
- **Price Prediction:** After collecting user inputs, the app employs the trained hypertune xgboost Regressor model to estimate the car's price. The predicted value is promptly displayed, enhancing user interaction.
- **Visualizations:** The app features visualizations that illustrate the effects of various attributes on car pricing, aiding users in

understanding how different factors influence the price.

6.3. Backend Implementation:

- **Encoder Loading:** The one hot encoder is integrated into the application using the pickle library, ensuring it is available for transform the categorical columns.
- **Model Loading:** The trained hyper Tune xgboost Regressor model is integrated into the application using the pickle library, ensuring it is available for making predictions.
- **Data Preprocessing:** User inputs are processed in the same manner as the training data to maintain consistency and accuracy in the predictions.

6.4. Deployment Process:

The application was hosted on a cloud platform, allowing it to be easily accessed through a web browser by both customers and sales representatives.

7. Reason behind the selection of the model:

7.1. xgboost Regressor:

- **Robustness**: XGBoost is generally considered a robust machine learning algorithm, especially for regression tasks, XGBoost utilizes gradient boosting, which builds trees sequentially and focuses on correcting errors (residuals) from previous trees. This iterative process helps the model learn complex patterns and relationships in the data, making it less susceptible to overfitting than some other algorithms.
- **Accuracy**: The model consistently delivered the most accurate predictions across all metrics (MSE, MAE, R^2).
- **Versatility**: It effectively manages both numerical and categorical data, making it well-suited for the diverse features in this dataset.

8.Conclusion:

8.1. Project Impact

The integration of the predictive model through the Streamlit application revolutionizes the customer experience at Car Dekho by delivering swift and precise price estimates. This advancement not only enhances the decision-making process for both customers and sales representatives but also paves the way for future improvements in predictive analytics.

9.Appendices:

9.1. Model Performance Metrics:

The Hyperparameter Tuned xgboost Regressor model was selected for deployment due to its superior performance, achieving the highest R^2 score and the lowest MSE/MAE, making it the most accurate and reliable model for predictions.

Model	R2 Score	MAE	MSE	RMSE
Hyper parameter Tuning	0.7578	155322.14	532565775673.83	729771.04
Linear Regression	0.6417	240873.76	787797486137.13	887579.56
Extra Tree Regression	0.7211	168249.17	613236970753.69	783094.48
XG Boosting	0.7448	162982.98	561011655578.76	749007.11
Decision Tree Regression	0.7112	217329.85	634926140928.11	796822.52
Random Forest Regression	0.7413	165855.38	568695419628.91	754118.96

References:

- Pandas Documentation
- Data visualization Documentation
- Scikit learn Documentation
- Streamlit Documentation
- **Software:**
- Visual studio code