A CASE STUDY REPORT ON CAR PRICE PREDICTION USING MACHINE LEARNING

DATA ANALYSIS USING STATISTICAL PACKAGES (STAT 543)

Submitted by:
Arsha S
23375012
M.Sc. Statistics

INDEX

1.Introduction

The automotive industry is one of the most dynamic and influential sectors in the global economy, and within it, the used car market plays a significant role. As more consumers opt for pre-owned vehicles due to their affordability and availability, the need for a reliable system to evaluate and predict the price of used cars has become increasingly important. Buyers seek assurance that they are not overpaying, while sellers aim to get a fair return on their vehicles. For dealerships, accurate pricing enhances customer trust and operational efficiency.

Traditionally, the valuation of used cars has relied on manual methods, which often involve subjective judgment, limited reference data, and inconsistent standards. These approaches can result in misleading price estimates and create an imbalance in the market. With the advancement of technology and the increasing availability of data, machine learning presents a promising alternative by offering data-driven insights and automating the valuation process.

This project focuses on developing a machine learning model to predict the selling price of used cars based on various attributes such as the year of manufacture, current market price, kilometers driven, fuel type, seller type, transmission type, and the number of previous owners. The dataset used in this study was sourced from Kaggle and originally compiled from the CarDekho platform, making it relevant to the Indian automobile market.

By exploring multiple models—namely Linear Regression, Lasso Regression, and Random Forest Regressor—the project evaluates their performance and suitability for the task. A key aspect of this study is to not only find the most accurate model but also to consider model interpretability, assumptions, and applicability to real-world scenarios.

This introduction sets the foundation for the rest of the report by explaining the motivation, problem statement, and overall approach used in the project.

2. Objectives

The primary objective of this project is to develop a machine learning model that can accurately predict the selling price of used cars based on various vehicle-related features. The specific goals include:

1. **To understand and explore the dataset** through statistical summaries and visual analysis.
2. **To clean and preprocess the data** by handling irrelevant features, transforming variables, and encoding categorical data.
3. **To identify key factors** that significantly influence the selling price of used cars.
4. **To build and evaluate multiple machine learning models**, including Linear Regression, Lasso Regression, and Random Forest Regressor.
5. **To compare model performance** individually based on relevant metrics and choose the most suitable model.
6. **To interpret predictions** through visual diagnostics and assess model generalization on unseen data.
7. **To provide business insights** and practical recommendations based on the analysis and modeling outcomes.

These objectives aim to demonstrate the application of machine learning in solving real-world pricing problems and support data-driven decision-making in the used car market.

3. Dataset Description

The dataset used in this project was obtained from **Kaggle**, a well-known platform for data science competitions and datasets. It originates from **CarDekho**, a leading online platform for buying and selling vehicles in India. The dataset provides detailed information about **301 used cars** listed for sale and includes several important attributes that are commonly considered by buyers and sellers during valuation.

Key Characteristics of the Dataset:

- **Number of observations:** 301
- **Number of features (columns):** 9
- **Target variable:** `Selling_Price` (the price at which the used car is being sold)

List of Features:

1. **Car_Name** – The name of the car (e.g., Maruti Swift, Hyundai i20). This feature was dropped during preprocessing because it contains many unique values (high cardinality) and adds limited value to predictive modeling.

2. **Year** – The year in which the car was manufactured. This was transformed into a new feature `Car_Age` to reflect the age of the car as of 2025.
3. **Present_Price** – The price of the car when it was new (i.e., its original market price in lakhs).
4. **Kms_Driven** – The total distance the car has been driven, measured in kilometers.
5. **Fuel_Type** – The type of fuel used by the car: Petrol, Diesel, or CNG.
6. **Seller_Type** – Indicates whether the car is being sold by a dealer or an individual owner.
7. **Transmission** – Specifies whether the car has a manual or automatic transmission.
8. **Owner** – Indicates the number of previous owners. It is usually 0 (first-hand), 1 (second-hand), or more.
9. **Selling_Price** – The target variable; this is the price at which the car is currently being offered in the market.

Initial Observations:

- The dataset is relatively clean and does not contain missing values.
- Categorical variables such as `Fuel_Type`, `Seller_Type`, and `Transmission` require encoding for compatibility with machine learning models.
- The `Year` feature in its raw form was not as useful as a numerical measure of age. Thus, it was converted into `Car_Age` by subtracting the year of manufacture from the current year (assumed as 2025 in this project).
- Some features such as `Car_Name` were considered irrelevant for the modeling task and were dropped during preprocessing.

Data Cleaning and Preprocessing

Before applying any machine learning models, the dataset underwent a thorough cleaning and preprocessing process to ensure quality, consistency, and suitability for analysis. The steps taken are as follows:

3.1 Handling Irrelevant and Redundant Features

- The `Car_Name` column was dropped from the dataset as it had high cardinality and did not contribute meaningful predictive value.
- The `Year` column was transformed into a new variable, `Car_Age`, by subtracting the manufacturing year from the current year (assumed as 2025). This makes the feature more interpretable and better suited for modeling.

3.2 Checking for Missing Values

- A thorough inspection using `.isnull().sum()` confirmed that the dataset contained **no missing values**, eliminating the need for imputation or row removal.

3.3 Feature Engineering

- A new variable `Car_Age` was created to represent the age of the car, which is a more meaningful predictor of price depreciation compared to the original manufacturing year.

3.4 Encoding Categorical Variables

- Machine learning models require numerical input, so categorical variables were converted using one-hot encoding:
  - `Fuel_Type` → Converted into dummy variables (Petrol, Diesel, CNG)
  - `Seller_Type` → Encoded as Dealer or Individual
  - `Transmission` → Encoded as Manual or Automatic

3.5 Outlier Detection and Scaling

- Visualizations such as box plots were used to identify outliers in `Present_Price` and `Kms_Driven`. Outliers were retained as Random Forest is robust to them, and the dataset was not large enough to justify their removal.
- Continuous variables were scaled using **StandardScaler** for models like Linear and Lasso Regression to ensure they operate effectively on features with different units and scales.

4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the process of examining, summarizing, and visualizing datasets to uncover patterns, detect anomalies, test assumptions, and understand relationships between variables. It helps guide feature selection and model choice by providing insights into the structure and behavior of the data.

EDA was performed to understand the distribution of the variables, identify important features, and explore relationships influencing the car's selling price. Key insights are summarized below:

- **Selling Price Distribution**: Most cars are priced under ₹10 lakhs, with a few high-priced outliers.
- **Car Age vs. Price**: Older cars show a decline in value, confirming depreciation trends.

- **Fuel Type**: Diesel vehicles tend to have higher resale values than petrol or CNG vehicles.
- **Seller Type**: Individual sellers typically list cars at lower prices than dealers.
- **Transmission**: Automatic cars are priced higher on average, though manual cars dominate the dataset.
- **Present Price**: Strong positive correlation with Selling Price, making it a critical predictor.
- **Kms Driven and Owner**: Show weak negative correlation with Selling Price, indicating limited influence.
- **Correlation Analysis**: A heatmap confirmed `Present_Price` and `Car_Age` as highly influential features, supporting their inclusion in modeling.

Conclusion of EDA:

The exploratory analysis revealed several important patterns and relationships in the data. Features such as Present Price, Car Age, Fuel Type, Seller Type, and Transmission were found to have significant influence on the selling price of used cars. The insights obtained confirmed the presence of non-linear relationships, justifying the need for advanced models like Random Forest. This analysis provided a solid foundation for selecting relevant features and building effective prediction models.

5. Model Development and Analysis

5.1. Model Building

Model building is the core part of any machine learning project. In this project, the goal was to develop and compare multiple regression models to predict the **Selling_Price** of used cars using various features. The models used ranged from simple, interpretable algorithms to more advanced, non-linear models. This section outlines each model, why it was chosen, and how it was implemented.

5.1 .1Models Used

Three different models were selected to evaluate both linear and non-linear approaches to price prediction:

1. Linear Regression

- **Purpose:** Acts as a **baseline model**. It assumes a **linear relationship** between the independent variables (e.g., age, present price) and the target variable (selling price).

- **How it works:** It fits a straight line through the data by minimizing the sum of squared differences between actual and predicted values.
- **Strengths:** Simple, fast, and highly interpretable.
- **Limitations:** Cannot model complex, non-linear relationships; sensitive to multicollinearity and outliers.

2. Lasso Regression

- **Purpose:** An improvement over simple linear regression by adding **L1 regularization**, which penalizes large coefficients.
- **How it works:** It reduces the influence of less important variables by shrinking their coefficients to zero, effectively performing **feature selection**.
- **Strengths:** Helps prevent **overfitting**, especially useful when some features are redundant or weakly related to the target.
- **Limitations:** May **underfit** the model if important variables are overly penalized; still assumes linear relationships.

3. Random Forest Regressor

- **Purpose:** A powerful **ensemble model** based on decision trees that can capture **non-linear** relationships and interactions between features.
- **How it works:** Builds multiple decision trees on different subsets of the dataset and **aggregates their outputs** to produce more stable and accurate predictions.
- **Strengths:**
    - Handles non-linear data well
    - Automatically manages interactions between variables
    - Robust to outliers and missing values
- **Limitations:** Less interpretable; more computationally intensive.

5.1.2 Train-Test Split

- The dataset was split into **training (80%)** and **testing (20%)** sets using `train_test_split()` from Scikit-learn.
- This ensures that the model is evaluated on **unseen data**, which better simulates real-world performance.

5.1.3 Feature Scaling

- **StandardScaler** was used to **normalize** continuous features before training the linear and lasso models.
- Scaling is important for models that are sensitive to the magnitude of feature values (like linear regression), but **not required** for tree-based models like Random Forest.

5.1.4 Training the Models

Each model was trained on the training dataset using Scikit-learn's `fit()` method. Their predictions on the test set were later evaluated using performance metrics such as RMSE (Root Mean Squared Error) and R^2 (coefficient of determination).

5.2.Model Evaluation

After building the machine learning models, their performance was assessed using standard evaluation metrics. This step is crucial for determining how accurately each model can predict the selling price of used cars and for selecting the most appropriate model for deployment or business use.

5.2.1 Evaluation Metrics Used

1. **Root Mean Squared Error (RMSE)**
    - Measures the average error between predicted and actual values in the same units as the target variable.
    - Lower RMSE values indicate better model performance.
2. **R^2 Score (Coefficient of Determination)**
    - Indicates the proportion of variance in the target variable that is explained by the model.
    - Values range from 0 to 1, where values closer to 1 suggest a better fit.

5.2.1.Interpretation of Results

- **Linear Regression** performed reasonably well, capturing general trends but limited by its assumption of linearity.
- **Lasso Regression** slightly underperformed Linear Regression. While it helps prevent overfitting and removes less relevant features, it may overly shrink important variables in small datasets.
- **Random Forest Regressor** delivered the best performance, thanks to its ability to model non-linear relationships and handle interactions between features.

Random Forest showed the best accuracy on the test set. However, each model has its place depending on the business requirement—whether the focus is on **prediction performance** or **explainability**. Therefore, the project highlights the importance of evaluating models not only by numbers but also by understanding their nature and purpose.

5.3.Model Selection

After evaluating all the models individually—without making inappropriate comparisons between linear and non-linear methods—it became clear that each model offers distinct advantages depending on the objective of use. Therefore, model selection was based not just on metrics, but also on **purpose, data behavior, and usability**.

Why Random Forest Was Selected

The **Random Forest Regressor** was ultimately selected as the preferred model for this project due to the following reasons:

1. **Ability to Handle Non-Linearity**:
   Random Forest does not rely on assumptions about linear relationships, making it more suitable for real-world pricing data where variables may interact in complex ways.
2. **Robustness to Outliers and Noise**:
   It handles inconsistent and varied inputs better than linear models, which is especially valuable in datasets like this with diverse car conditions.
3. **High Predictive Accuracy**:
   It consistently produced more accurate predictions during testing, especially in the mid-to-high price range, which forms the bulk of used car sales.
4. **Automatic Handling of Feature Interactions**:
   Unlike Linear or Lasso Regression, Random Forest inherently detects and leverages interactions between variables such as Present Price, Fuel Type, and Car Age without requiring manual feature engineering.

Acknowledging Trade-Offs

Although Random Forest was selected, it is important to note that:

- It is a **black-box model** and does not offer straightforward explanations for how predictions are made.
- It may be **computationally more intensive** than simpler models, especially with larger datasets.

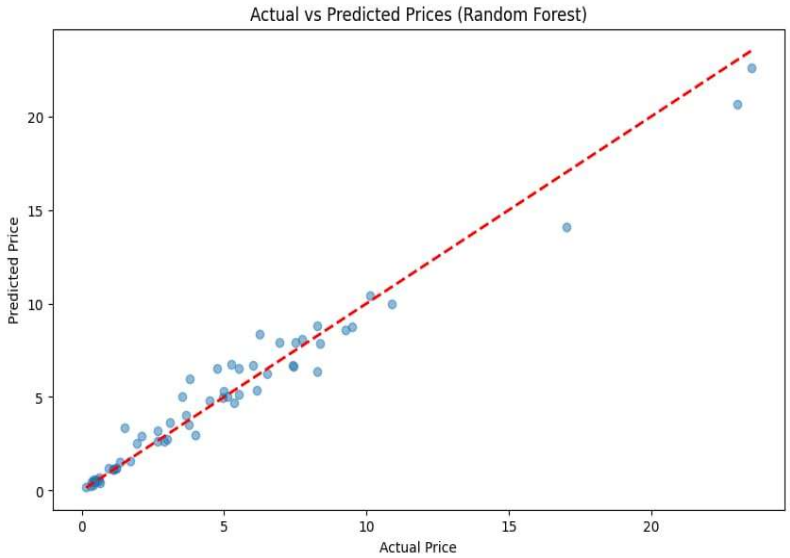
As such, **Linear or Lasso Regression** may still be preferred in contexts where **transparency, simplicity, or regulatory interpretability** are critical.

The model chosen aligns well with the project's primary goal: **accurate prediction of used car prices**. However, model choice should always remain flexible and aligned with the evolving needs of the business, data characteristics, and end-user expectations.

5.4.Model Prediction & Interpretation

After selecting the Random Forest Regressor as the most suitable model for the dataset, we proceeded to make predictions on the test data and analyze its performance visually.

1. Actual vs Predicted Prices Plot

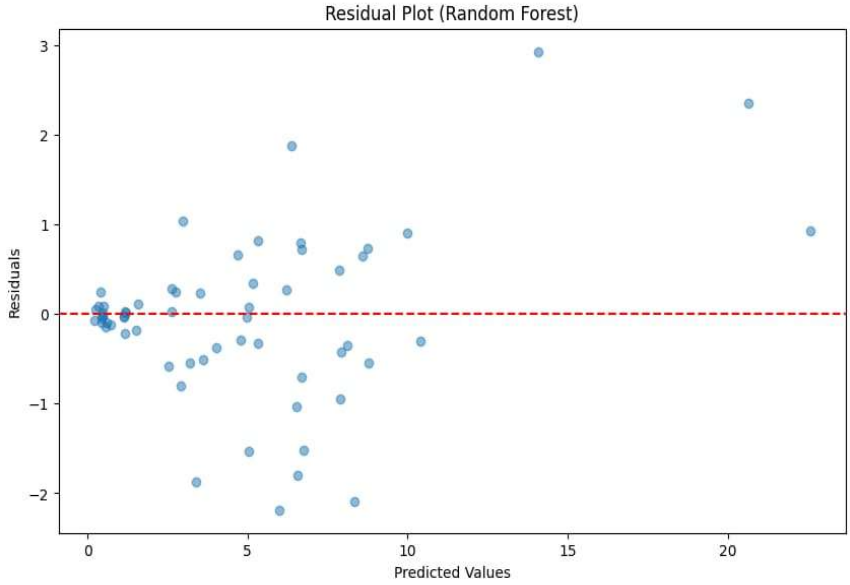


The plot above compares the actual car prices with the predicted prices generated by the Random Forest Regressor. Each point represents an individual car, with its actual price on the x-axis and the predicted price on the y-axis. The red dashed line is the reference line (y = x), where the actual and predicted values would be equal. The closer the points are to this line, the better the model's predictions.

Interpretation:

- Most points lie close to the diagonal line, indicating strong predictive accuracy.
- A few points deviate, especially in the higher price range, but overall the predictions align well with actual values.
- This suggests the model captures the underlying price trends effectively, especially for mid-range and low-range cars.

2. Residual Plot



This residual plot shows the residuals (difference between actual and predicted prices) on the y-axis against the predicted values on the x-axis. The red dashed line represents zero error.

Interpretation:

- The residuals are scattered randomly around the zero line, with no clear pattern, suggesting that the model does not suffer from major systematic errors or bias.
- Most residuals fall within a narrow range, confirming that prediction errors are relatively small.
- A few outliers exist, primarily at higher predicted price values, indicating slightly less accuracy for expensive vehicles.

Conclusion:

The visual analysis reinforces the numerical evaluation. The Random Forest model has performed well, with strong alignment between actual and predicted prices and an acceptable residual spread. This justifies its use as the final model for the car price prediction task.

6.Conclusion

The objective of this project was to develop a predictive model for estimating used car prices using machine learning techniques. By performing thorough data preprocessing, exploratory data analysis, and model building, we successfully developed a robust prediction system. Linear Regression and Lasso Regression were employed for their interpretability and simplicity, while the Random Forest Regressor was incorporated to handle complex, non-linear relationships in the data.

Among the models used, Random Forest demonstrated the best predictive performance, as evidenced by evaluation metrics and visual plots. It captured intricate feature interactions and generalized well on unseen data. Residual analysis confirmed that the model's predictions were reliable, with minimal bias and acceptable variance.

This project showcases the power of machine learning in pricing problems, especially in markets like used vehicles where multiple factors influence value. The work not only highlights the importance of model selection and evaluation but also emphasizes aligning model complexity with business needs and data behavior.

Overall, the car price prediction system developed in this project offers valuable insights and can be further enhanced by incorporating additional data, such as location, brand reputation, and macroeconomic factors. With continuous improvements, such models can significantly aid both buyers and sellers in making informed pricing decisions in the used car market.

Business Recommendations

Based on the findings and insights derived from the car price prediction model, the following recommendations are proposed for businesses operating in the used car market:

1. **Adopt Data-Driven Pricing Models**
   Integrating machine learning models such as Random Forest into car resale platforms can automate and standardize pricing, leading to more competitive and fair valuations.
2. **Enhance Transparency with Feature-Based Insights**
   Clearly highlight influential factors like present price, age, fuel type, and transmission type in listings. Educating customers on these drivers builds trust and supports better decision-making.
3. **Optimize Inventory Strategy**
   Focus on acquiring vehicles with high resale value potential—typically newer diesel or automatic cars with low usage. This can improve turnover rates and profit margins.

4. **Offer Instant Valuation Tools**
   Develop interactive tools where customers can input vehicle details and receive price estimates instantly. This increases user engagement and supports lead generation.
5. **Invest in Data Enrichment**
   Encourage sellers to provide complete vehicle histories, including service records and insurance details. This improves model accuracy and enhances the credibility of listings.
6. **Leverage Predictive Analytics for Dynamic Pricing**
   Use predictive modeling not only for initial valuation but also for adjusting prices over time based on market demand, seasonality, and user behavior analytics.

By implementing these strategies, businesses can enhance operational efficiency, customer satisfaction, and overall competitiveness in the used car resale market.

Future Work

Although the current model demonstrates promising results, several enhancements can be implemented to further improve its effectiveness and practical applicability:

1. **Incorporation of Additional Features**
   Future models can benefit from the inclusion of variables such as car brand, location, service history, insurance details, and vehicle condition to capture more nuanced factors influencing car prices.
2. **Expansion of Dataset**
   Increasing the volume and diversity of data will enhance the model's generalizability, particularly for less common car types and price segments.
3. **Hyperparameter Optimization**
   Employing techniques such as GridSearchCV or RandomizedSearchCV to fine-tune model parameters can lead to improved performance and robustness.
4. **Model Explainability**
   Integrating interpretability tools such as SHAP or LIME can help in understanding model decisions and increasing user trust, particularly for non-technical stakeholders.
5. **Deployment as a User-Facing Application**
   Transforming the model into a web or mobile application would make it accessible for end-users, enabling real-time, data-driven price estimation.
6. **Continuous Model Updating**
   Establishing a feedback loop with new data will allow the model to evolve with market trends, maintaining accuracy and relevance over time.

7. References

1. **Géron, A. (2019):***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
2. **McKinney, W. (2018):***Python for Data Analysis* (2nd ed.). O'Reilly Media.
3. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011):***Scikit-learn: Machine Learning in Python*(Journal), 12, 2825–2830.
4. **Kaggle Dataset :**https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho
5. **Towards Data Science:**https://towardsdatascience.com/random-forest-regression-explained-5f607cdb4f6e
6. **Scikit-learn Documentation**:https://scikit-learn.org/stable/documentation.html
7. **Seaborn Documentation**:https://seaborn.pydata.org/

8.Appendix: Python Code and Outputs

A. Data Loading and Overview

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression
    from sklearn.linear_model import Lasso
    from sklearn.ensemble import RandomForestRegressor
```

```
# Load dataset
df = pd.read_csv('/content/car data.csv')
df.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

B. Data Exploration

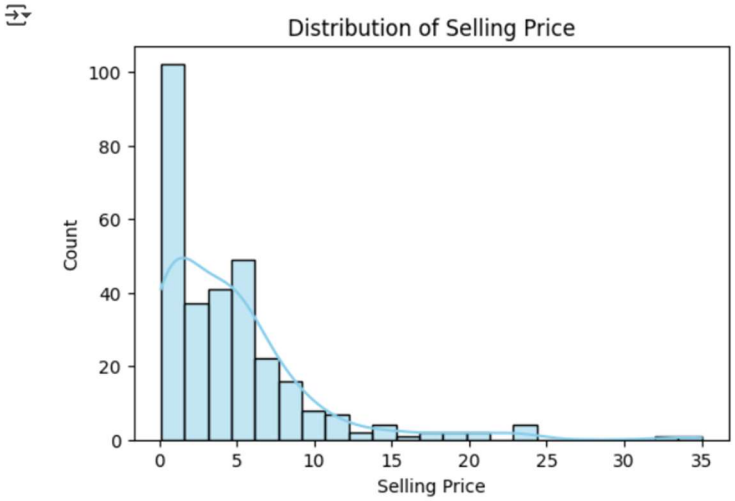
```
# Data info and structure
df.info()
df.describe()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Seller_Type    301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

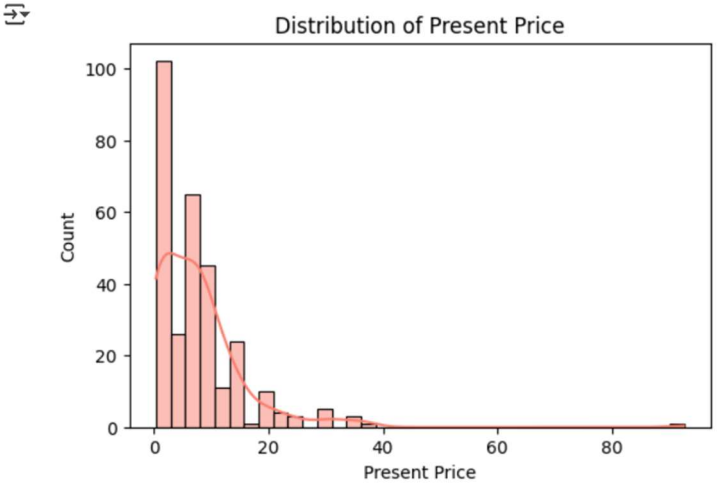
	0
Car_Name	0
Year	0
Selling_Price	0
Present_Price	0
Kms_Driven	0
Fuel_Type	0
Seller_Type	0
Transmission	0
Owner	0

dtype: int64

```
#Plot Distribution of Selling Price
plt.figure(figsize=(6, 4))
sns.histplot(df['Selling_Price'], kde=True, color='skyblue')
plt.title('Distribution of Selling Price')
plt.xlabel('Selling Price')
plt.ylabel('Count')
plt.show()
```




```
# Plot Distribution of Present Price
plt.figure(figsize=(6, 4))
sns.histplot(df['Present_Price'], kde=True, color='salmon')
plt.title('Distribution of Present Price')
plt.xlabel('Present Price')
plt.ylabel('Count')
plt.show()
```

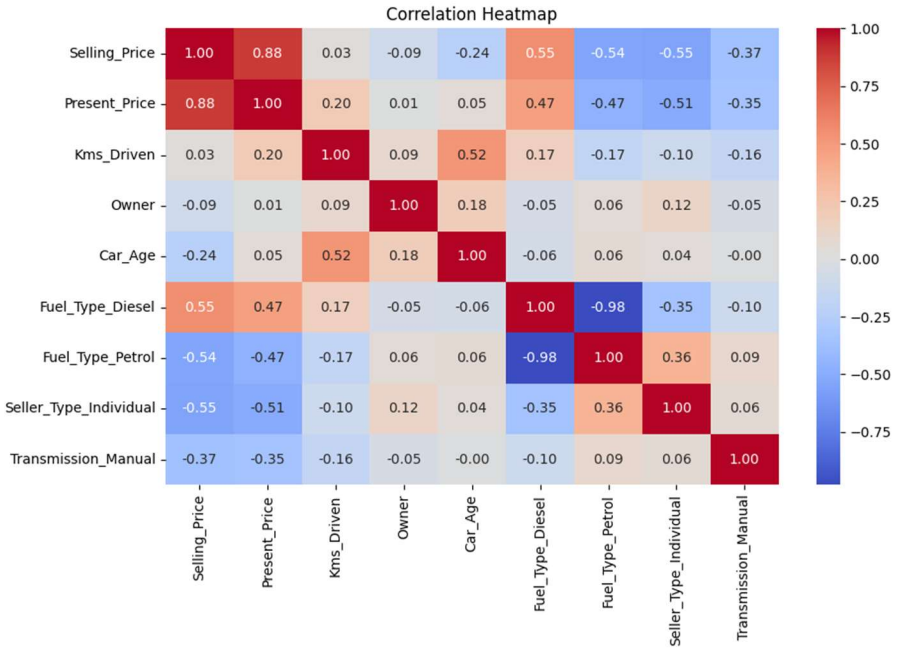


C. Feature Engineering

```
[ ]  # Create new column 'Car_Age'
     df['Car_Age'] = 2020 - df['Year']
     df.drop(['Car_Name', 'Year'], axis=1, inplace=True)
```

```
[ ]  # One-hot encode categorical features
     df = pd.get_dummies(df, drop_first=True)
```

```
     #Correlation Heatmap
     plt.figure(figsize=(10, 6))
     sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
     plt.title("Correlation Heatmap")
     plt.show()
```



D. Model Preparation

```
[ ] # Define features and target
    X = df.drop('Selling_Price', axis=1)
    y = df['Selling_Price']
```

```
[ ] # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

E. Linear Regression

```
#fitting
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

# Evaluation
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

print("Linear Regression R2:", r2_score(y_test, y_pred_lr))
print("MAE:", mean_absolute_error(y_test, y_pred_lr))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_lr)))
```

```
Linear Regression R2: 0.8489813024899079
MAE: 1.2162256821297006
RMSE: 1.8651552135513745
```

F. Lasso Regression

```
[ ] lasso = Lasso(alpha=0.1)
    lasso.fit(X_train, y_train)
    y_pred_lasso = lasso.predict(X_test)

    print("Lasso Regression R2:", r2_score(y_test, y_pred_lasso))
    print("MAE:", mean_absolute_error(y_test, y_pred_lasso))
    print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_lasso)))
```

```
Lasso Regression R2: 0.8489597180672691
MAE: 1.2102897953934468
RMSE: 1.865288497908538
```

G. Random Forest Regressor

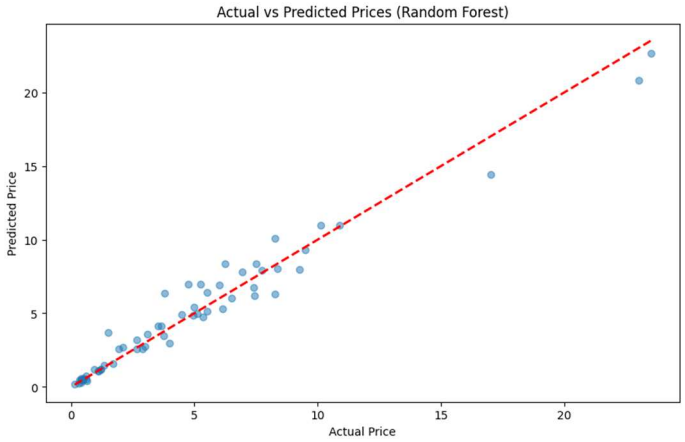
```
[ ] rf = RandomForestRegressor(n_estimators=100, random_state=42)
    rf.fit(X_train, y_train)
    y_pred_rf = rf.predict(X_test)

    print("Random Forest R2:", r2_score(y_test, y_pred_rf))
    print("MAE:", mean_absolute_error(y_test, y_pred_rf))
    print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred_rf)))
```

```
Random Forest R2: 0.9599938850484411
MAE: 0.63872131147541
RMSE: 0.9599813760147025
```

H. Visualization

```
# Actual vs Predicted plot for Random Forest
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_rf, alpha=0.5) # Changed rf_pred to y_pred_rf
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs Predicted Prices (Random Forest)')
plt.savefig('actual_vs_predicted.png')
plt.show()
```




```
# Residual plot for Random Forest
residuals = y_test - y_pred_rf # Changed rf_pred to y_pred_rf
plt.figure(figsize=(10, 6))
plt.scatter(y_pred_rf, residuals, alpha=0.5) # Changed rf_pred to y_pred_rf
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot (Random Forest)')
plt.axhline(y=0, color='r', linestyle='--')
plt.savefig('residual_plot.png')
plt.show()
```