Car Price Prediction - Model Training and Evaluation Report

Author: Jonathan Ndayisenga
Date: April 2025
Toolkits: Python, Scikit-learn, Pandas, Seaborn, Matplotlib
Modeling Techniques: Linear Regression, Random Forest Regression

---

1. Dataset Overview

- **Source**: `car data.csv` (301 rows, 9 columns) from kaggle https://www.kaggle.com/datasets/vijayaadithyanvg/car-price-predictionused-cars
- **Objective**: Predict the **Selling Price** of used cars based on several attributes.

Key Features:

Feature	Description
Car_Name	Name of the car (used to extract brand)
Year	Year of manufacture
Present_Price	Current ex-showroom price (in lakhs)
Driven_kms	Distance driven (in kilometers)
Fuel_Type	Type of fuel (Petrol/Diesel/CNG)

Selling_type	Individual or Dealer
Transmission	Manual or Automatic
Owner	Number of previous owners

---

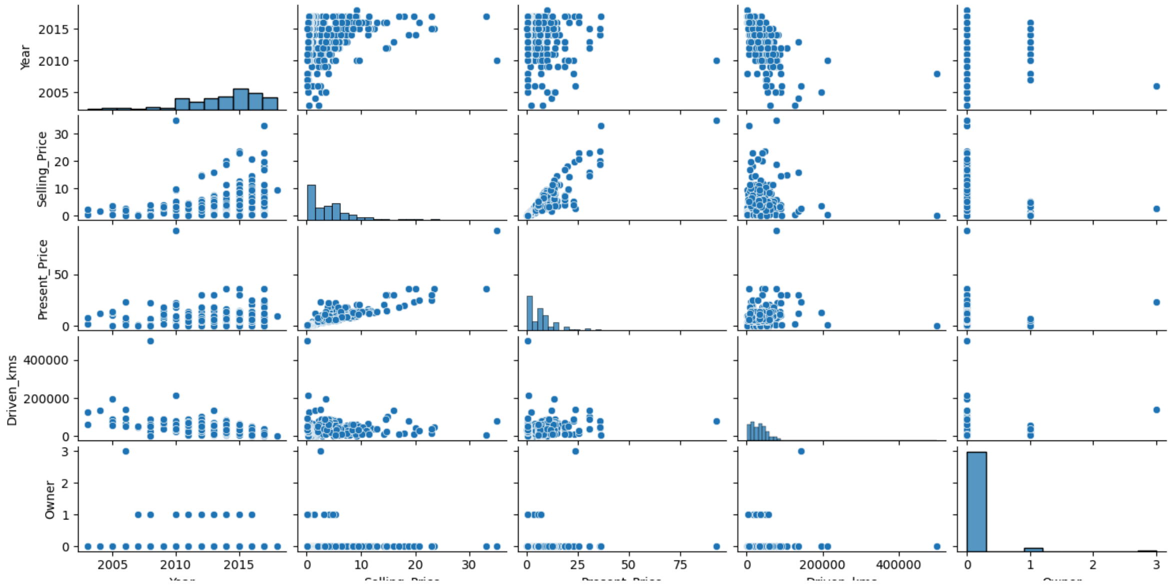
2. Data Preprocessing

- Dropped `Car_Name` after extracting the **brand**.
- Removed null values (dataset had none).
- Converted categorical features to numeric using one-hot encoding (`pd.get_dummies()`).
- Split dataset into **training (80%)** and **testing (20%)** subsets using `train_test_split`.

---

3. Exploratory Data Analysis (EDA)

- **Pairplot** to visualize relationships among numeric features.
- **Heatmap** of correlations:
  - `Present_Price` and `Selling_Price`: Strong positive correlation.
  - `Year` and `Selling_Price`: Mild positive correlation.
- Feature distributions were explored to understand the spread and potential outliers.



Notable insights:

1. **Present_Price vs Selling_Price**:
   - Clear **positive linear relationship** — cars that cost more originally tend to sell for more.
2. **Year vs Selling_Price**:
   - Newer cars tend to sell for higher prices.
3. **Driven_Kms vs Selling_Price**:
   - Slight negative correlation — cars driven more tend to have a slightly lower selling price.
4. **Owner vs Selling_Price**:
   - Higher number of owners may negatively impact selling price, though the pattern is sparse.

4. Model Training & Evaluation

A. Linear Regression (Baseline Model)

Metric	Value
MAE	1.221
MSE	3.459
RMSE	1.860
R^2 Score	0.850

Interpretation: The model explains ~85% of the variance in selling prices. Decent, but might underfit in complex scenarios.

B. Random Forest Regression (Main Model)

Metric	Value
MAE	0.679
MSE	1.007
RMSE	1.003
R^2 Score	0.962

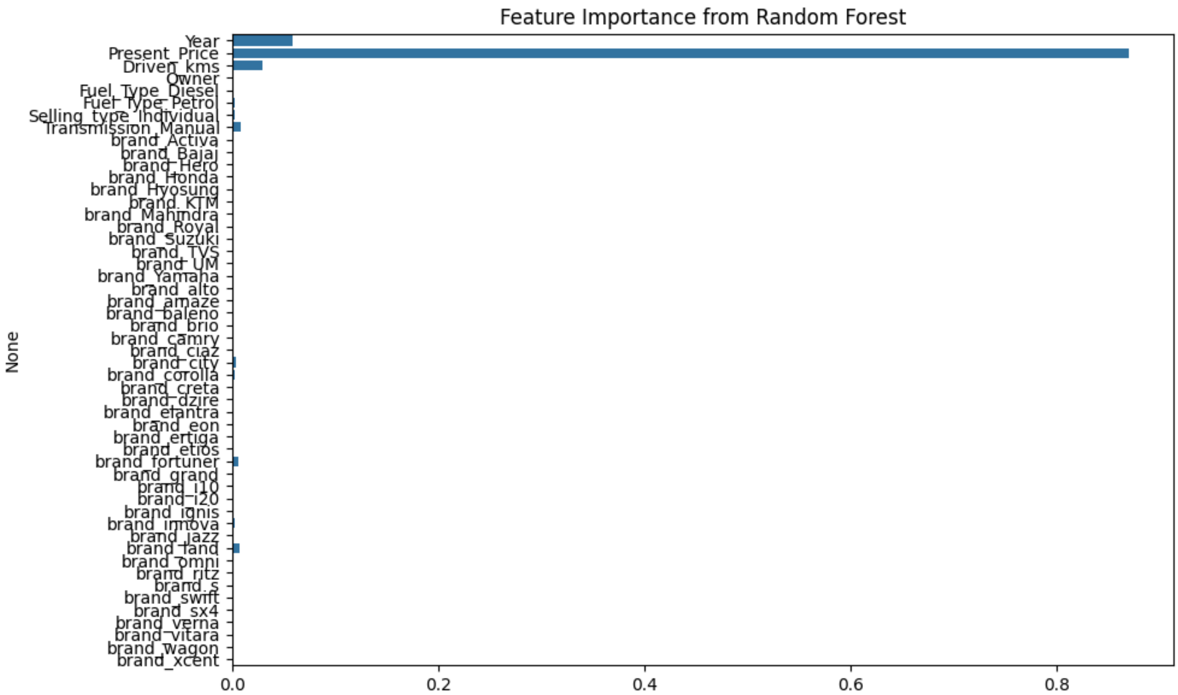
Interpretation: The Random Forest model significantly outperforms Linear Regression, explaining over **96%** of the variance. It captures non-linear relationships effectively.

5. Feature Importance (Random Forest)

Top features contributing to price prediction:

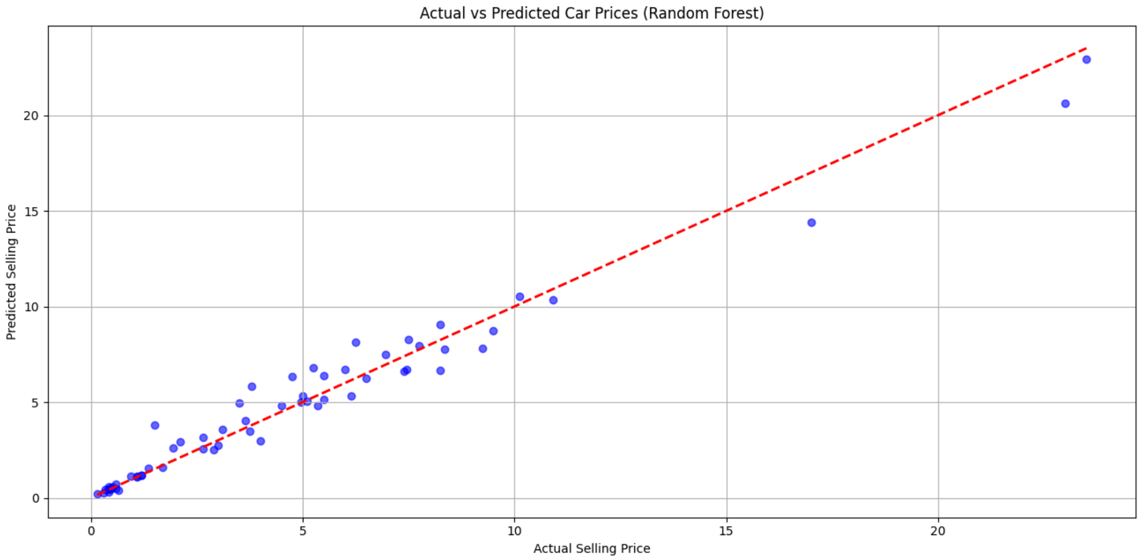
1. **Present_Price**
2. **Year**
3. **Driven_kms**
4. **Fuel_Type_Diesel**
5. **Transmission_Manual**

Visualized using a Seaborn bar plot of feature importances from the trained Random Forest model.



6. Actual vs Predicted Plot

A scatter plot comparing true vs predicted values for the test set shows strong alignment around the diagonal, indicating accurate predictions.



Blue dots: each one represents a car from your test set (actual vs predicted).

Red dashed line: ideal prediction line (where predicted = actual).

---

7. Conclusion

- **Best Model**: Random Forest Regressor
- **Accuracy**: ~96% R^2 on test data
- **Next Steps**:
  - Save the trained model using `pickle` for deployment.
  - Deploy a Streamlit web app for interactive predictions.
  - Host code and model on GitHub for collaboration and deployment.