CAR PRICE PREDICTION

Using Machine Learning

ARSHA S
23375012

CONTENTS

INTRODUCTION

In the used car market, pricing is often inconsistent and based on personal judgment. This leads to confusion, unfair deals, and lack of trust between buyers and sellers. To solve this, our project uses machine learning to predict the fair selling price of a used car based on important factors like age, kilometers driven, fuel type, transmission, and ownership.We used a real-world dataset containing car listings from the Indian market. This makes our model both practical and relevant.

Goal:

To build a reliable, data-driven pricing model that improves transparency and helps users make informed decisions.

Tools & Technologies

- Programming Language: Python
- Libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- Models Used: Linear Regression, Lasso Regression, Random Forest Regressor
- Evaluation Metrics: MSE, RMSE, R^2

OBJECTIVES

- To analyze and clean used car sales data for modeling
- To identify key features that influence car prices
- To compare different regression models (Linear, Lasso, Random Forest)
- To build a machine learning model for accurate price prediction
- To evaluate model performance using appropriate metrics
- To generate price predictions for new/unseen data

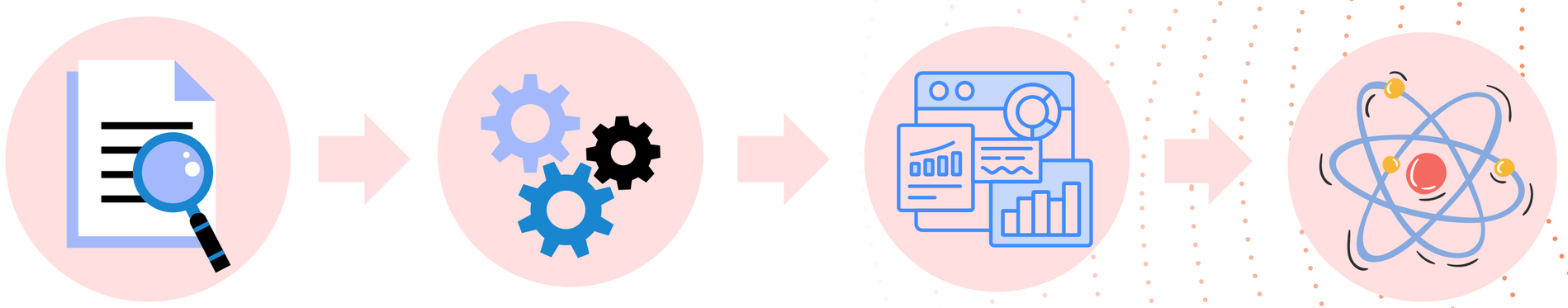
THE DATA

- Source: Secondary data of Indian used cars from Kaggle
- Total Records: 301
- Total Features: 9 columns (including target)
- Target Variable: Selling Price
- Key Features:
  - *Year* – Manufacturing year of the car
  - *Present_Price* – Current market value of car
  - *Kms_Driven* – Distance the car has been driven
  - *Fuel_Type* – Petrol, Diesel, or CNG
  - *Seller_Type* – Individual or Dealer
  - *Transmission* – Manual or Automatic
  - *Owner* – Number of previous owners
  - *Car_Name* – Name/brand of the car (dropped during cleaning)

DATASET

	A	B	C	D	E	F	G	H	I
1	Car_Name	Year	Selling_Pri	Present_Pı	Kms_Drive	Fuel_Type	Seller_Type	Transmissi	Owner
2	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
3	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
4	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
5	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
6	swift	2014	4.6	6.87	42450	Diesel	Dealer	Manual	0
7	vitara brez	2018	9.25	9.83	2071	Diesel	Dealer	Manual	0
8	ciaz	2015	6.75	8.12	18796	Petrol	Dealer	Manual	0
9	s cross	2015	6.5	8.61	33429	Diesel	Dealer	Manual	0
10	ciaz	2016	8.75	8.89	20273	Diesel	Dealer	Manual	0
11	ciaz	2015	7.45	8.92	42367	Diesel	Dealer	Manual	0
12	alto 800	2017	2.85	3.6	2135	Petrol	Dealer	Manual	0
13	ciaz	2015	6.85	10.38	51000	Diesel	Dealer	Manual	0
14	ciaz	2015	7.5	9.94	15000	Petrol	Dealer	Automatic	0
15	ertiga	2015	6.1	7.71	26000	Petrol	Dealer	Manual	0

APPROACH AND METHODOLOGY



Understand & Explore Data

Data preprocessing

Model Building and Evaluation

Make predictions

Note: EDA is done early in the analysis process, starting right after loading the data. Initial EDA helps identify issues like missing values, outliers, and data types. After cleaning the data, we perform deeper EDA to explore distributions, relationships, and patterns. So, EDA is both a starting point and an ongoing step that guides data preparation and modeling.

DATA CLEANING

1. **Dropped Unnecessary Columns**
   - Removed *Car_Name* (since it is not valid for modeling)
2. **Created New Feature**
   - Converted Year to car_dataset["Age"] = 2025 - car_dataset["Year"]
   - More meaningful than the raw year of manufacture
3. **Checked for Missing Values**
   - No missing values in the dataset- confirmed with **.isnull().sum()**
4. **Handled Categorical Variables**
   - Applied One-Hot Encoding on Fuel_Type (Petrol/Diesel/CNG),Seller_Type (Dealer/Individual) and Transmission (Manual/Automatic)

```
#one hot encoding
car_dataset = pd.get_dummies(car_dataset, columns=['Fuel_Type', 'Seller_Type', 'Transmission'], drop_first=True)
```

5. **Checked for Outliers**
   - Observed outliers in *Present_Price* and *Kms_Driven* using boxplot.
   - Log transformation is used for linear & lasso models to make it more linear.
   - Random Forest handles them well, so no removal was needed

KEY INSIGHTS FROM EDA

1. **Price Distribution**: Most cars are sold under ₹10 lakhs, and a few high-priced ones indicate luxury vehicles

2. **Age Impact on Price**: Selling price decreases with car age, and Noticeable drop in price after ~5 years.

3. **Fuel Type Trends:** Petrol cars dominate the dataset, and Diesel cars generally retain more resale value.

4. **Seller & Transmission Patterns**: Most cars are sold by dealers, and manual transmissions are more common than automatics.

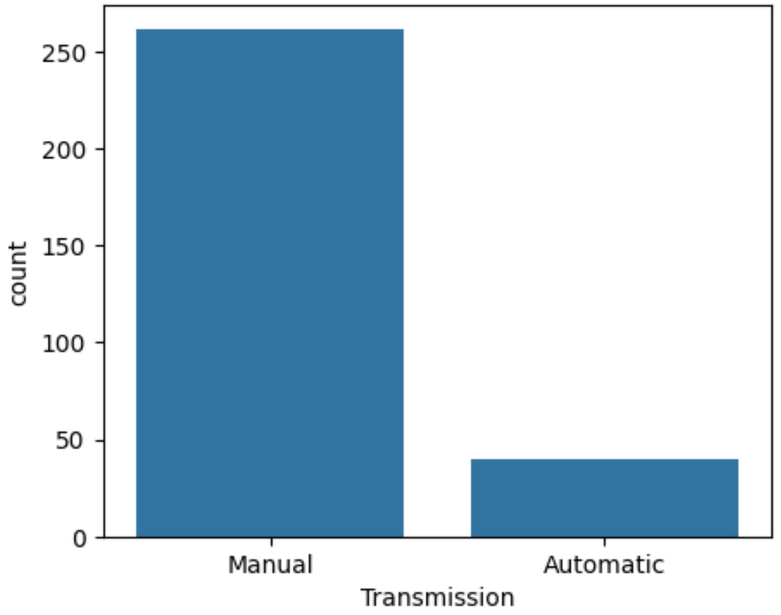
5. **Present Price Correlation:** Strong positive correlation with Selling Price, and Cars with higher original prices hold better value.

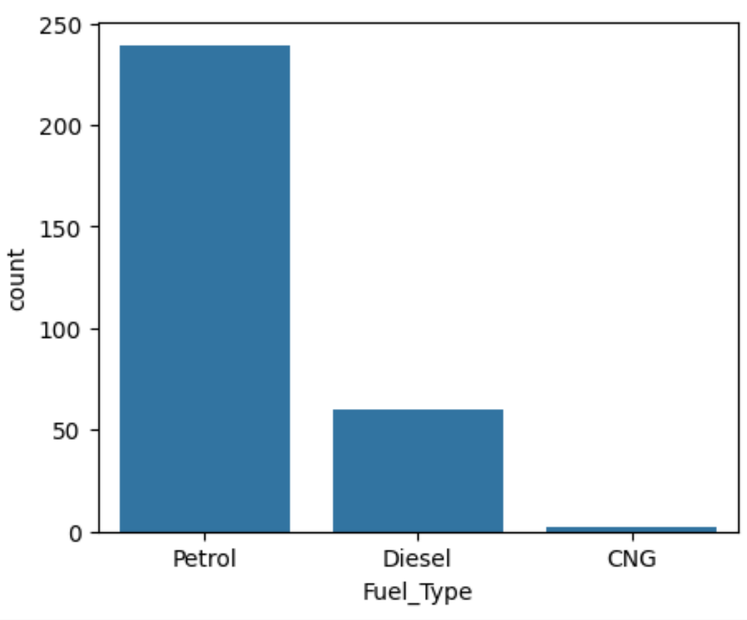
6. **No Missing Values**: Dataset is complete and ready for modeling.

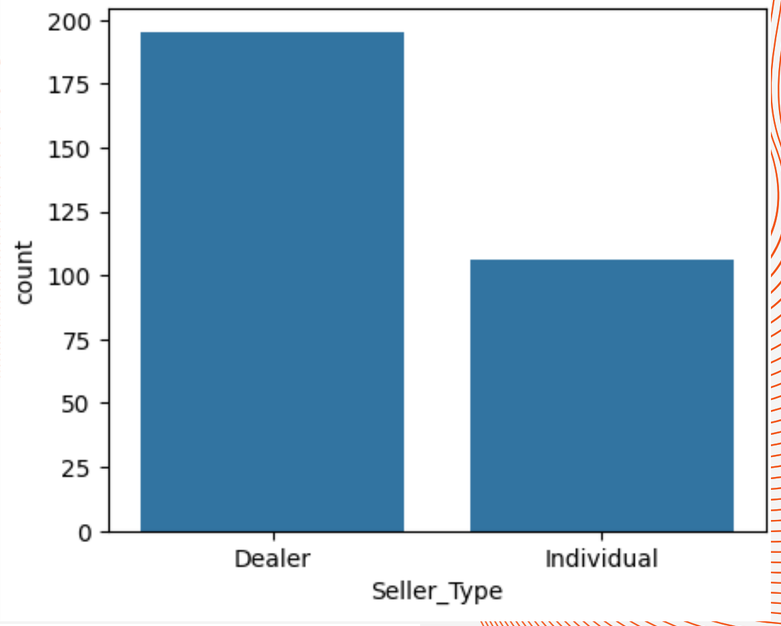
7. **Outliers Detected**: Found in Present_Price and Kms_Driven and Retained for modeling as Random Forest handles them well.

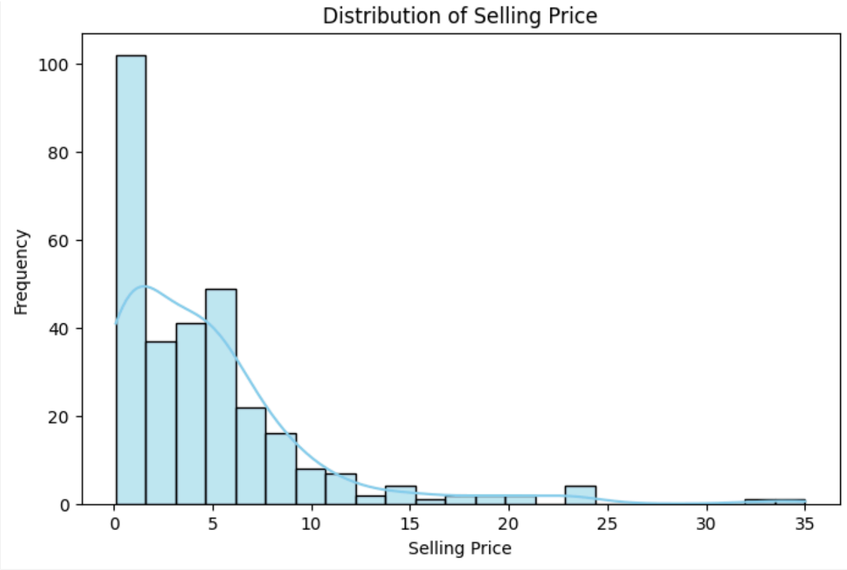
8. **No Multicollinearity**: Features are not strongly correlated — suitable for regression analysis.

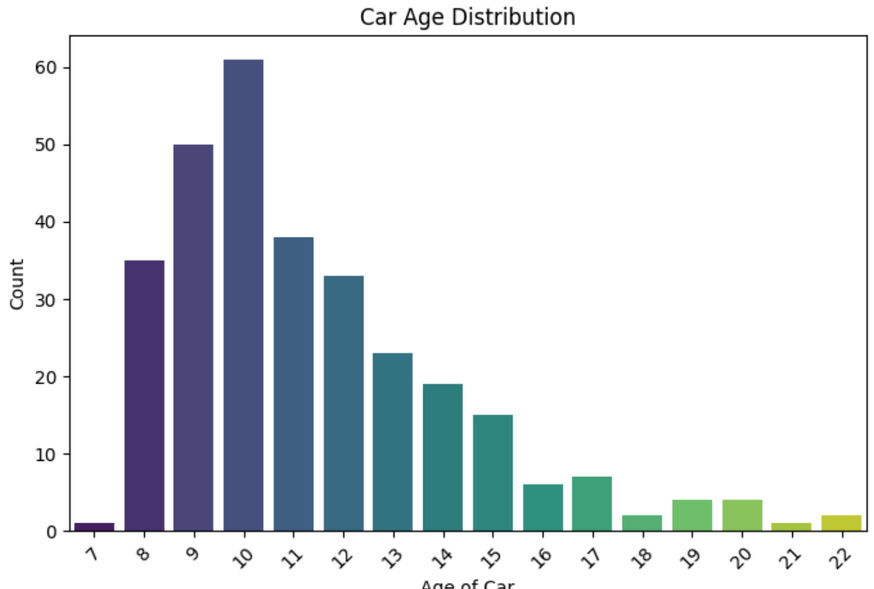
MAIN PLOTS OF EDA







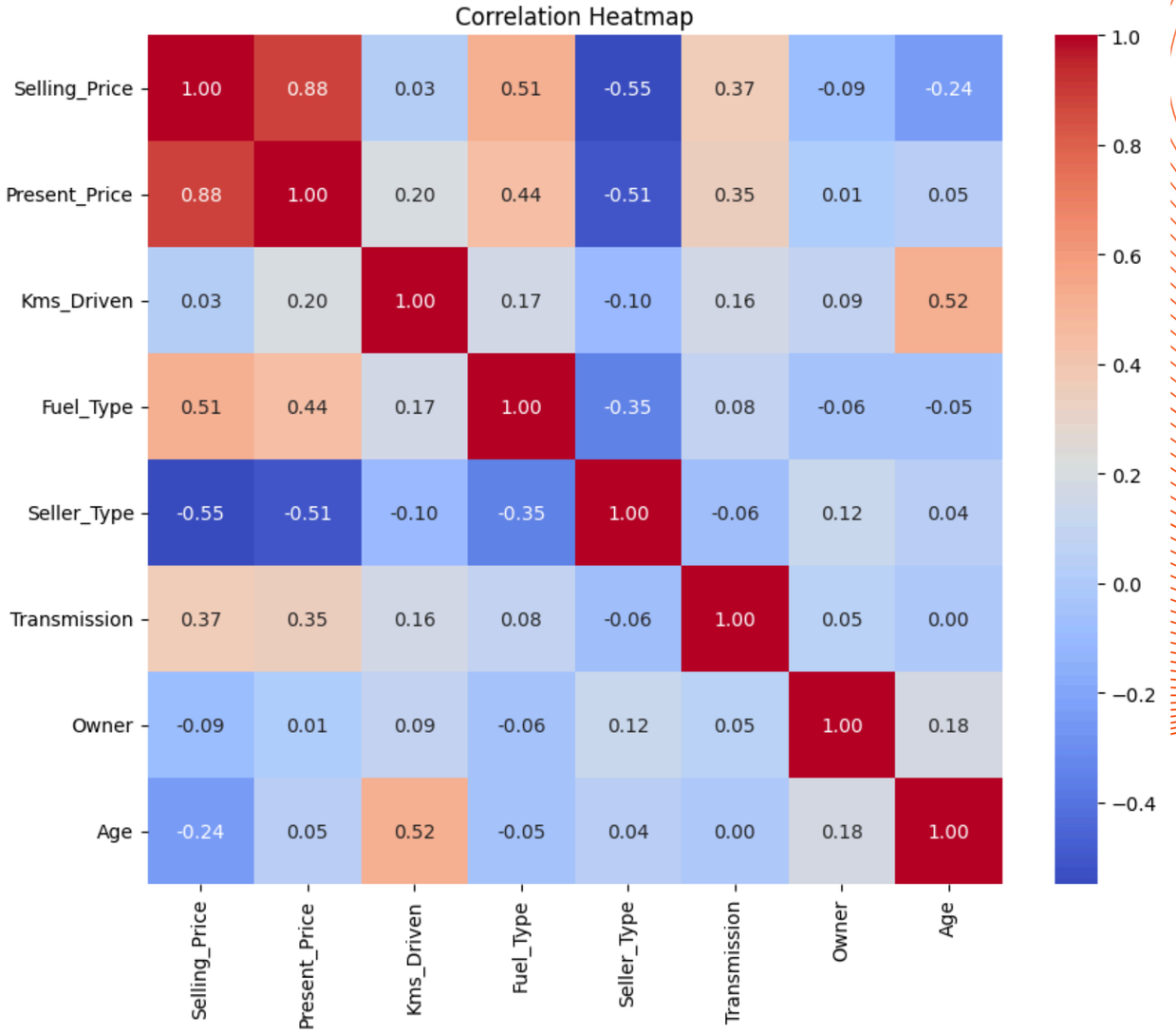




FEATURE CORRELATION

- Selling_Price is strongly positively correlated with:
  - Present_Price (0.88) → most important predictor
  - Fuel_Type (0.51) → diesel cars may sell for more
  - Transmission (0.37) → manual/automatic may impact price
- Negative correlation with:
  - Seller_Type (-0.55) → individual sellers tend to price lower
  - Age (-0.24) → older cars sell for less
- Very weak or no correlation with:
  - Kms_Driven, Owner — minimal impact on selling price

This confirms that **Present_Price, Fuel_Type, and Age** are key features for predicting car prices.



MODEL BUILDING

Models Used:

- Linear Regression – Simple, interpretable baseline model
- Lasso Regression – Linear model with feature selection (L1 regularization)
- Random Forest Regressor – Ensemble model for better accuracy and handling non-linearity

Steps Followed:

1. Train-Test Split
   80% for training, 20% for testing

2.Feature Scaling
   Standardized features using StandardScaler() for Linear and Lasso

3.Model Training
   Fit each model on training data

4.Model Evaluation
   Evaluated using MSE, RMSE, and R^2 Score

LINEAR REGRESSION

- Fits a straight line using all available features
- Assumes all features contribute to the prediction
- Prone to overfitting when there are many or irrelevant features
- No built-in feature selection
- Simpler, more interpretable when the number of features is low

LASSO REGRESSION

- Adds L1 regularization (penalty) to reduce model complexity
- Automatically selects important features by shrinking less important ones to zero
- Helps prevent overfitting in high-dimensional data
- More robust when some features are irrelevant or highly correlated
- Useful when feature selection is desired

RANDOM FOREST REGRESSOR

- Ensemble learning method based on decision trees
- Combines predictions of multiple trees for better accuracy
- Each tree is trained on a random subset of data and features

Why It’s Used:

- Captures non-linear relationships in data
- Handles outliers and missing values better than linear models
- Less prone to overfitting than a single decision tree
- Works well with both numerical and categorical variables

Benefits in Our Project:

- Provided highest accuracy among all models used
- Handled outliers in Present_Price and Kms_Driven without preprocessing
- Performed well even without feature scaling

MODEL EVALUATION

Purpose:

- To assess how well the model performs on unseen data
- Helps compare models and select the most effective one

Metrics Used:

MSE (Mean Squared Error):

- Measures average squared difference between predicted and actual values

(Lower is better)

RMSE (Root Mean Squared Error):

- Square root of MSE, easier to interpret in original units(Lower is better)

R^2 Score (Coefficient of Determination):

- Explains the proportion of variance explained by the model
- Closer to 1 indicates better performance

How We Evaluated:

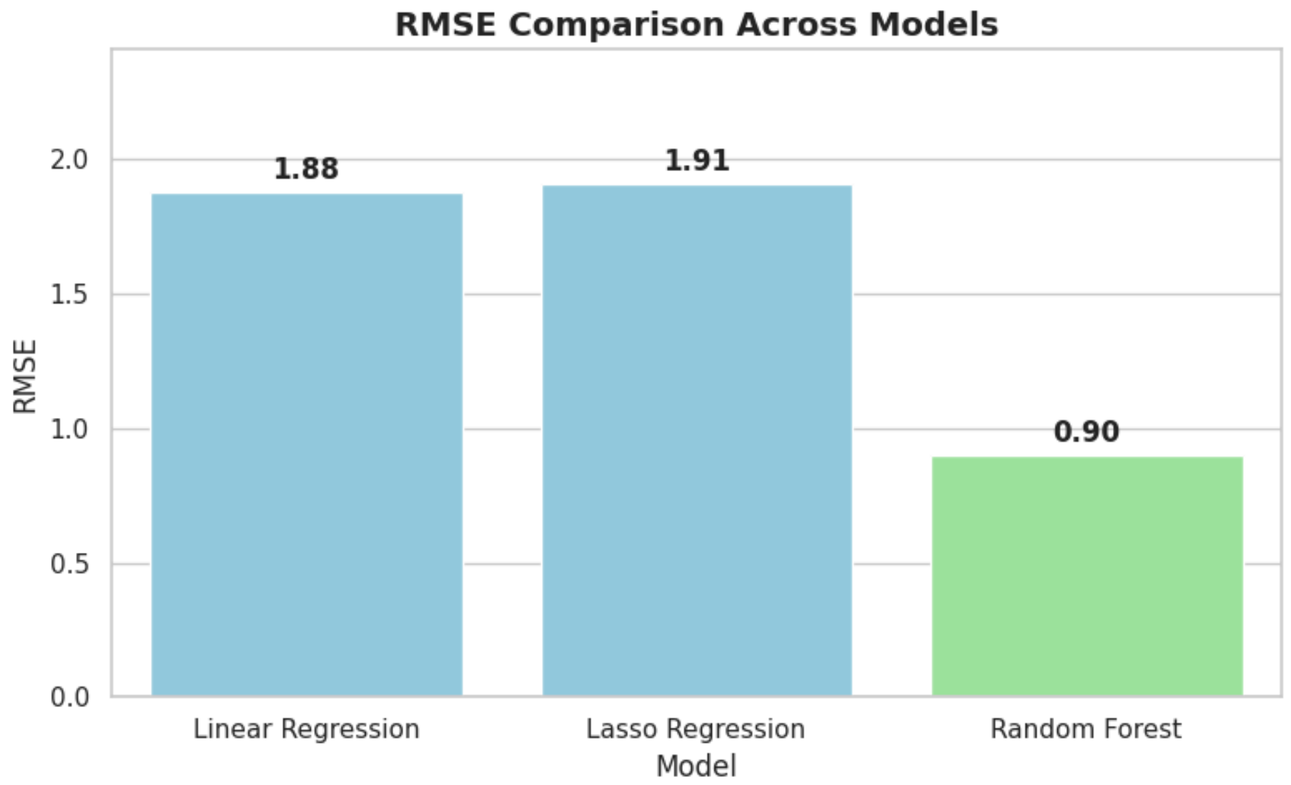
- Models were evaluated on the test set only (unseen data)
- Compared all models (Linear, Lasso, Random Forest) using the same metrics

MODEL COMPARISON

Interpretation

- **Random Forest** outperforms both Linear and Lasso Regression across all metrics.
- It has the lowest error (RMSE = 0.90), and highest accuracy (R^2 = 0.96), suggesting it captures complex nonlinear relationships in the data better.
- Linear and Lasso are similar, but Lasso underperformed possibly due to over-regularization or missing some important variables.

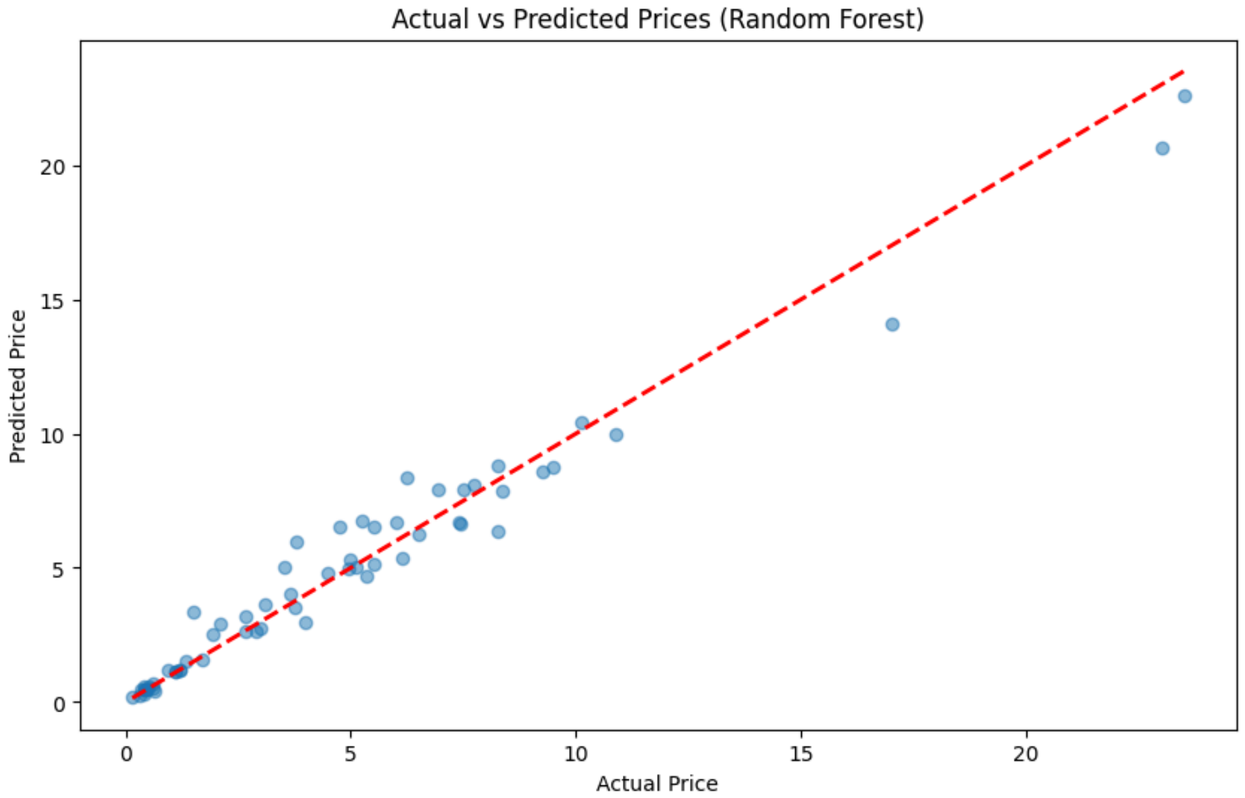
```
Model Performance Comparison:

Linear Regression:
  MSE: 3.53
  RMSE: 1.88
  R²: 0.85

Lasso Regression:
  MSE: 3.67
  RMSE: 1.91
  R²: 0.84

Random Forest:
  MSE: 0.82
  RMSE: 0.90
  R²: 0.96
```



Interpretation

So,among the models evaluated, the **Random Forest regression model** showed the best performance, achieving the lowest RMSE and highest R^2. This suggests it's the **most suitable for our car price prediction task."**

PREDICTION RESULTS



Interpretation:

1.Good Alignment with Red Line:

- Most points lie close to the red dashed line, which suggests that the model is performing quite well — it's able to predict prices close to the actual ones.

2.Some Outliers:

- A few points deviate significantly from the red line, especially in the upper range of prices.This may indicate the model slightly struggles with very high-priced cars possibly due to fewer training examples in that range (a common issue in regression tasks).

3.No Major Systematic Bias:

- There's no clear trend of consistent over- or under-prediction across the entire range. That suggests the model is fairly balanced overall.

PREDICTED SELLING PRICE -EXAMPLE

```python
# Example prediction
sample_car = {
    'Car_Name': 'swift',
    'Year': 2018,
    'Present_Price': 7.5,
    'Kms_Driven': 25000,
    'Fuel_Type': 'Petrol',
    'Seller_Type': 'Dealer',
    'Transmission': 'Manual',
    'Owner': 0
}

predictions = predict_car_price_all_models(sample_car)
print("\nPredictions for sample car:")
for model, pred in predictions.items():
    print(f"{model}: {pred:.2f} lakhs")
```

```
Predictions for sample car:
Linear: 6.21 lakhs
Lasso: 6.23 lakhs
Random Forest: 5.81 lakhs
```

Model Prediction:

- Predicted Selling Price: ₹ 5.81 Lakhs

(Actual Selling Price: ₹ 5.4Lakhs)

Interpretation:

- The model predicted the price very close to the actual price
- Indicates that the model captures key pricing patterns accurately
- Can be useful for both sellers and buyers in setting realistic expectations

INTERPRETATION

1. Performance Summary

- Random Forest Regressor gave the best results with high accuracy
- Linear and Lasso Regression were also effective but slightly less accurate

2. Real-World Meaning

- The model successfully predicts used car prices based on historical data
- Can be used to estimate fair selling prices for buyers and sellers

3. Key Observations

- Car Age and Present Price heavily influence the selling price
- The model generalizes well and can assist in pricing decisions in the market

CONCLUSION

1. Project Goal Achieved

- Successfully built models to predict used car prices
- Compared multiple algorithms to identify the best-performing model

2. Best Model

- *Random Forest Regressor* delivered the most accurate predictions

with an R^2 score of 0.96 and the lowest RMSE

3. Insights Gained

- Car Age, Present Price, and Kms Driven are key factors in determining price
- Proper data cleaning, EDA, and feature engineering were crucial for model success

4. Practical Impact

- The model can be used in real-world scenarios to help set fair car prices
- Beneficial for buyers, sellers, and automobile platforms

FUTURE WORK

1. Improve Model Performance

- Explore advanced algorithms like XGBoost or Gradient Boosting
- Perform hyperparameter tuning for better accuracy

2. Expand Dataset

- Include more recent data for current market trends
- Add features like location, brand reputation, or service history

3. Feature Engineering

- Create new features such as: Car Age Category (e.g., new, mid-age, old),Price per km driven

4. Build a User-Friendly Tool

- Develop a web application to predict car prices using the trained model
- Allow users to input car details and get instant estimates

REFERENCES

1.Journal of Machine Learning Research, 12, 2825–2830.
🔗 https://jmlr.org/papers/v12/pedregosa11a.html

2.Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to Linear Regression Analysis (5th ed.). Wiley.

3.McKinney, W. (2018). Python for Data Analysis. O'Reilly Media.

4.Towards Data Science Article – Random Forest for Regression Explained:
🔗 https://towardsdatascience.com/random-forest-regression-explained-5f607cdb4f6e

5.Towards Data Science Article – A Guide to Lasso Regression:
🔗 https://towardsdatascience.com/lasso-regression-explained-fcd175f546f4


THANK YOU