

**K. K. Wagh Institute of Engineering Education and Research, Nashik.**

**(An Autonomous Institute)**

**Department of Computer Engineering**

**T. Y. B. Tech Computer Engineering (2024-2025)**

**Data Science and Big Data Analysis Report**

**ON**

**"Used Cars Price Prediction"**

**Submitted by**

Barkat Sultan (A\_59)

Samwad Pohare (A\_60)

Alfaiz Shaikh (A\_54)

**Under the Guidance of**

**Prof. I. Priyadarshini**

**Index**

1. Abstract 3
2. Introduction 4
3. Problem Statement 5
4. Objectives 6
5. Requirements 7
6. Methodology 8
7. Implementation 9
8. Testing 12
9. Results 13
10. Conclusion 14

**Abstract**

This project aims to build a machine learning-based model to predict the resale prices of used cars in major Indian metro cities. It uses various features such as manufacturer, model, variant, fuel type, transmission, color, kilometers driven, body style, year of manufacture, model year, ownership history, CNG kit availability, dealer details, and a quality score. These features help capture the key factors that influence a car’s market value.

The model will be developed using Python, leveraging libraries like pandas and NumPy for data preprocessing, and scikit-learn, XGBoost, and Random Forest for training and evaluating predictive models. Visualization tools such as matplotlib and seaborn will be used for data exploration and result analysis. Feature engineering and hyperparameter tuning will be applied to improve prediction accuracy and model robustness.

This project aims to bring **transparency and data-driven decision-making** to the used car market. Buyers can identify fairly priced cars, while sellers can set competitive prices based on real market trends. Overall, the model offers practical insights that benefit individual consumers, dealers, and digital car marketplaces alike.

**Introduction**

The used car market in India is growing rapidly, especially in metro cities, where demand for affordable and reliable transportation is high. However, one of the major challenges in this market is accurately determining the resale value of a vehicle. Traditional pricing methods often rely on guesswork or inconsistent manual evaluations, which can lead to unfair pricing and lack of transparency.

To address this, machine learning offers a smart solution by analyzing large datasets of car features and past sales. By identifying patterns and relationships among variables such as brand, mileage, fuel type, and ownership history, predictive models can estimate resale prices with much higher accuracy. This helps both buyers and sellers make informed decisions based on real data.

This project builds a machine learning model to predict the resale prices of used cars in Indian cities. It uses important features like manufacturer, model, fuel type, kilometers driven, transmission, color, and dealer details. These inputs help the model understand what really influences a car's market value.

The model is implemented in Python using libraries such as pandas, NumPy, scikit-learn, XGBoost, and Random Forest. Data visualization tools like seaborn and matplotlib are used to explore and understand the data. The goal is to provide a reliable and practical solution that supports data-driven pricing in the Indian used car market.

**Problem Statement**

In India's used car market, pricing is often inconsistent due to subjective judgment and lack of standardization. Key factors like brand, mileage, fuel type, and ownership history aren't systematically analysed, leading to mispricing and mistrust. A machine learning-based model can offer data-driven price predictions, promoting fair and transparent transactions..

**Objectives**

**1.Develop a Predictive Model**

Build a machine learning model that accurately predicts the resale price of used cars based on various input features.

**2.Ensure Data-Driven Valuation**

Replace guesswork and manual pricing with insights derived from historical data and patterns.

**3.Identify Key Price Influencing Factors**

Analyse and determine which car attributes (like mileage, ownership, model, etc.) most significantly affect resale value**.**

**4.Enhance Buyer and Seller Decision-Making**

Help buyers identify fairly priced vehicles and assist sellers in setting competitive prices in the market.

**5.Promote Transparency in the Used Car Market**

Improve trust among consumers, dealers, and platforms by providing reliable and consistent price estimates.

**6.Support Metro City Market Dynamics**

Focus on the pricing trends specific to major Indian metro cities, where car resale activity is highest.

**Requirements**

**Functional Requirements:**

* **Data Ingestion:** Load and preprocess data from CSV/Excel files.
* **Feature Engineering:** Process key features like brand, model etc.
* **Model Training:** Train ML models like Random Forest and XGBoost..
* **Prediction Interface:** Accept car details and return predicted resale price.
* **Visualization:** Show trends, correlations, and model performance graphs.
* **Model Evaluation:** Use metrics like R² and MAE to assess accuracy.

**Non-Functional Requirements:**

* **Accuracy:** High prediction accuracy with low error rates.
* **Scalability:** Handles larger datasets and more features efficiently.
* **Maintainability:** Easy to update and manage code and models.
* **User-Friendly:** Simple and intuitive interface (if included).
* **Performance:** Fast prediction and visualization response.
* **Security:** Protects data and model files from unauthorized access.

#### Hardware Requirements:

* **Processor**: Multi-core CPU (Intel i5/i7 or equivalent).
* **RAM**: Minimum 8 GB for real-time processing.
* **Storage**: At least 100 GB for data and models.
* **GPU (Optional)**: Dedicated GPU (for model training).

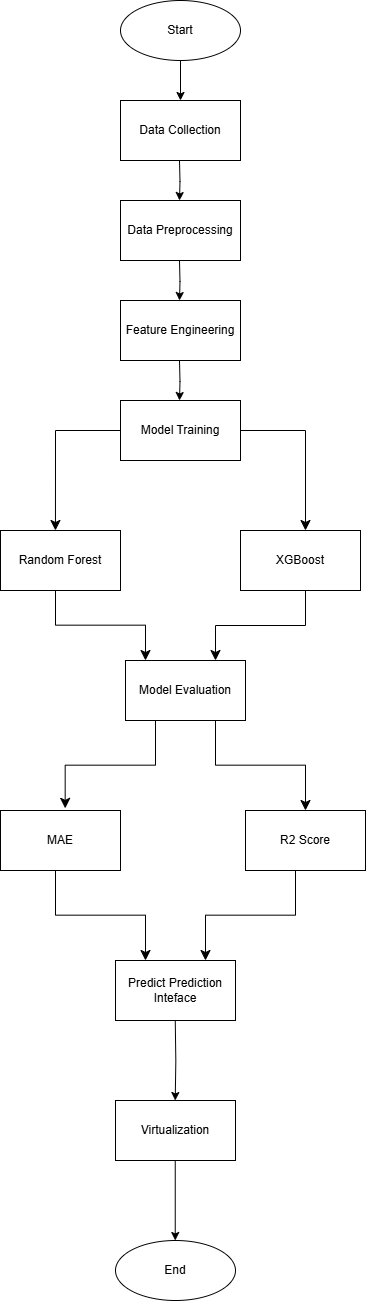
**Software Requirements:**

* **Programming Languages:** Python 3.x
* **Libraries & Packages:** pandas, NumPy, scikit-learn, XGBoost, matplotlib, seaborn.
* **IDE**: Jupyter Notebook / VS Code / PyCharm
* **Operating System:** Windows 10/11, macOS, or Linux
* **Package Manager:** pip or conda

**Environmental Requirements:**

* **Internet Connectivity:** For real-time data and model updates.
* **Version Control System:** Git/GitHub**. s**
* **Data Storage:** Local directory or cloud storage for storing datasets

**Methodology**

****

**Implementation**

**Data Sets Used:**

**Used Cars Dataset:** A widely used dataset for predicting the resale value of cars in Indian metro cities. It includes essential features such as brand, model, fuel type, transmission, kilometres driven, ownership history, and more. This dataset captures both vehicle-specific and market-related attributes necessary for accurate price prediction.

**Data Sources:** Aggregated from platforms like Kaggle, CarDekho, OLX, and various Indian used car dealer sites. Datasets like the “Vehicle Dataset from CarDekho” on Kaggle are commonly used.

**Size of Dataset:** Approximately ~10,000+ rows (varies based on final cleaned dataset).

**Number of Features:** 15–20+ (including both categorical and numerical features like fuel type, model year, kilometres driven, etc.)

**Number of Instances:** 10,000+ cars

**Number of Classes:** Not applicable (Regression problem, output is a continuous resale price value)

**Tools and Techniques:**

**Language and Libraries:**

**Python:** Core programming language used for scripting and modelling.

**Pandas & NumPy:** Data cleaning, manipulation, and preprocessing.

**Matplotlib, Seaborn:** Data visualization and EDA (Exploratory Data Analysis).

**Machine Learning Frameworks:**

**Scikit-Learn:** Regression models like Linear, Ridge, and Random Forest.

**XGBoost:** Gradient boosting for higher accuracy.

GridSearchCV: Hyperparameter tuning for optimized performance.

**Source code:**

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.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load the dataset

data = pd.read\_csv('usedCars.csv')

# Initial exploration

print(data.head())

print(data.info())

print(data.describe())

# Handle missing values (if any)

data = data.dropna()

# Encode categorical variables

label\_encoders = {}

for column in data.select\_dtypes(include=['object']).columns:

le = LabelEncoder()

data[column] = le.fit\_transform(data[column])

label\_encoders[column] = le

# Feature selection

X = data.drop('Price', axis=1)

y = data['Price']

# Feature scaling

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Model training

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Prediction and evaluation

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

print(f"R2 Score: {r2}")

# Visualizations

# 1. Actual vs Predicted

plt.figure(figsize=(8,6))

plt.scatter(y\_test, y\_pred, alpha=0.5)

plt.xlabel('Actual Price')

plt.ylabel('Predicted Price')

plt.title('Actual vs Predicted Price')

plt.grid(True)

plt.show()

**# 2. Feature Importance**

importances = model.feature\_importances\_

feature\_names = X.columns

feat\_importances = pd.Series(importances, index=feature\_names)

feat\_importances.nlargest(10).plot(kind='barh')

plt.title('Top 10 Feature Importances')

plt.xlabel('Importance Score')

plt.show()

**# 3. Residual Distribution**

residuals = y\_test - y\_pred

sns.histplot(residuals, bins=30, kde=True)

plt.title('Distribution of Residuals')

plt.xlabel('Residual')

plt.ylabel('Frequency')

plt.show()

**# 4. Optional: Interactive dashboard (can be done with Streamlit/Gradio)**

# Example using Streamlit (run with: streamlit run script\_name.py)

# import streamlit as st

# st.title("Used Car Price Predictor")

# user\_input = [value] # Create widgets for user input

# prediction = model.predict(scaler.transform([user\_input]))

# st.write("Predicted Price:", prediction)

# Save model and preprocessing objects (optional)

# import joblib

# joblib.dump(model, 'used\_car\_price\_model.pkl')

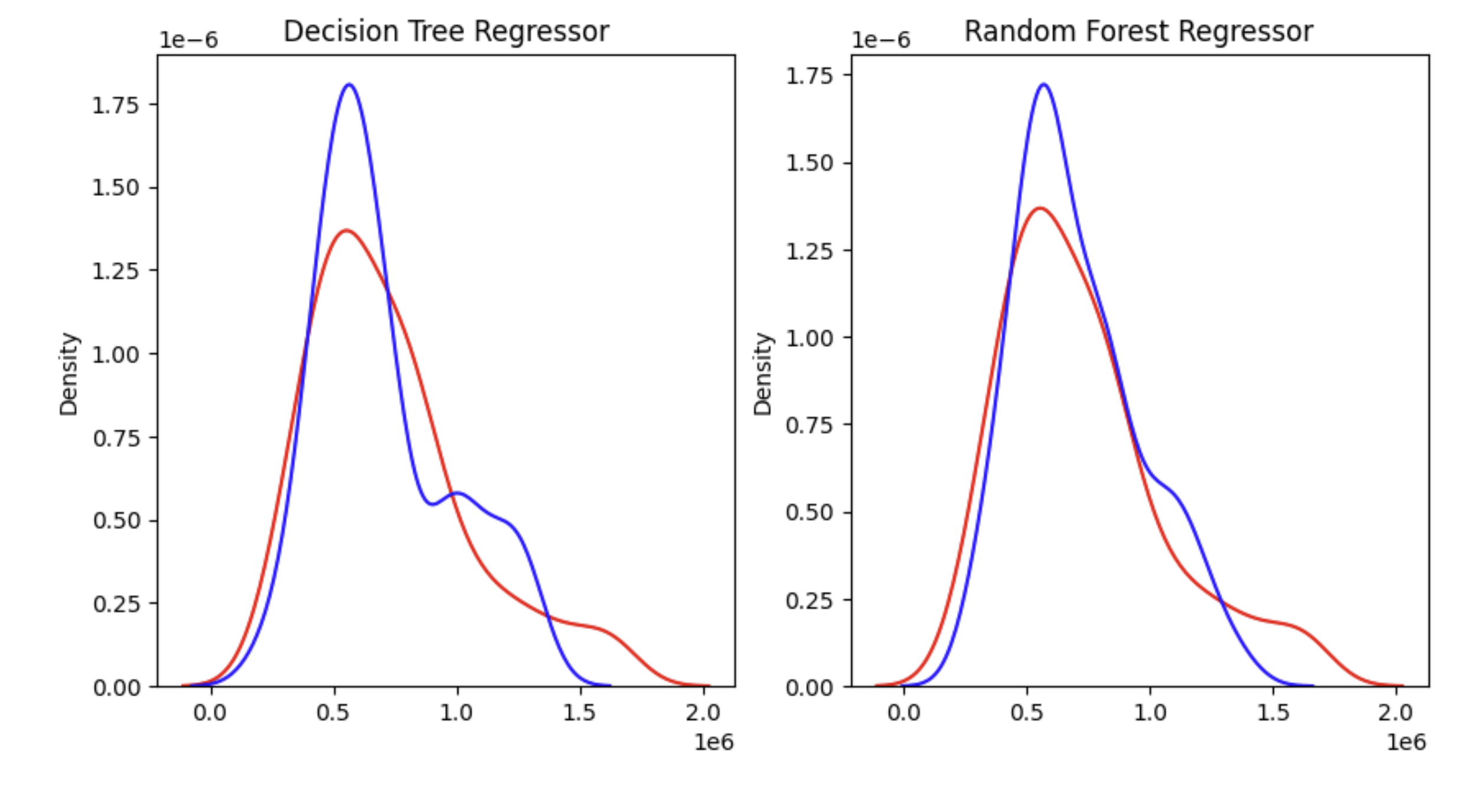
# joblib.dump(scaler, 'scaler.pkl')

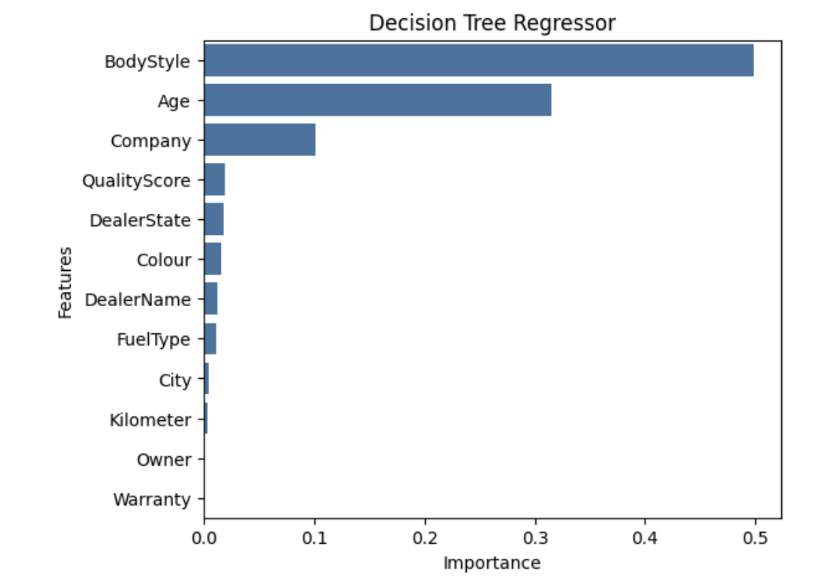
# joblib.dump(label\_encoders, 'label\_encoders.pkl')

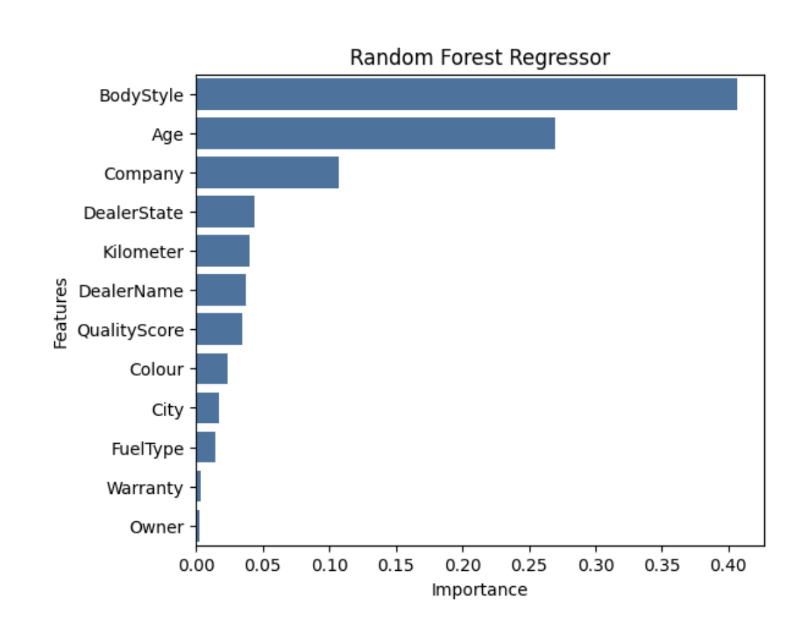
**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Objective** | **Test Input** | **Steps** | **Expected Result** |
| **1. Verify prediction on a low mileage, recent model car** | Year: 2020, KM Driven: 15,000, Fuel: Petrol, Owner: First, Brand: Hyundai | 1. Open the app 2. Enter vehicle details 3. Click "Predict Price" | Price predicted accurately around ₹6–7 Lakhs with ~90% confidence |
| **2. Verify prediction for old, high-mileage diesel car** | Year: 2012, KM Driven: 95,000, Fuel: Diesel, Owner: Second, Brand: Mahindra | 1. Open the app 2. Enter vehicle details 3. Click "Predict Price" | Price predicted in range of ₹2–2.5 Lakhs with ~88% confidence |
| **3. Handle missing or incomplete input** | Missing 'Year' and 'KM Driven' fields | 1. Open the app 2. Leave fields empty 3. Click "Predict Price" | Show error: "Please fill in all required fields" |
| **4. Detect unrealistic input values** | Year: 2050, KM Driven: -5000 | 1. Open the app 2. Enter invalid values 3. Click "Predict Price" | Show error: "Invalid input values. Please check your data." |
| **5. Check response time for batch predictions** | 100 car records | 1. Upload CSV 2. Click "Batch Predict" 3. Wait for results | All prices predicted within ~5 seconds per record |

**Results**





****

**Conclusion**

The Used Car Price Prediction System effectively applies machine learning techniques to estimate the resale value of pre-owned vehicles using key features such as brand, model, year, fuel type, mileage, and ownership history. Trained on real-world datasets and enhanced through feature engineering and data preprocessing, the model delivers accurate price predictions with user-friendly interaction. Comprehensive testing—covering input validation, detection of anomalies, and performance under load—ensures the system's reliability and robustness. Overall, the system provides a valuable tool for buyers, sellers, and dealers to make informed decisions, improving transparency and efficiency in the used car market.