# **Used car price Prediction**

**Used Car Price Prediction**

This repository contains a Used Car Price Prediction Model built using machine learning to estimate Car Prices. The model uses cars data to estimate car prices, helping customers to estimate accurate prices and prevent financial losses. By analyzing patterns in the data, this model can offering a valuable tool for customers and businesses.



**Project Overview**

**Project Overview**

This project develops a machine learning model to predict the prices of used cars based on various factors, enabling buyers, sellers, and dealerships to make informed decisions. Leveraging historical data, the model aims to provide accurate price predictions

**Objective**

Design and train a robust machine learning model to predict used car prices with high accuracy, reducing errors and improving decision-making for stakeholders.

**Dataset Overview**

Dataset contains these Features:  
**Car ID**: A unique identifier for each car listing.  
Brand: The brand or manufacturer of the car (e.g., Toyota, Honda, Ford, etc.).  
**Model**: The model of the car (e.g., Camry, Civic, Mustang, etc.).  
**Yea**r: The manufacturing year of the car.  
**Kilometers Driven**: The total kilometers driven by the car.  
**Fuel Type:** The type of fuel used by the car (e.g., Petrol, Diesel, Electric, etc.).  
**Transmission**: The transmission type of the car (e.g., Manual, Automatic).  
**Owner Type**: The number of previous owners of the car (e.g., First, Second, Third).  
**Mileage**: The fuel efficiency of the car in kilometers per liter.  
**Engine**: The engine capacity of the car in CC (Cubic Centimeters).  
**Power**: The maximum power output of the car in bhp (Brake Horsepower).  
**Seats**: The number of seats available in the car.  
**Price**: The selling price of the car in INR (Indian Rupees), which is the target variable to predict.

## **Modeling Approach**

The approach to building this used car price prediction model includes:

**Data Preprocessing**:

Handle missing values and outliers.

Normalize or standardize transaction data for consistency.

**features Selection**:

Time-Based Features:

1. Year (manufacturing year)

2. Kilometers Driven (total kilometers driven)

3. Owner Type (number of previous owners)

Behavioral Features:

1. Fuel Type (type of fuel used)

2. Transmission (transmission type)

3. Mileage (fuel efficiency)

4. Engine (engine capacity)

5. Power (maximum power output)

Vehicle Characteristics:

1. Brand

2. Model

3. Seats

**Model Selection:**

Linear Models:

1. Linear Regression: Baseline model for price prediction

2. Lasso Regression: Feature selection and regularization

3. Ridge Regression: Handling multicollinearity

Ensemble Models:

1. Random Forest: Capturing complex relationships

**Evaluation Metrics:**

1. Mean Absolute Error (MAE)

2. Mean Squared Error (MSE)

3. Root Mean Squared Error (RMSE)

4. R-Squared (R²)

**Usage**:

**Data Loading**: Load your transaction dataset in the notebook.

**Data Preprocessing**: Follow the preprocessing steps to clean and transform the data.

**Feature Engineering**: Apply feature engineering methods as defined in the notebook.

**Model Training and Evaluation**: Train and evaluate the model using the chosen techniques and metrics.

## **Technologies Used:**

* **Python**: Programming language for model development and data manipulation.
* **Jupyter Notebook**: Interactive environment for data analysis and modeling.
* **Scikit-Learn**: Machine learning library for model training and evaluation.
* **Pandas and NumPy**: For data processing and manipulation.
* **Matplotlib/Seaborn**: For visualizations.

**Deployment:**

**GUI Interface:**

**1. Design:** Create a simple, intuitive interface for users to input car features.

**2. Widgets:** Text boxes, dropdown menus, sliders, or checkboxes for feature input.

**3. Buttons:** "Predict Price" button to trigger prediction.

**Joblib Integration:**

1. **Model Loading:** Load the trained model using Joblib.

2. **Prediction:** Use the loaded model to predict prices based on user input.

3. **Output:** Display predicted price on the GUI interface.



**Benefits:**

1. Accurate price predictions for buyers and sellers

2. Improved decision-making for dealerships and investors

3. Enhanced market transparency and efficiency

4. Reduced risks associated with price uncertainty

## **Contributors:**

**. Aakash Prathipati**