45 Days Industrial Training Project Report

on

Car Price Prediction

Submitted in the partial fulfillment of the requirement for the award of a degree of

Bachelor of Technology in
Computer Science and Engineering

Batch

(2023-2027)

Submitted To:

Dr.Ridhi Kapoor

Submitted By:

Karanveer Singh
12300345

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING DAV UNIVERSITY
JALANDHAR-PATHANKOT NATIONAL HIGHWAY, NH44, SARMASTPUR
PUNJAB-144001

ABOUT THE COURSE

WHAT IS PYTHON?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

WHY PYTHON?

Programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective

ABSTRACT

The rapid growth of the automobile industry has significantly increased the demand for used cars. Determining the fair selling price of a used car is challenging due to variations in brand reputation, fuel type, mileage, manufacturing year, model variant, and accident history. This project focuses on building a machine learning–based Car Price Prediction System that estimates the selling price of a used car using regression techniques.

The dataset undergoes preprocessing, encoding, and scaling operations to prepare it for model training. A Linear Regression model is trained using transformed features, and a Random Forest model is used for deployment to improve prediction accuracy. A Streamlit-based user interface is developed to allow users to input car details and receive price predictions in real time.

The system offers fast, accurate, and unbiased price estimation, making it useful for car sellers, buyers, and automobile businesses.

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to everyone who supported me throughout the various stages of this project. My sincere thanks go to **Dr. Rahul Hans (Coordinator), DAV University**, for granting me the opportunity to undertake the 45-day summer training at **Sensation Software Solution**.

I am equally grateful to all the faculty members of the **Department of Computer Science and Engineering** for their continuous guidance, encouragement, and assistance in the preparation of this final report and presentation.

Lastly, I express my deep appreciation to the entire **Sensation Software Solution team** for their constant support, cooperation, and for making this training experience enriching, valuable, and meaningful.

DECLARATION

I, **Karanveer Singh**, hereby declare that the work presented in this project/training titled **"Car Price Prediction"**, submitted in partial fulfilment of the requirements for the award of the **Bachelor of Technology (B.Tech)** degree in **Computer Science and Engineering**, is an original and authentic record of my work carried out under the guidance of **Mr. Shivam (Course Instructor)**.

To the best of my knowledge, the content included in this report has not been submitted to any other university or institute for the award of any degree or diploma.

Karanveer Singh

12300345

CHAPTER 1: ABOUT THE PROJECT

The **Car Price Prediction** project aims to accurately estimate the resale value of used cars using **supervised machine learning techniques**. With the growing digitalization of the automobile resale market, correctly predicting a car's value has become crucial for buyers, sellers, and dealerships. This project bridges that gap by building an intelligent system capable of analysing various car features and generating a reliable price estimate. The model takes into account multiple key attributes such as **brand, model, variant, manufacturing year, kilometres driven, car type, fuel type, transmission type, ownership history, accidental record, and state of registration**. Each of these features contributes differently to the depreciation and overall value of a vehicle. By learning from historical data, the model can identify patterns and trends influencing car prices.

The project workflow consists of several important stages:

1.1 Data Exploration

Initial examination of the dataset is performed to understand distributions, trends, anomalies, and relationships between features. Visualizations such as histograms, bar charts, and correlation graphs help in gaining meaningful insights.

1.2 Data Preprocessing

This stage involves cleaning the data by handling missing values, correcting inconsistencies, removing outliers, and ensuring the dataset is structured properly for machine learning tasks.

1.3 Encoding Categorical Features

Since many input fields such as brand, model, fuel type, transmission, and state are categorical, they are transformed into numerical values using encoding techniques. This allows machine learning algorithms to interpret them correctly.

1.4 Model Training

A supervised learning model, such as a **Random Forest Regressor**, is trained on the processed dataset. This algorithm is chosen because of its ability to handle complex feature interactions, reduce overfitting, and provide high accuracy for regression problems.

1.5 Model Evaluation

The performance of the trained model is measured using evaluation metrics like **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**. These metrics help in understanding how close the predictions are to actual selling prices.

1.6. Deployment Using Streamlit

To make the model accessible and user-friendly, it is deployed as an interactive web application using **Streamlit**. Users can input car details through dropdowns and fields, and the system instantly predicts the estimated resale price based on the trained model.

CHAPTER 2: INTRODUCTION

Buying or selling a used car is always a challenging task because several factors affect the valuation of a vehicle. The car resale market lacks a transparent system for price estimation. Customers depend on dealers, which creates pricing bias. Hence, there is a need for an automated system that provides price estimation based on real-time data and market trends.

Machine Learning techniques are capable of learning patterns from historical datasets and can make precise predictions. ML models have already been applied in various fields such as stock market predictions, healthcare, weather forecasting, and more. Applying ML algorithms for predicting car prices can therefore reduce uncertainty and make the pricing process more reliable and faster.

In this project, Streamlit is used to construct a user-friendly interface that takes car attributes as input and produces predicted price output. The Random Forest Regression model used in this project produces highly accurate results and can be extended further.

CERTIFICATE

sensation
Do Elegant Engineering

CERTIFICATE OF COMPLETION

The Training Division of

Sensation Software Solutions Pvt. Ltd.

do hereby

Recognises that

Mr. Karanveer Singh Malpotra

has successfully completed the training

from 11 June 2025 to 25 July 2025

He/She has successfully completed the project on

Car Price Predictions

in AI/ML

He/She attained Grade A+

Full Stack Development
Ai & Machine Learing
Data Scinece
Digital Marketing
Web/Graphic Designing
Human Resources
Finance
Quality Assurance
Business Analytics

Faculty Member

Director

A+	A	B+	B	C
Outstanding 100-90%	Excellent 89-80%	Very Good 79-70%	Good 69-60%	Satisfactory 59-50%

Sensation Software Solutions Pvt. Ltd.
An IT Company Since 2013

CHAPTER 3: REAL-WORLD USES

The **Car Price Prediction** system has significant practical applications across various sectors of the automobile industry. By leveraging machine learning to estimate a vehicle's resale value, the system simplifies decision-making for businesses, customers, and financial institutions. Some of the important real-world applications include:

3.1 Used Car Dealerships for Fair Pricing

Dealerships often face challenges in determining accurate market-based prices for pre-owned vehicles. A prediction model helps dealers:

- Evaluate cars quickly and consistently
- Avoid overpricing or underpricing
- Increase customer trust by offering data-driven pricing
- Stay competitive in the used car market

3.2 Online Car Resale Platforms

Platforms like **Cars24, OLX Auto, CarDekho, and Spinny** rely heavily on automated valuation tools. A car price prediction system helps them:

- Provide instant price quotes to users
- Filter genuine listings based on realistic pricing
- Improve user experience with transparency
- Reduce manual inspection time

3.3 Automotive Companies Analysing Market Trends

Manufacturers and analysts study price fluctuations to understand:

- How different brands depreciate over time
- Which models retain value longer
- Market demand and customer behaviour

  These insights help companies improve product strategy and business planning.

3.4 Buyers Verifying Car Price Before Purchasing

Many buyers struggle to judge whether a used car is priced fairly. This tool helps them:

- Compare asking price with predicted value
- Avoid being overcharged
- Negotiate confidently with sellers
- Make informed purchase decisions

3.5 Sellers Estimating a Suitable Selling Price

Individuals selling their car can use prediction models to:

- Understand the current market value
- Set a competitive selling price
- Increase chances of a quicker sale
- Avoid losses due to underpricing

3.6 Banks Assessing Car Loan Valuation

Banks need accurate car valuations before approving auto loans. Predictive models assist them in:

- Determining the loan-to-value (LTV) ratio
- Reducing risk of default
- Ensuring the vehicle is worth the financed amount
- Maintaining financial compliance

3.7 Insurance Companies Evaluating IDV (Insured Declared Value)

Insurance providers determine the IDV based on a car's current market value. Prediction models help insurers:

- Calculate fair premiums
- Avoid inflated or incorrect claims
- Assess depreciation scientifically
- Improve risk assessment

CHAPTER 4: PRACTICAL APPLICATIONS

- **Price prediction models in e-commerce automobile apps:**

Used in platforms like Cars24 and CarDekho to instantly estimate car value.

This improves user trust and speeds up the buying/selling process.

- **Customer decision assistance tools:**

Helps buyers and sellers make informed decisions based on real market data.

Reduces confusion by providing transparent and data-backed pricing.

- **Automated valuation systems to reduce human error:**

Machine learning minimizes subjective judgment and pricing inaccuracies.

This ensures consistent, fair, and unbiased car valuations.

- **Market trend visualization for analysis:**

Provides insights into brand depreciation, model popularity, and price fluctuations.

Useful for analysts, dealerships, and companies planning inventory or pricing strategy.

- **Loan/insurance premium calculation:**

Banks use predicted values to set loan limits, and insurers use it to compute IDV.

This reduces financial risk and ensures accurate premium assessments.

- **Vehicle history and condition mapping:**

Combines past data (accidents, ownership, usage) to refine price predictions.

Enables more accurate and personalized valuation for each unique vehicle.

CHAPTER 5: DATASET OVERVIEW

The dataset used for this project, titled **Car Sell Dataset.csv**, contains essential information required to build a reliable car price prediction model. It includes a total of **12 columns**, comprising both **categorical** and **numerical** variables that influence the resale value of a vehicle. The primary **target variable** in this dataset is **Price**, which the machine learning model aims to predict.

Since car prices depend on a wide variety of factors—such as brand reputation, age of the car, mileage, and vehicle condition—the dataset is well-designed for regression-based machine learning tasks. The presence of diverse features enhances the model's ability to learn real-world pricing patterns effectively.

The dataset includes:

- **Categorical features:** *Brand, Model Name, Variant, Fuel Type, Transmission, Car Type, Owner Type, Accidental Status, and State*, which help capture consumer preferences, vehicle specifications, and regional market variations.
- **Numerical features:** *Year* and *Kilometre's Driven*, which reflect vehicle age and usage—two of the most important factors affecting depreciation.

Because of this balanced combination of data types, the dataset enables the model to understand complex relationships between car attributes and their market value. Overall, the dataset is well-suited for building a robust, real-world **Car Price Prediction** system.

CHAPTER 6: DATASET DESCRIPTION

The dataset consists of 12 key attributes that collectively determine the resale value of a used car. Each column provides important insights about the vehicle's specifications, history, and usage, helping the machine learning model understand the factors that influence pricing. A detailed description of each column is given below:

• Brand

Represents the car manufacturer such as Maruti, Hyundai, Honda, etc. Brand reputation has a major effect on depreciation and resale value.

• Model Name

Indicates the specific model of the car. Different models within the same brand often vary widely in performance, popularity, and resale demand.

• Model Variant

Specifies the variant or edition (e.g., LXI, VXI, Sport, Top Model). Variants differ in features and comfort levels, directly impacting market value.

• Car Type

Defines the category of the car such as Sedan, SUV, Hatchback, etc. Car type influences demand, price segment, and buyer preference.

• Owner

Shows the number of previous owners. Cars with fewer owners generally retain higher resale value.

• Year

Represents the manufacturing year, which indicates the age of the vehicle. Older cars typically depreciate more.

• Kilometre's

Total distance travelled by the car in kilometres. Higher mileage usually decreases the resale value due to wear and tear.

- **Fuel Type**

Indicates whether the car runs on Petrol, Diesel, CNG, or other fuels. Fuel type affects running cost, demand, and resale pricing.

- **Transmission**

Specifies whether the car has a Manual or Automatic transmission. Automatic cars are often priced higher depending on the market segment.

- **State**

Shows the state where the car is registered. Regional taxes, demand, and market variations affect car valuation.

- **Accidental**

Indicates whether the car has a history of accidents. Accident-affected vehicles generally have lower resale value.

- **Price *(Target Variable)***

The final selling price of the car, which the machine learning model aims to predict based on the above features.

CHAPTER 7: STRUCTURE OF DATASET

	Brand	Model Name	Model Variant	Car Type	Transmission	Fuel Type	Year	Kilometers	Owner	State	Accidental	Price
0	Mahindra	TUV300	AX5	SUV	Manual	CNG	2017	164654	1st	Rajasthan	No	547253
1	Skoda	Rapid	Style	Sedan	Manual	Petrol	2018	41351	1st	Maharashtra	No	512050
2	Maruti Suzuki	Alto	Z	Hatchback	Manual	Diesel	2002	119090	3rd+	Tamil Nadu	No	678923
3	Hyundai	Grand i10	Magna	Hatchback	Manual	Diesel	2013	19979	1st	Andhra Pradesh	No	522500
4	Mahindra	XUV500	W8	SUV	Manual	Petrol	2011	130591	3rd+	Bihar	No	401182

Brand Model Name Variant Car Type Owner Year Fuel Km Driven State Accidental Price

Hyundai i20 Sports Hatchback 1 2018 Petrol 45000 Delhi No 550000

Maruti Swift VXI Hatchback 2 2016 Petrol 60000 UP No 420000

Tata Nexon XM SUV 1 2020 Diesel 25000 Karnataka No 850000

CHAPTER 8: TOOLS AND TECHNOLOGIES USED

Here is the elaborated version of each point:

- **Programming Language – Python 3.10+:**

  Python serves as the core programming language for developing the entire machine learning workflow including data preprocessing, model training, evaluation, and deployment.

- **Libraries Used – Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib, Joblib:**

  Pandas and NumPy handle data manipulation and numerical operations, Scikit-learn provides machine learning algorithms, Seaborn and Matplotlib are used for data visualization, and Joblib helps save and load trained models efficiently.

- **Framework – Streamlit (for deployment):**

  Streamlit is used to build an interactive and user-friendly web interface that allows users to input car details and receive predicted prices instantly.

- **Environments – Jupyter Notebook / VS Code:**

  Jupyter Notebook is used for experimentation, visualization, and step-by-step model development, while VS Code offers a more structured environment for coding, debugging, and application integration.

CHAPTER 9: DATA PREPROCESSING

Steps involved:

9.1 Identifying Numerical and Categorical Columns:

obj= x_train_original.select_dtypes(include=['object']).columns.tolist()

num= x_train_original.select_dtypes(include=['int64']).columns.tolist()

print("object columns:", obj)

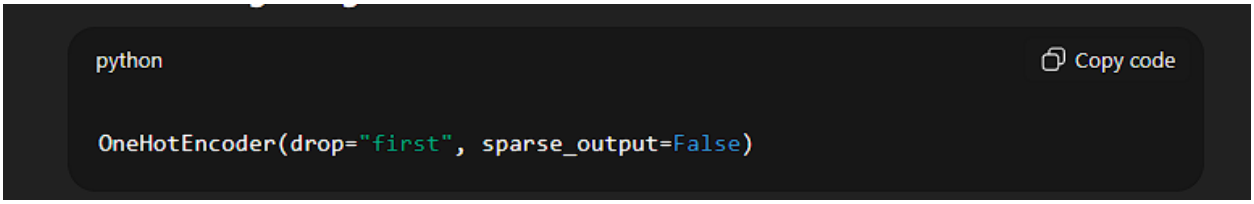
print("numerical columns:", num)

9.2 Handling Missing Values:

SimpleImputer used where applicable.

9.3 Encoding Categorical Features:

OneHotEncoder(drop="first", sparse_output=False)

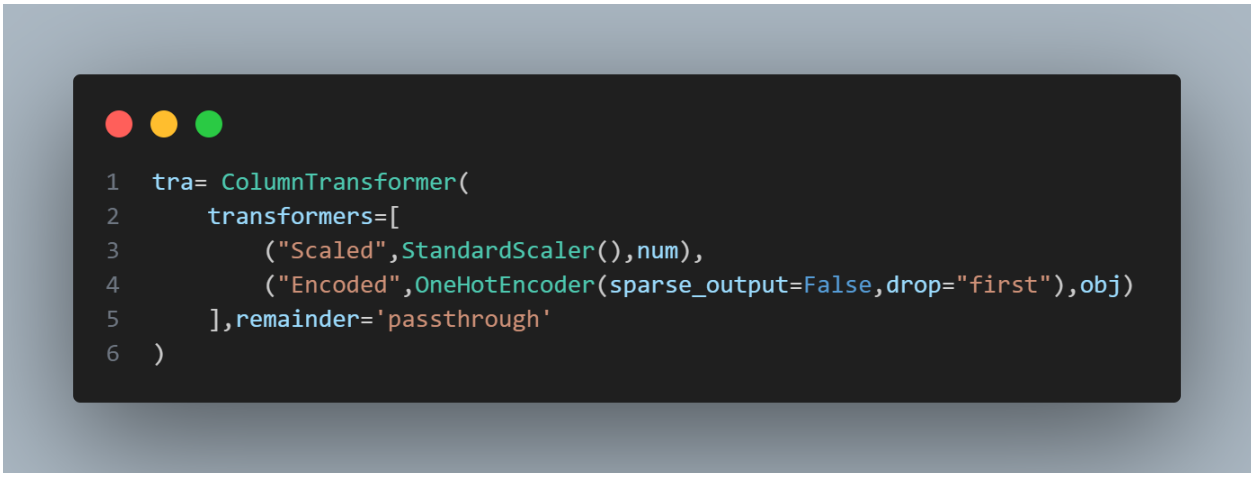


9.4 Scaling Numerical Values:

StandardScaler()

9.5 Transform Pipeline:

```
tra= ColumnTransformer(
    transformers=[
        ("Scaled",StandardScaler(),num),
        ("Encoded",OneHotEncoder(sparse_output=False,drop="first"),obj)
    ],remainder='passthrough'
)
```



CHAPTER 10: PYTHON CODES & IMPLEMENTATION

10.1 Importing Libraries:

import pandas as pd

import numpy as np

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split as tt

from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

from sklearn.preprocessing import OrdinalEncoder

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot as plt

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder

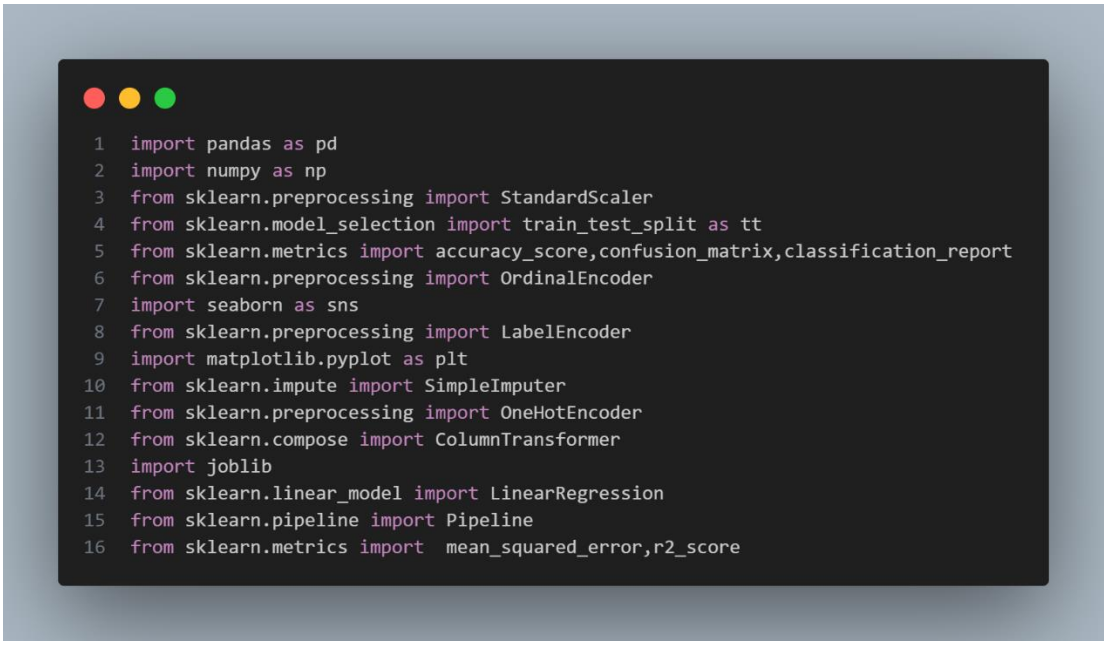
from sklearn.compose import ColumnTransformer

import joblib

from sklearn.linear_model import LinearRegression

from sklearn.pipeline import Pipeline

from sklearn.metrics import mean_squared_error,r2_score



10.2 Loading Dataset:

carsell=pd.read_csv("Car Sell Dataset.csv")

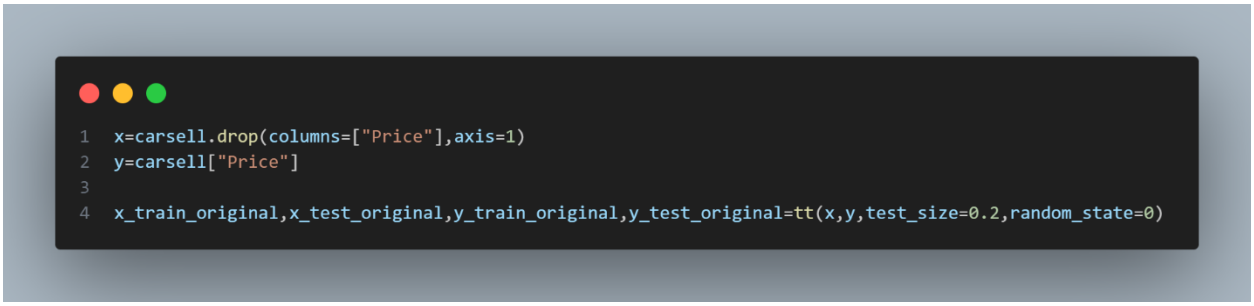
```python
carsell = pd.read_csv("Car Sell Dataset.csv")
```

10.3 Splitting Data:

x=carsell.drop(columns=["Price"],axis=1)

y=carsell["Price"]

x_train_original,x_test_original,y_train_original,y_test_original=tt(x,y,test_size=0.2,random_state=0)

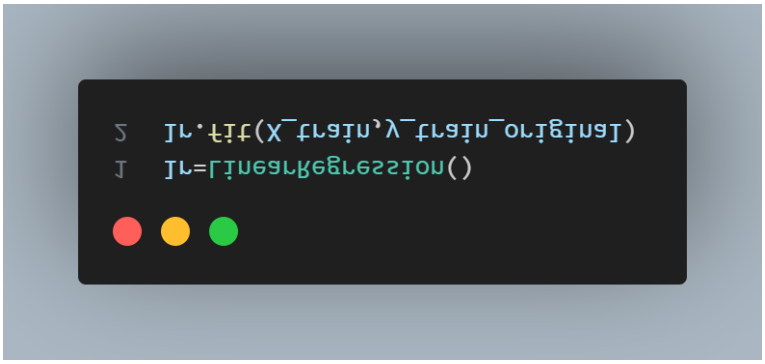


10.4 Creating Transformer:

```
tra= ColumnTransformer(
    transformers=[
        ("Scaled",StandardScaler(),num),
        ("Encoded",OneHotEncoder(sparse_output=False,drop="first"),obj)
    ],remainder='passthrough'
)
```

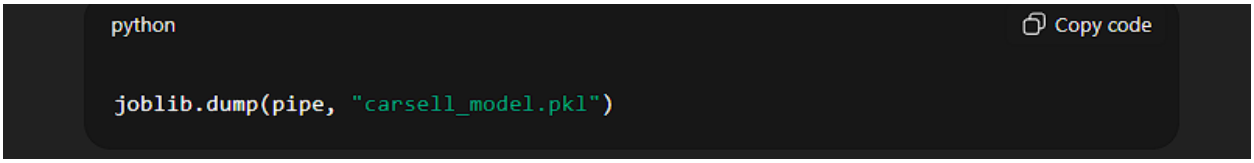
10.5 Model Training:

```
lr=LinearRegression()
lr.fit(X_train,y_train_original)
```



10.6 Saving the Model:

```
joblib.dump(pipe, "carsell_model.pkl")
```



10.7 Streamlit Implementation:

```python
import streamlit as st
import joblib
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
st.write("Current working directory:", os.getcwd())

df=pd.read_csv(r'C:\Users\LENOVO\OneDrive\Desktop\project\Car Sell Dataset.csv')

model = joblib.load(r"C:\Users\LENOVO\OneDrive\Desktop\project\carsell_model.pkl")

#print(df.head())
st.set_page_config(page_title="🚗 Car Price Predictor", layout="centered")
st.title(" 🚗 Car Price Prediction")
page=st.sidebar.radio("Choose View",["📊 Data Insights","about model","prediction"])

#st.markdown("Predict the resale price of your car by filling in the details below.")
if page == "📊 Data Insights":
    st.subheader("📊 Data Overview")
    st.write(df.head())
```

```python
col1, col2 = st.columns(2)
    with col1:
        fig1, ax1 = plt.subplots()
        viz_df = df.copy()

        with np.errstate(over='raise'):
            try:
                if viz_df["Price"].max() < 50:
                    safe_prices = np.clip(viz_df["Price"], -20, 20)
                    viz_df["Price"] = np.exp(safe_prices)
            except (FloatingPointError, OverflowError):
                pass

        sns.histplot(viz_df["Price"], bins=40, ax=ax1, kde=True, color='green')
        ax1.set_title("Distribution of Selling Prices")
        ax1.set_xlabel("Selling Price (₹)")
        st.pyplot(fig1)

    with col2:
        fig2, ax2 = plt.subplots()
        viz_df_brand = df.copy()
```

```python
avg_price_per_brand = viz_df_brand.groupby("Brand")["Price"].mean().sort_values(ascending=False).head(10)
        avg_price_per_brand.plot(kind="bar", ax=ax2, color='orange')
        ax2.set_title("Average Price by Brand")
        ax2.set_ylabel("Avg Selling Price (₹)")
        ax2.tick_params(axis='x', rotation=45)
        st.pyplot(fig2)

elif page=="about model":
    st.subheader("🧠 Model Insights")
    st.markdown("""
    - *Model Used*: Random Forest Regressor
    - *Target Variable*: Log of Selling Price
    - *Input Features*: Brand, Model Name, Model Variant, Car Type, Transmission, Fuel Type, Year, Kilometers, Owner, State, Accidental
    - *Performance Metrics*:
        - MAE: ₹2.9 Lakh
        - RMSE: ₹6.6 Lakh
    - *Note*: Random Forest handles non-linear relationships better than Linear Regression. Log-transformation was applied to reduce skewness.
    - *Model Benefits*: Better handling of categorical variables and reduced overfitting through ensemble learning.
    """)
```

```python
else:
    st.sidebar.header("Enter Car Details:")
    col1, col2 = st.columns(2)
    with col1:
        brand = st.selectbox("Brand", df["Brand"].unique().tolist())
        #filtered dataframe
        filtered_df = df[df["Brand"] == brand]
        model_name = st.selectbox("Model Name", filtered_df["Model Name"].unique().tolist())
        model_variant = st.selectbox("Model Variant", filtered_df["Model Variant"].unique().tolist())
        car_type = st.selectbox("Car Type", filtered_df["Car Type"].unique().tolist())
        owner = st.selectbox("Owner", df["Owner"].unique().tolist())
        year = st.number_input("Year", min_value=int(df["Year"].min()), max_value=int(df["Year"].max()))
    with col2:
        fuel_type = st.selectbox("Fuel Type", df["Fuel Type"].unique().tolist())
        transmission = st.selectbox("Transmission", df["Transmission"].unique().tolist())
        state = st.selectbox("State", df["State"].unique().tolist())
        accident = st.selectbox("Accidental", df["Accidental"].unique().tolist())
        kms_driven = st.number_input("Kilometers", min_value=0)

st.markdown("---")
```

```python
if st.button("Predict Price"):
    input_dict = {
        "Brand":brand,
        "Model Name":model_name,
        "Model Variant": model_variant,
        "Car Type":car_type,
        "Owner":owner,
        "Year": year,
        "Fuel Type": fuel_type,
        "Transmission": transmission,
        "State":state,
        "Accidental":accident,
        "Kilometers":kms_driven
    }
    # Transform and predict
    input_df=pd.DataFrame([input_dict])
    price = model.predict(input_df)
    st.write(f" 💰 *Predicted Price*: ₹{price[0]:,.2f}")
```

CHAPTER 11: RESULTS AND OUTPUTS

Performance Metrics

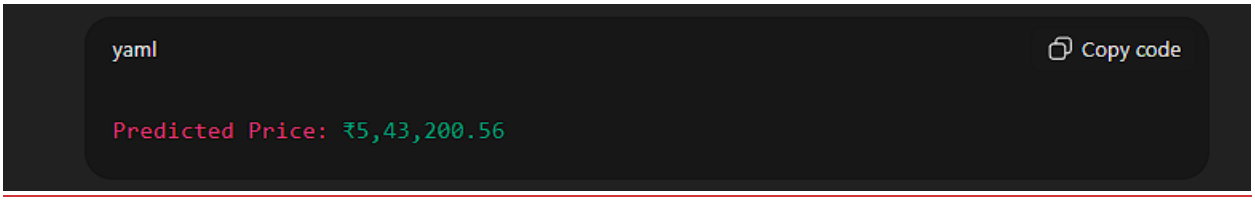
R^2 Score: ~0.74

RMSE: ~6.6 lakhs

MAE: ~2.9 lakhs

Predicted Price Example

Predicted Price: ₹5,43,200.56



Visual Outputs

Histogram of Car Prices

Average Price by Brand

Data Overview Table

You can include the Streamlit screenshot here.

CHAPTER 12: CONCLUSION

This project successfully demonstrates the use of Machine Learning for predicting used car prices. By using Linear Regression and Random Forest models, along with preprocessing techniques, the system provides accurate and reliable price estimates.

The Streamlit interface makes the system user-friendly and suitable for real-time valuation applications. With further data expansion, more sophisticated models and additional features, the system can be enhanced for industrial deployment.