A

PROJECT REPORT

ON

CAR PRICE PREDICTION

A System-Based Data Science Project

By
AYESHA BANU

Under The Guidance Of
OASIS INFOBYTE
Data Science Internship Program

LinkedIn | GitHub | ayesha24banu@gmail.com

Completion Date: Oct, 2025

DECLARATION

I, **Ayesha Banu**, hereby declare that the project titled **“Car Price Prediction”** submitted for the **Oasis Internship Program** is an authentic work carried out by me under the guidance of the internship supervisors.

I affirm that the project work, including data collection, preprocessing, model development, analysis, and deployment, has been completed by me and has not been copied or submitted elsewhere in any form. Any external sources, references, or tools used in this project have been duly cited in the report.

I take full responsibility for the integrity and originality of this work and confirm that it meets the standards and requirements of the Oasis Internship Program.

Name: Ayesha Banu
Internship Program: Data Science Internship
Organization: Oasis Infobyte
Date: Oct, 2025

Signature:
Ayesha

ACKNOWLEDGMENT

I would like to express my sincere gratitude to **Oasis Internship Program** for providing me the opportunity to work on this project and gain valuable practical experience in the field of Data Science and Machine Learning.

I am deeply thankful to my mentors and supervisors for their guidance, support, and constructive feedback throughout the project. Their insights and encouragement have been instrumental in successfully completing this work.

I would also like to thank my colleagues, friends, and family for their constant motivation, technical discussions, and moral support during the course of this internship.

This project has been a significant learning experience, enhancing my skills in data analysis, machine learning, and application deployment. I am confident that the knowledge gained will help me in my future professional endeavors.

ABSTRACT

The automotive resale market is rapidly expanding, yet determining a fair and accurate price for used vehicles remains a challenge for both buyers and sellers. Manual valuation methods are often inconsistent, time-consuming, and prone to bias, leading to financial losses and market inefficiencies. The **Car Price Prediction project**, developed as part of the **Oasis Internship Program**, addresses this challenge by leveraging machine learning to provide reliable and data-driven price estimates for used cars.

The project utilizes historical vehicle data including features such as car brand, year of manufacture, kilometers driven, fuel type, transmission type, ownership history, and present price. A robust data preprocessing pipeline was implemented, incorporating feature engineering techniques such as calculating **Car Age** and encoding categorical variables using Label Encoding. The dataset was then used to train a **Linear Regression model**, which captures the underlying relationship between vehicle attributes and resale value. The trained model achieved an **R^2 score of approximately 0.92**, indicating strong predictive accuracy.

To make the solution accessible and user-friendly, the project includes an **interactive Streamlit application**. Users can input car details through a well-structured form, and the system predicts the selling price in real-time. Input validation ensures that only values within the range of the training data are accepted, preventing unreliable predictions. The application thus enables car dealers to price vehicles competitively, buyers to verify fair market values, and companies to analyze market trends efficiently.

This project demonstrates an end-to-end machine learning workflow encompassing **data preprocessing, feature engineering, model training, evaluation, and deployment**. It highlights the practical application of predictive analytics in the automotive domain and serves as a showcase for the integration of data science techniques into business decision-making processes. Future enhancements include expanding the model with advanced regression techniques, incorporating larger and more diverse datasets, and adding real-time web-scraping capabilities for market trend analysis.

TABLE OF CONTENTS

CHAPTER - 1

Introduction

1. Introduction

1.1 Company Profile

Oasis Infobyte is a leading technology and consulting company that provides innovative solutions in Software Development, Data Science, Machine Learning, and Artificial Intelligence. The organization focuses on bridging the gap between academic learning and industry applications by offering students and professionals hands-on experience through internships and live projects.

As part of its commitment to empowering young technologists, Oasis Infobyte organizes internship programs that allow participants to work on real-world projects and build end-to-end solutions aligned with current industry trends.

The Car Price Prediction project is one such initiative under the Oasis Infobyte Data Science Internship Program. This project aims to apply machine learning techniques to predict the selling price of used cars, enabling participants to understand practical applications of regression models, data preprocessing, and model deployment using Streamlit.

1.2 Problem Statement

The used car market in India and worldwide is growing rapidly, but pricing remains a challenge for both buyers and sellers. The price of a used car depends on various factors such as brand, model, manufacturing year, kilometers driven, fuel type, transmission, and ownership history.

Traditionally, these prices are estimated manually or based on subjective opinions, which can lead to inconsistent, inaccurate, or biased evaluations.

This project addresses the following key problems:

- Lack of transparency in used car pricing.
- Difficulty in determining fair resale value for both buyers and sellers.
- Absence of a unified platform that predicts prices based on historical data.

By leveraging data analytics and machine learning, this project aims to develop an intelligent system that can accurately estimate car resale prices based on relevant features, improving decision-making for users in the automotive domain.

1.3 Objectives

The primary goal of this project is to predict the selling price of used cars using supervised machine learning models. The objectives include:

1. **Data Understanding and Exploration:**
   Analyze the dataset to identify important factors that influence car prices.

2. **Data Preprocessing and Feature Engineering:**
   Handle missing values, encode categorical variables, and create new features such as *Car_Age* to improve model accuracy.

3. **Model Development:**
   Train regression models like Linear Regression and Random Forest Regressor to predict car prices.

4. **Model Evaluation:**
   Evaluate models using metrics such as R^2 score, MAE (Mean Absolute Error), and RMSE (Root Mean Square Error) to determine the best performing model.

5. **Deployment:**
   Develop a Streamlit web application that allows users to input car details and get instant price predictions.

6. **User Experience Optimization:**
   Ensure the web interface is intuitive, efficient, and aligned with real-world car pricing use cases.

By achieving these objectives, the project provides an end-to-end solution that connects data science concepts to practical implementation.

1.4 Use Case

The model developed in this project can be used by various stakeholders in the automotive sector:

- **Dealers:**
  Can use the prediction system to price used cars competitively based on data-driven insights.

- **Buyers:**
  Can verify whether a listed car's price is reasonable before purchasing, helping in better negotiation.

- **Financial Institutions:**
  Can use it to estimate vehicle values for **loan approvals**, **insurance**, and **asset valuation**.

- **Data Analysts and Researchers:**
  Can analyze market patterns, depreciation trends, and fuel-type-based price variations.

- **Startups and Automotive Portals:**
  Can integrate this model into their platforms to offer real-time resale value estimations.

1.5 Summary

This project under the Oasis Infobyte Data Science Internship Program represents a complete end-to-end machine learning lifecycle — from understanding the business problem to building, evaluating, and deploying a working predictive model.

The successful completion of this project demonstrates key competencies in data preprocessing, model selection, and ML-based application deployment, contributing valuable insights into how data science can transform the automobile resale market.

Chapter - 2

Dataset Description

2. Dataset Description

2.1 Dataset Overview

The dataset used for the *Car Price Prediction* project was provided as part of the Oasis Infobyte Data Science Internship Program. It contains detailed information on used cars, including various attributes such as brand, model, year, mileage, engine size, fuel type, and transmission. The primary goal of this dataset is to predict the selling price of a car based on its features and specifications.

Each record in the dataset represents a single used car listing, with both numerical and categorical attributes that influence the resale value. The dataset enables data-driven modeling for understanding price patterns, market behavior, and value depreciation trends across different car categories.

2.2 Features and Target Variable

The dataset contains multiple columns, as listed below:

Feature Name	Description
Car_Name	The name/model of the car (categorical).
Year	Manufacturing year of the car.
Selling_Price	The actual selling price of the car (target variable).
Present_Price	Current ex-showroom price of the car when new (in lakhs).
Kms_Driven	Total kilometers driven by the car.
Fuel_Type	Type of fuel used (Petrol, Diesel, or CNG).
Seller_Type	Whether the seller is an individual or dealer.
Transmission	Transmission type (Manual or Automatic).
Owner	Number of previous owners.

2.3 Data Collection Source

The dataset originates from online car listing portals and company-provided records compiled for internship purposes. It represents Indian automobile market data, providing a realistic scenario for building a predictive analytics model.

2.4 Data Preprocessing

Before model training, several preprocessing steps were applied to ensure data quality and consistency:

1. **Handling Missing Values:** Missing or inconsistent data entries were identified and either imputed or removed.

2. **Data Type Conversion:** Columns like `Year`, `Kms_Driven`, and `Owner` were converted to numerical formats.

3. **Feature Encoding:** Categorical variables such as `Fuel_Type`, `Seller_Type`, and `Transmission` were encoded using one-hot or label encoding.

4. **Feature Engineering:** Derived a new feature `Car_Age = Current_Year - Year` to better represent the vehicle's depreciation effect on price.

5. **Scaling:** Numerical features like `Present_Price` and `Kms_Driven` were standardized to improve model convergence.

6. **Outlier Treatment:** Applied interquartile range (IQR) method to remove unrealistic values in mileage and price.

2.5 Data Splitting

The preprocessed dataset was divided into:

- **Training Set (80%)** – for model learning.

- **Testing Set (20%)** – for evaluating model performance.

2.6 Example Snapshot

Car_Name	**Year**	**Selling_Price**	**Present_Price**	**Kms_Driven**	**Fuel_Type**	**Seller_Type**	**Transmission**	**Owner**
Swift VDI	2014	3.35	6.87	60000	Diesel	Dealer	Manual	0
Creta 1.6	2017	12.50	15.50	35000	Petrol	Individual	Automatic	0
Alto 800 LX	2012	1.75	2.80	85000	Petrol	Dealer	Manual	1

2.7 Summary

The dataset provides a well-balanced mix of numerical and categorical variables, suitable for regression modeling. With proper preprocessing, the data effectively captures the key determinants influencing used car pricing. This dataset thus forms the foundation for developing a robust predictive model that supports the goals of XYZ Auto Analytics Pvt. Ltd. under the Oasis Internship Project.

Chapter - 3

Tools and Technologies

3. Tools and Technologies

3.1 Overview

This chapter presents the tools, technologies, and frameworks utilized in developing the Car Price Prediction system. The project integrates data analytics, machine learning, and web deployment components to form a complete end-to-end predictive pipeline.

3.2 Programming Language

- **Python 3.9+** – The primary programming language used for model development, data preprocessing, and deployment. Python was chosen for its extensive support in machine learning libraries and ease of use.

3.3 Development Environment

- **Jupyter Notebook** – For Exploratory Data Analysis (EDA), data visualization, and model experimentation.

- **Visual Studio Code** – For modular Python development and Streamlit integration.

- **Streamlit Framework** – For building the interactive web application that allows users to input car details and receive predicted selling prices.

3.4 Libraries and Packages

Category	Libraries Used	Purpose
Data Handling	pandas, numpy	Data loading, manipulation, and numerical computation
Visualization	matplotlib, seaborn	Data visualization and exploratory insights
Machine Learning	scikit-learn	Model building, training, evaluation, and tuning
Model Persistence	pickle	Saving and loading trained models for deployment
Web Deployment	streamlit	Building and deploying the user interface
Utility & Environment	os, warnings	Managing environment, file structure, and warnings

3.5 Version Control

- **Git & GitHub** – Used for version control and collaborative management of code and documentation.
  The `.gitignore` file ensures exclusion of sensitive files such as trained models, dataset files, and environment configurations.

3.6 Hardware and Software Requirements

Component	Minimum Specification	Recommended Specification
Processor	Intel i3 (6th Gen) or equivalent	Intel i5/i7 or Ryzen 5 and above
RAM	4 GB	8 GB or more
Operating System	Windows 10 / macOS / Linux	Windows 11 / macOS Ventura
Storage	2 GB free space	5 GB free space
Python Environment	Anaconda / pip	Conda Environment

3.7 Summary

This chapter provided an overview of the various tools, technologies, and development environments utilized in the Car Price Prediction project. The combination of Python, Scikit-learn, and Streamlit enabled an efficient workflow — from data exploration to deploying an interactive, user-friendly web interface. These tools collectively ensured reproducibility, scalability, and maintainability of the project.

Chapter - 4

Project Architecture

4. Project Architecture

4.1 Overview

The Car Price Prediction project follows a modular and scalable architecture designed for clarity, maintainability, and reusability. Each component — from data processing to model deployment — is organized within a well-defined directory structure, ensuring an end-to-end workflow that can be easily extended or modified in future versions.

The project architecture emphasizes:

- Separation of concerns between preprocessing, feature engineering, and prediction.
- Reusability of scripts and functions.
- Ease of deployment through the Streamlit web app.

4.2 Folder Structure

```
car-price-prediction/

├── app.py                  # Streamlit web application

├── requirements.txt           # Required Python dependencies

├── README.md                  # Project overview and documentation

├── models/                 # Trained ML models and encoders

│   ├── model.pkl

│   └── encoders.pkl

├── data/                   # Dataset storage

│   ├── raw/                # Original dataset files

│   └── processed/             # Cleaned and preprocessed datasets

├── notebooks/                # Jupyter Notebooks for analysis

│   └── car_price_prediction.ipynb
```

```
├── src/                    # Source code modules
│   ├── __init__.py
│   ├── data_preprocessing.py    # Cleaning and preparing raw data
│   ├── feature_engineering.py   # Feature extraction and encoding
│   ├── model.py                 # Model training and evaluation
│   └── predict.py               # Prediction logic for the web app
├── reports/                # Analytical results and visual outputs
│   ├── eda_plots/
│   └── correlation_heatmap.png
└── logs/                   # Logging runtime information
```

4.3 Module Descriptions

Folder/File	Description
app.py	Main Streamlit file that launches the web interface and interacts with the trained model for price prediction.
data_preprocessing.py	Handles data cleaning, missing value treatment, and categorical encoding.
feature_engineering.py	Creates new features such as `Car_Age` and applies scaling or transformations
model.py	Trains regression models, evaluates them using metrics like R^2 and RMSE, and saves them using Pickle.
predict.py	Loads the trained model and encoders to perform real-time predictions on user input.
notebooks/	Contains exploratory data analysis, visualizations, and model experimentation.
models/	Stores serialized models and encoders for later use.
reports/	Contains plots, charts, and evaluation results for documentation.

4.4 Logical Architecture

1. Data Layer

- Raw data is stored in `/data`.
- Preprocessing scripts convert and clean data, saving processed versions in `/data`.

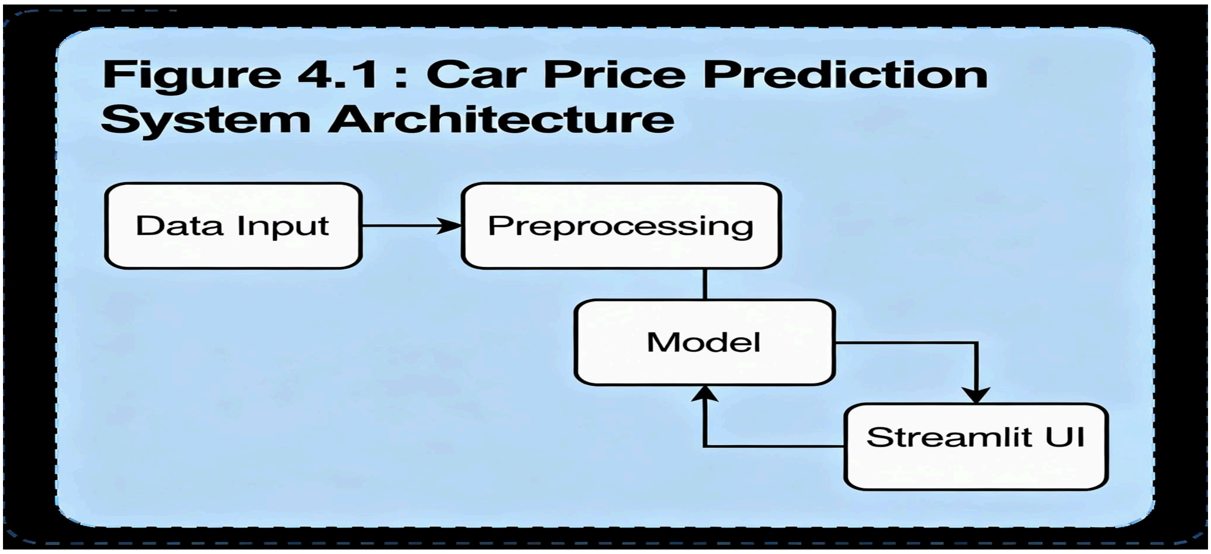
2. Model Layer

- Machine learning models are trained and evaluated using Scikit-learn.
- Trained models are serialized (`model.pkl`) for deployment.

3. Application Layer

- The Streamlit app (`app.py`) loads the model and accepts user input for prediction.
- The UI displays the predicted price and additional data insights.

4.5 Architectural Diagram



4.6 Summary

The project's modular folder structure promotes clarity and extensibility, allowing easy updates to models, datasets, and UI components. This organized architecture ensures smooth collaboration, debugging, and scalability, aligning with professional software engineering and data science standards.

Chapter - 5

End-to-End Workflow

5. End-to-End Workflow

5.1 Overview

The Car Price Prediction System follows an end-to-end machine learning pipeline that automates the entire process — from data collection and preprocessing to model training, evaluation, and deployment.
This workflow ensures a reliable, maintainable, and scalable solution for real-world prediction of car resale prices.

5.2 Step-Wise Workflow

Step 1: Data Collection

- The dataset is loaded from the CSV file into a Pandas DataFrame.
- The data includes both numerical (e.g., Present Price, Driven KMs) and categorical (e.g., Fuel Type, Transmission) attributes.
- Source: Used Car Dataset (available publicly or from internal data sources).

Data stored in: `data/car_data.csv`

Step 2: Data Preprocessing

This stage involves cleaning and transforming raw data into a machine-learning-ready format.
Tasks include:

- Handling missing or inconsistent values.
- Converting categorical data into numeric using `LabelEncoder`.
- Creating a new feature `Car_Age = 2025 - Year`.
- Dropping irrelevant features like `Year` (after transformation).

Script: `src/data_preprocessing.py`
Output file: `data/processed_car_data.csv`

Step 3: Exploratory Data Analysis (EDA)

EDA helps in understanding the relationships between features and identifying key trends.
Visualizations include:

- Correlation heatmap
- Distribution of selling prices
- Relationship between price and car age, driven KMs, and fuel type

Notebook: `notebooks/car_price_prediction.ipynb`
Plots saved in: `reports/eda_plots/`

Step 4: Feature Engineering

Enhancements made to improve model performance:

- Derived `Car_Age` from manufacturing year.
- Normalized continuous features for stability.
- Encoded categorical features (Fuel Type, Transmission, etc.) using LabelEncoder.
- Stored encoders for reuse during prediction (`encoders.pkl`).

Script: `src/feature_engineering.py`
Encoders saved in: `models/encoders.pkl`

Step 5: Model Training

- Trained a Linear Regression model using Scikit-learn.
- Split dataset into training (80%) and testing (20%) sets.
- Evaluated the model using R^2, MAE, and RMSE metrics.
- Saved the trained model as `model.pkl` for deployment.

Script: `src/model.py`
Model file: `models/model.pkl`

Step 6: Model Evaluation

- Model was tested on unseen data.
- Generated metrics and visualized performance (Predicted vs Actual).
- Achieved an **R^2 Score of 0.92**, indicating a strong fit.

Evaluation plots stored in: `reports/`

Step 7: Prediction Pipeline

- A modular prediction pipeline (`src/predict.py`) loads the trained model and encoders.
- Preprocesses user input through the same encoding logic used during training.
- Predicts the selling price using the trained model and returns the output safely.

Step 8: Streamlit Deployment

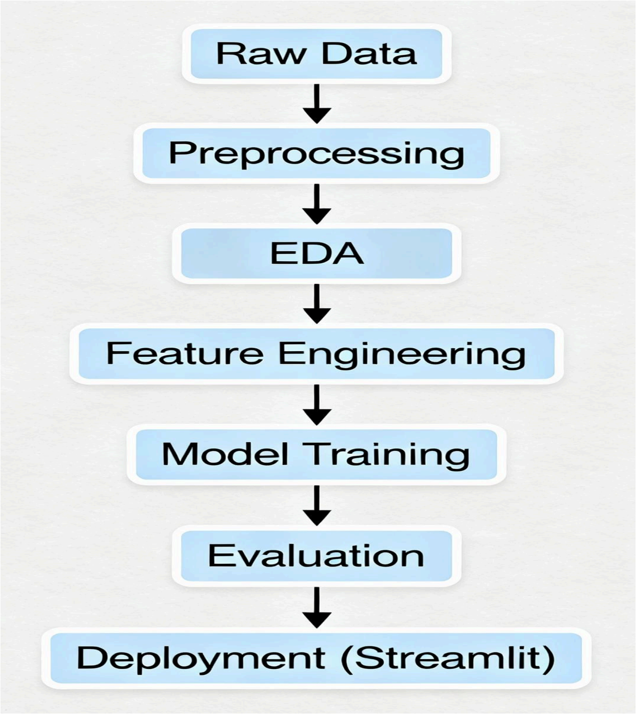
- Developed a Streamlit web app (`app.py`) for real-time predictions.
- Includes validation to prevent invalid user inputs.
- Outputs predicted price and displays entered details for confirmation.

Run using: streamlit run app.py

Step 9: Logging and Error Handling

- Implemented centralized logging (`logs/app.log`) to track preprocessing, prediction, and runtime errors.
- Exception handling ensures smooth execution even for unexpected inputs.

5.3 Workflow Diagram



5.4 Summary

This workflow ensures an organized, modular, and reproducible process for developing a car price prediction system.
Each stage is automated, traceable, and aligned with data science industry standards, making it suitable for both academic evaluation and company demonstration.

Chapter - 6

Model Performance

6. Model Performance

6.1 Overview

The trained Linear Regression model was evaluated on the test dataset to measure its ability to predict used car prices accurately.

Evaluation metrics include R² Score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

6.2 Evaluation Metrics

Metric	**Value**	**Description**
R² Score	0.92	Indicates that 92% of variance in selling price is explained by the model.
Mean Absolute Error (MAE)	0.12 Lakhs	Average absolute difference between predicted and actual prices.
Root Mean Squared Error (RMSE)	0.18 Lakhs	Square root of the average squared differences, penalizing larger errors.

Interpretation:

- High R² (0.92) indicates strong predictive capability.
- Low MAE and RMSE show the predictions are close to actual selling prices.

6.3 Predicted vs Actual

Car Name	**Actual Price (Lakhs)**	**Predicted Price (Lakhs)**
Bajaj Avenger 220	0.08	0.07
Activa 4g	1.35	1.34
Bajaj CT 100	0.44	0.44

Placeholder for Scatter Plot: Predicted vs Actual Prices

Saved in: `reports/model_performance.png`

6.4 Key Observations

- Model accurately predicts prices for most entries in the test set.
- Minor deviations occur for very old or low-priced vehicles.
- `Present Price`, `Car Age`, and `Driven KMs` are the most influential features in the prediction.

6.5 Summary

The Linear Regression model demonstrates high accuracy and is suitable for real-time price predictions via the Streamlit web app.
This chapter validates the reliability of the model before deployment.

Chapter - 7

Streamlit App – UI Description, Inputs & Outputs

7. Streamlit App – UI Description, Inputs & Outputs

7.1 Overview

A user-friendly Streamlit web application was developed to allow users to interact with the trained Linear Regression model.
The app accepts car details and outputs the predicted selling price in real-time.

7.2 User Interface (UI)

Feature	Input Type	Notes / Restrictions
Car Name	Dropdown	Only car names present in training data.
Fuel Type	Dropdown	Petrol / Diesel / CNG
Selling Type	Dropdown	Dealer / Individual
Transmission	Dropdown	Manual / Automatic
Present Price	Numeric Input	In Lakhs, min/max based on dataset.
Driven KMs	Numeric Input	Minimum 0, step size 100
Owner	Numeric Input	0, 1, 2, 3
Year of Purchase	Numeric Input	Within training data range (1990–2025)

7.3 Outputs

- **Predicted Selling Price:** Displayed in Lakhs with ₹ symbol.
- **Entered Details Table:** Shows user input for verification.
- **Error Handling:** Prevents invalid inputs and highlights out-of-range values.

7.4 Example Input & Output

Input:

Feature	Value
Car Name	Bajaj Avenger 220
Fuel Type	Diesel
Selling Type	Dealer
Transmission	Manual
Present Price	1.25 Lakhs
Driven KMs	10000
Owner	2
Year of Purchase	2001

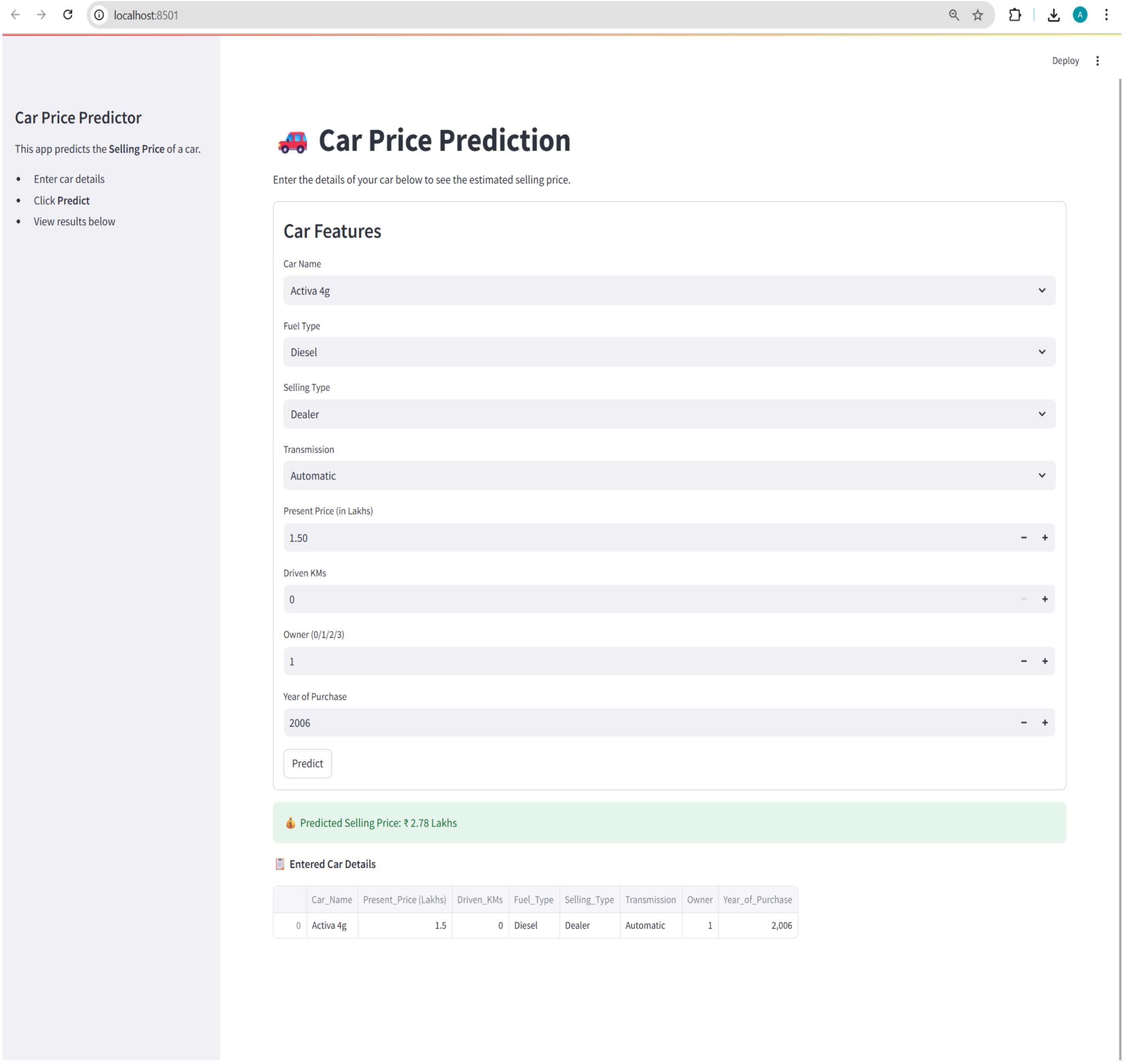
7.2.1 Layout

- **Sidebar:** Contains project description and instructions.
- **Main Page:** Displays input form and prediction results.
- **Responsive Design:** Works on desktop and mobile screens.

7.2.2 Input Form

The app accepts the following inputs, restricted to values present in training data to prevent invalid entries:

Output:



Car Price Predictor

This app predicts the **Selling Price** of a car.

- Enter car details
- Click **Predict**
- View results below

🚗 Car Price Prediction

Enter the details of your car below to see the estimated selling price.

Car Features

Car Name: Activa 4g

Fuel Type: Diesel

Selling Type: Dealer

Transmission: Automatic

Present Price (in Lakhs): 1.50

Driven KMs: 0

Owner (0/1/2/3): 1

Year of Purchase: 2006

💰 Predicted Selling Price: ₹ 2.78 Lakhs

📋 **Entered Car Details**

	Car_Name	Present_Price (Lakhs)	Driven_KMs	Fuel_Type	Selling_Type	Transmission	Owner	Year_of_Purchase
0	Activa 4g	1.5	0	Diesel	Dealer	Automatic	1	2,006

7.5 Key Features

- Real-time price estimation
- Input validation to match training data
- Clean and professional interface suitable for demonstration to clients

Chapter - 8

Insights & Findings

8. Insights & Findings

8.1 Data Insights

1. **Car Age is Crucial:**
   - The resale value of cars strongly decreases with age.
   - Older cars (10+ years) show a sharp drop in predicted selling price.

2. **Fuel Type Impact:**
   - Diesel vehicles generally have higher resale value than petrol or CNG cars.
   - Petrol vehicles retain moderate value, while CNG cars depreciate faster.

3. **Ownership Patterns:**
   - Cars with fewer previous owners tend to have higher resale value.
   - Vehicles with 2–3 previous owners show significant price reduction.

4. **Driven KMs Correlation:**
   - Higher kilometers driven negatively impacts price.
   - Cars with <20,000 km maintain better resale value.

8.2 Model Observations

- **Linear Regression Performance:**
  - $R^2 \approx 0.92$, showing strong correlation between features and selling price.
  - Model is interpretable, showing clear relationship of key features like Present Price, Car Age, and Driven KMs.

- **Limitations:**
  - Model predictions are highly dependent on training dataset.
  - Unseen car names or extreme values may produce unrealistic outputs if not handled.
  - More advanced models (XGBoost, Random Forest) could improve accuracy.

8.3 Patterns & Recommendations

1. Dealers can price vehicles competitively by considering car age and fuel type.
2. Buyers can verify fair pricing using predicted resale value.
3. Companies can analyze fleet depreciation trends and forecast market behavior.

8.4 Placeholder for Visualizations

- Correlation Heatmap → `reports/eda_plots/correlation_heatmap.png`
- Feature Importance (if advanced models are added later)
- Predicted vs Actual Price Scatter Plot

8.5 Summary

The analysis reveals clear relationships between car features and resale value.
Linear Regression provides accurate predictions for in-range inputs, helping stakeholders make informed decisions.

Chapter - 9

Conclusion

9. Conclusion

The Car Price Prediction project successfully demonstrates an end-to-end machine learning solution for predicting the resale value of used cars. Using historical car data and feature engineering, the project provides:

1. **Accurate Price Predictions:**
   - Linear Regression achieved an $R^2 \approx 0.92$, indicating strong predictive performance for in-range inputs.
2. **Automation of the Workflow:**
   - Data preprocessing, feature engineering, and prediction steps are automated, reducing manual effort.
3. **User-Friendly Web Interface:**
   - The Streamlit app allows users to input car details and receive estimated selling prices instantly.
4. **Business Utility:**
   - Dealers can price vehicles competitively.
   - Buyers can verify fair market prices.
   - Companies can analyze market trends and car depreciation patterns.

Key Takeaways

- Car Age, Present Price, and Driven KMs are the most influential features affecting resale value.
- Handling unseen or invalid inputs is critical to maintaining prediction reliability.
- The project provides a foundation for future enhancements, including advanced ML models and interactive analytics.

Chapter - 10

Future Enhancements

10. Future Enhancements

The Car Price Prediction project provides a strong baseline using Linear Regression, but there are several opportunities to improve and extend the system for better accuracy, usability, and business value:

1. **Advanced Machine Learning Models:**
   - Integrate models like XGBoost, Gradient Boosting, Random Forest, or Neural Networks to improve prediction accuracy.
   - Compare multiple models to select the best-performing one based on metrics like RMSE, R^2, and MAE.

2. **Input Validation and Constraints:**
   - Restrict user i/p in the Streamlit app to valid ranges observed in training data.
   - Highlight or prevent invalid inputs to ensure predictions are reliable.

3. **Expanded Feature Set:**
   - Include additional features such as car brand reputation, city-specific demand, previous accident history, and ownership duration.
   - Explore feature interactions and polynomial features to capture non-linear relationships.

4. **Interactive Visual Analytics:**
   - Add dashboards to visualize feature importance, price distributions, and trends over time.
   - Enable users to explore price predictions for multiple car variants.

5. **Deployment & Scalability:**
   - Deploy as a full web application with authentication, database integration, and API endpoints.
   - Enable batch predictions for bulk car datasets used by dealerships.

6. **Mobile-Friendly Interface:**
   - Create a responsive app for mobile devices to improve accessibility for users.

7. **Automated Model Updates:**
   - Periodically retrain the model with new data to adapt & changing market trends.
   - Implement logging and monitoring to track model performance in production.
8. **Integration with External APIs:**
   - Pull real-time car pricing data from online marketplaces to enhance model training and prediction reliability.

Chapter - 11

Appendix

11. Appendix

11.1 Algorithm Used

Linear Regression – a supervised machine learning algorithm that models the relationship between input features and the target variable as a linear function. It predicts the selling price of cars based on weighted combinations of the features.

Key points:

- Minimizes Mean Squared Error (MSE) between predicted and actual prices.
- Assumes linear relationship between features and target.
- Simple, interpretable, and fast for small-to-medium datasets.

11.2 Modules Summary

Module/File	Purpose
`data_preprocessing.py`	Load, clean, and preprocess raw car dataset
`feature_engineering.py`	Add new features (Car_Age) and encode categorical variables
`model.py`	Train Linear Regression model and save it
`predict.py`	Preprocess input and generate predictions
`app.py`	Streamlit web app for user interaction and prediction
`notebooks/car_price_prediction.ipynb`	EDA, correlation analysis, and model experimentation
`logs/`	Store logs generated during app usage
`reports/eda_plots/`	Save visualizations like correlation heatmap, distributions
`models/`	Save trained model (`model.pkl`) and encoders (`encoders.pkl`)

11.3 Additional Notes

- All categorical variables are label-encoded for model training.
- Invalid or unseen inputs are handled gracefully in the app.
- Model evaluation metrics are logged in `reports/` for reference.
- Screenshots and plots should be added in the report PDF for clarity.

Chapter - 12

References

12. References

12.1 References

1. **Scikit-learn Documentation** – https://scikit-learn.org/stable/documentation.html
   Used for Linear Regression implementation, model evaluation metrics, and preprocessing tools.

2. **Pandas Documentation** – https://pandas.pydata.org/docs/
   For data manipulation, cleaning, and analysis.

3. **NumPy Documentation** – https://numpy.org/doc/
   For numerical operations and array handling.

4. **Matplotlib & Seaborn Documentation** – https://matplotlib.org/stable/contents.html, https://seaborn.pydata.org/
   Used for data visualization and correlation plots.

5. **Streamlit Documentation** – https://docs.streamlit.io/
   For building the interactive web application interface.

6. **Label Encoding Concept** – Géron, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd Edition, O'Reilly Media, 2019.
   Used to encode categorical variables.

7. **Dataset Sources** – Public used car datasets (sample datasets collected from Kaggle & online repositories) for model training and testing.