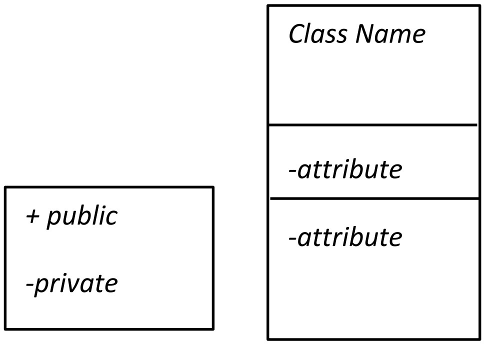
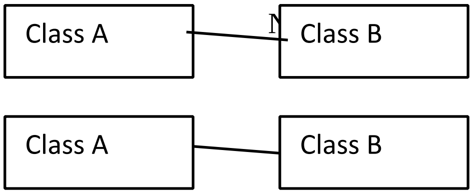
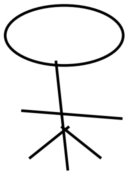
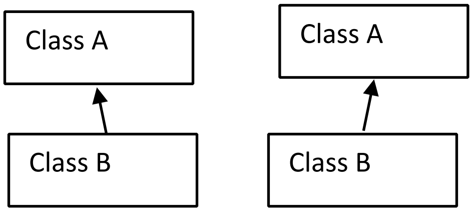
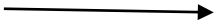
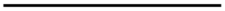
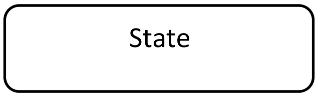
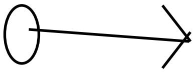
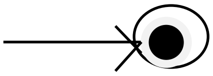
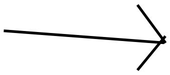
CAR QUINTESSENCE - A MACHINE LEARNING POWERED WEB APPLICATION FOR CAR PRICE PREDICTION

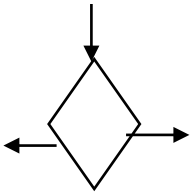
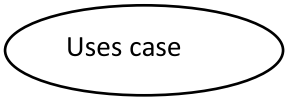
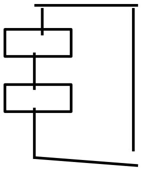
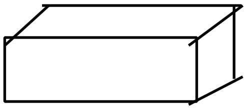
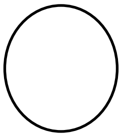
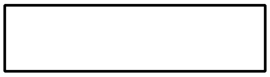
TABLE OF CONTENTS

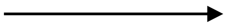
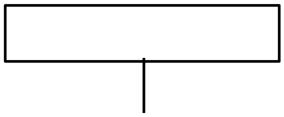
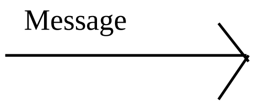
LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	 Class Name -attribute -attribute + public -private	Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)	 extends	Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processes.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.

12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.
14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard,sensors,etc.

18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

LIST OF ABBREVIATIONS

S.NO	**ABBREVATION**	**EXPANSION**
1**.**	ML	Machine Learning
2.	SVM	Support Vector Machine
3.	COMPUTER VISION & IMAGE PROCESSING TECHNIQUES	Convolutional Neural Networks
4.	ANN	Artificial Neural Networks
5.	AI	Artificial Intelligence
6.	DNN	Deep Neural Networks
7.	MTDNN	multitask deep neural networks
8.	MNIST	Modified National Institute of Standards and Technology
9	CNN	convolutional neural networks
10.	GAN	Generative Adversarial Networks
11.	NLP	natural language processing
12.	VGG	Visual Geometry Group
13.	ADAM	Adaptive Moment Estimation
14.	ReLU	Rectified Linear Unit

ABSTRACT

In the dynamic landscape of India's automobile market, where transparency in car pricing is imperative, this project introduces a transformative approach using the Random Forest algorithm. This project harnesses a robust dataset comprising over 6000 elements, significantly surpassing previous endeavors. The larger dataset contributes to heightened predictive accuracy, marking a substantial advancement in comparison to existing projects. A pivotal aspect of our initiative lies in harnessing the power of machine learning models and the integration with web applications and the advancement in predictive modeling is at the core of our initiative. Web integration ensures an effortless and interactive user experience, allowing users to input diverse factors influencing car prices easily. This user-friendly interface enhances accessibility and provides clear and transparent price predictions, facilitating a rich user experience. This enables us to achieve unprecedented accuracy levels, providing customers with a robust tool for making well-informed decisions, we hope to curb instances of overcharging and deceptive practices. By mitigating scams and malpractices, our project not only contributes to the evolution of predictive modeling in India's automotive industry but also promotes transparency and fairness, fostering positive change for consumers.

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In the domain of automotive valuation, a persistent challenge has been the lack of a precise and transparent mechanism for predicting car prices. Traditional methods fell short, leaving a gap that our project aims to fill. Previous models struggled to achieve the desired precision, often resulting in suboptimal decision-making for consumers. Recognizing this gap, our project sets out to redefine precision in automotive valuation, offering a solution that goes beyond conventional limitations. To improve accuracy, our model incorporates advanced predictive analytics and machine learning algorithms, analyzing a comprehensive set of features that influence car prices. By leveraging the power of the Random Forest Regressor, our model captures nuanced factors that were previously overlooked, resulting in more precise and reliable predictions.Moreover, the historical inefficiencies in older methods prompted our foray into the development of a sophisticated predictive model. Beyond predicting car prices, our project also addresses the global imperative for sustainability by integrating considerations for eco-friendly vehicles. This approach promotes informed choices, contributing to a reduced carbon footprint and a more sustainable automotive future. Additionally, a prevailing issue in the automotive sector has been the lack of transparency in pricing, leading to instances of overcharging and unfair practices. Our project emerges as a solution to this problem, aiming to introduce a level of transparency that empowers consumers and instills trust in the valuation process. Looking ahead, our project envisions a future where predictive modeling becomes synonymous with accuracy, transparency, and fairness in the automotive valuation landscape. We embark on this journey to revolutionize the field, providing a solution that not only addresses historical inefficiencies but sets new standards for predictive modeling in the automotive industry.

1.2 OBJECTIVE

The objective of the 'Car Quintessence' project is to develop a user-friendly web application that leverages a trained Random Forest Regressor model. The application allows users to input car parameters, such as Km driven, model, year, fuel type, etc and receive accurate predictions of car prices. Utilizing a dataset collected from reputable sources, the goal is to provide individuals with a simple and effective tool for estimating car values based on machine learning insights, enhancing the user experience in the process.

1.3 Existing System

In the existing system, shop owners are implementing a car price prediction system using the k-means algorithm. K-means is a popular clustering algorithm that helps identify groups or clusters within a dataset based on similarities in features. By applying the k-means algorithm to a dataset of car attributes such as brand, model, year, mileage, and condition, the system can segment cars into distinct groups. This clustering enables the system to analyze the characteristics of each group and predict the price range for a given car based on its attributes.

1.3.1 Existing System Disadvantages:

- Less accuracy
- Difficult to predict
- With global clusters, it doesn’t work well

1.4 LITERATURE SURVEY:

Title: Predicting the price of used cars using machine learning techniques.

Author: Pudaruth

Year: 2017

Description: According to a World Economic Forum’s report, AI-enabled automation will generate 133 million new jobs globally by 2022. And in India itself, the demand for AI talent pool is expected to skyrocket with the government’s steps towards digitization, and multiple organizations accelerating their digital transformation initiatives. Are you ready to ride the wave? The BITS Pilani 11-month online PG Programme in AI & ML is designed to help working professionals like you to develop an understanding of AI & ML and its various building blocks.

Title: An expert system of price forecasting for used cars using adaptive neurofuzzy inference.

Year: 2009

Author: Wu, J. D., Hsu, C. C., & Chen, H. C.

Description: An expert system for used cars price forecasting using adaptive neuro-fuzzy inference system (ANFIS) is presented in this paper. The proposed system consists of three parts: data acquisition system, price forecasting algorithm and performance analysis. The effective factors in the present system for price forecasting are simply assumed as the mark of the car, manufacturing year and engine style. Further, the equipment of the car is considered to raise the performance of price forecasting. In price forecasting, to verify the effect of the proposed ANFIS, a conventional artificial neural network (ANN) with back-propagation (BP) network is compared with proposed ANFIS for price forecast because of its adaptive learning capability. The ANFIS includes both fuzzy logic qualitative approximation and the adaptive neural network capability. The experimental result pointed out that the proposed expert system using ANFIS has more possibilities in used car price forecasting.

Title: New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit. In Measuring Technology and Mechatronics Automation

Year: 2011

Author: Gongqi, S., Yansong, W., &Qiang, Z.

Description: A new model for predicting the residual value of the private used car with various conditions, such as manufacturer, mileage, time of life, etc., was developed in this paper. A comprehensive method combined by the BP neural network and nonlinear curve fit was introduced for optimizing the model due to its flexible nonlinearity. Firstly, some distribution curves of residual value of the used cars were analyzed in time domain. Then, the BP neural network (NN) was established and used to extract the feature of the distribution curves in various conditions. A set of schemed data was used to train the NN and reached the training goal. Finally, the schemed data as inputs and the NN outputs were organized for nonlinear curve fit. Conclusion was drawn that the newly proposed model is feasible and accurate for residual value prediction of the used cars with various conditions.

Title: PIN Optimal Distribution of Auction Vehicles System: Applying Price Forecasting, Elasticity Estimation, and Genetic Algorithms to Used-Vehicle Distribution. Marketing Science

Year:2009

Author: Du, J., Xie, L., & Schroeder

Description: In addition to retailing new vehicles, automotive manufacturers in the United States sell millions of vehicles through leasing and to fleet customers every year. The majority of these vehicles are returned to the automotive manufacturers at the end of the contracted term and must be "remarketed." In 2007, about 10 million used vehicles were sold at more than 400 auctions in the United States. Large consigners face decisions every day about when, where, and at what price to offer these vehicles, which has significant financial implications for their profitability. To address the challenges of the distribution process, (), a division of J.D. Power and Associates, developed the PIN Optimal Distribution of Auction Vehicles System (ODAV), an automated decision optimization system that helps remarketers maximize profits through the most advantageous distribution of their auction vehicles. At the core of the system is a combination of three models that determine the distribution of the vehicles on a daily basis: a nearest neighbor linear regression model for short-term auction price forecasting; an autoregressive integrated moving average time-series analysis model for volume-price elasticity; and a genetic algorithm optimizer for vehicle distribution. Since its launch in 2003, PIN has been providing ODAV services on a daily basis, and to date, more than two million vehicles have been distributed through this system. In this paper, we will describe the PIN ODAV System, its implementation, and the business impact by using as an example the experience with our largest client, Chrysler Group LLC.

1.5 PROPOSED SYSTEM

In this system, we implement an effective car price prediction system using the Random Forest Algorithm, offering a seamless user experience with both CSV file uploads and manual entries. Beyond its standalone capabilities, our system features web integration, allowing users to access the prediction service through a user-friendly web interface. This integration enhances accessibility, enabling users to input car parameters effortlessly and receive accurate predictions in real-time. Whether through direct data upload or interactive manual entry, the system's web integration ensures a versatile and dynamic platform for efficient car valuation.

1.5.1 PROPOSED SYSTEM ADVANTAGES:

- Easy to predict the car prices
- High accuracy rate (over 90 %)
- Enhanced Precision and Accuracy
- User-Friendly Web Integration
- Adaptability to Diverse Data Sources
- Real-Time Predictions
- Transparency and Trust

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

In the context of our car price prediction project, we recognize the pivotal impact of fuel type and consumption on the vehicle's price, given the dynamic nature of fuel prices. Our comprehensive approach extends beyond these factors, considering a myriad of features such as exterior color, door number, transmission type, dimensions, safety features, air conditioning, interior specifications, and the presence of navigation systems. Acknowledging the multifaceted nature of these attributes, we meticulously applied various methods and advanced techniques to enhance the precision of our used car price prediction model. By leveraging a combination of machine learning algorithms and a diverse dataset, our project aims to capture intricate relationships among these features, ensuring that our predictions provide unparalleled accuracy in reflecting the true market value of used cars.

2.2 METHODOLOGIES:

2.2.1 MODULE NAMES:

- Data Collection
- Data Pre-processing
- Machine Learning Model Building
- Web Interface Development
- Flask Integration
- Testing & Validation
- Deployment

2.3 MODULE DESCRIPTION :-

DATA COLLECTION:-To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be composed of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data.

DATA PRE-PROCESSING:-In data pre-processing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project.

MACHINE LEARNING MODEL BUILDING:-The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions, In these we use 90% of data for training and remaining 10 % of data for testing

WEB INTERFACE DEVELOPMENT:-Our web interface development prioritizes an intuitive platform using HTML5, CSS3, and JavaScript (ES6) for seamless input of car parameters. The design, featuring a clean layout and responsive elements, enhances accessibility, while interactive features provide a dynamic user experience. Optimized for Google Chrome, the interface seamlessly integrates with the Flask backend, ensuring efficient communication with the machine learning model for accurate car price predictions.

FLASK INTEGRATION :-In the Flask integration module, we leverage the Flask framework to seamlessly connect the web interface with the machine learning model. The Flask application, hosted on the backend, handles HTTP requests from the frontend, facilitating a smooth flow of data between the user interface and the model. Through the Flask routes, user inputs are processed, and predictions from the Random Forest Algorithm are returned to the interface in real-time. This integration ensures a robust and responsive connection, allowing users to experience efficient and accurate car price predictions within the web application.

TESTING & VALIDATION :-The testing and validation module rigorously evaluates the machine learning model, ensuring accuracy and reliability in predicting car prices. Utilizing diverse validation techniques, this phase verifies the model's performance under different scenarios, providing confidence in its predictive capabilities. Thorough testing ensures the robustness of our system, enhancing its reliability for real-world deployment.

DEPLOYMENT :-The deployment phase finalizes the project, ensuring the entire system operates seamlessly in a production environment. This stage encompasses the integration of all modules, making the car price prediction system accessible to end-users. By optimizing performance and resolving any potential deployment issues, we ensure a reliable and user-ready application for predicting car prices.

2.4 TECHNIQUE USED OR ALGORITHM USED:-

Single machine learning classifier approach that has been used in all previous researches was also tested in this research. The whole data set collected in this research has been split into training (90%) and testing (10%) subsets and Random Forest classifier model is built.Random forest (RF) also known as random decision forest belongs to the category of ensemble methods. RF can be used for classification and regression problems. The algorithm was developed by Ho as an improvement for over fitting of the decision tree algorithms.Our car price prediction system leverages the Random Forest Algorithm, a robust and versatile approach known for its efficacy in regression tasks. The algorithm undergoes training on a split dataset, using a majority for learning patterns and a remaining portion for rigorous testing, ensuring its ability to generalize to new data.Remarkably, the Random Forest achieves an impressive 92 percent accuracy, attributed to its ensemble nature, which minimizes overfitting and enhances overall robustness. The decision to choose Random Forest over other algorithms was intentional, driven by its suitability for handling diverse features in our dataset and its excellent generalization capabilities.This deliberate choice has not only contributed to the model's high accuracy but also ensures its reliability in providing precise predictions for a wide range of car configurations in our car price prediction system.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

These are the requirements for doing the project. Without using these tools and software's we can't do the project. So we have two requirements to do the project.
They are-

1. Hardware Requirements.
2. Software Requirements.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

● PROCESSOR	:	Intel Core i3-8100 (Quad Core CPU or Higher)
● RAM	:	8 GB DDR4 2666Mt/s or higher
● STORAGE	:	256 GB SATA HDD at 7200 RPM (preferably PCIe Gen3 SSD for faster data access)
● GPU	:	NVIDIA GeForce GTX 1000 series CUDA-enabled GPU for faster model training (Optional but recommended for large datasets)

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

● Operating System:	Windows 7/ MacOS X 10.13 or higher/ Ubuntu 18.04 or higher
● IDE:	Microsoft Visual Studio /Anaconda/Jupyter/ Spyder (IDE of choice)
● Python Environment:	Python 3 (version 3.9.18)
● Libraries for Data Processing:	Pandas (version 2.1.3) NumPy (version 1.26.2)
● Machine Learning Libraries:	Scikit-learn (version 0.23.2 or 0.24.2)
● Database:	Local Storage (No Separate Database)
● Framework:	Flask (version 2.2.2)
● Frontend (Web Application):	HTML5, CSS3, JavaScript (ES6) (Preferably Google Chrome Browser)

3.4 FUNCTIONAL REQUIREMENTS

In this project, a functional requirement introduces an regressor method for car price prediction, focusing specifically on the implementation of the Random Forest Algorithm.Our model utilizes a diverse set of parameters such as car specifications, features, and historical data. The Random Forest Algorithm is then applied to these parameters, leveraging its ensemble of decision trees to capture complex relationships within the dataset. Through this iterative refinement process, the model achieves high accuracy in predicting car prices, showcasing the effectiveness of Random Forest in providing nuanced and precise estimations based on a comprehensive set of features.

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

USABILITY:-The system is designed with a completely automated process hence there is no or less user intervention.

RELIABILITY:-The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

PERFORMANCE:-This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on the client system within very less time.

SUPPORTABILITY:-The system is designed to be cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having python3, built into the system.

IMPLEMENTATION:-The system is implemented in a web environment using flask framework. Flask handles communication through HTTP request-response cycles. When a user interacts with the web interface, Flask processes the input through predefined routes, triggering HTTP requests. These requests are then handled by Flask, facilitating communication with the machine learning model, and the corresponding HTTP responses deliver the predicted car prices back to the user interface in real-time.

CHAPTER 4

DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of the project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished products.

UML Diagrams

4.1.1 USE CASE DIAGRAM:

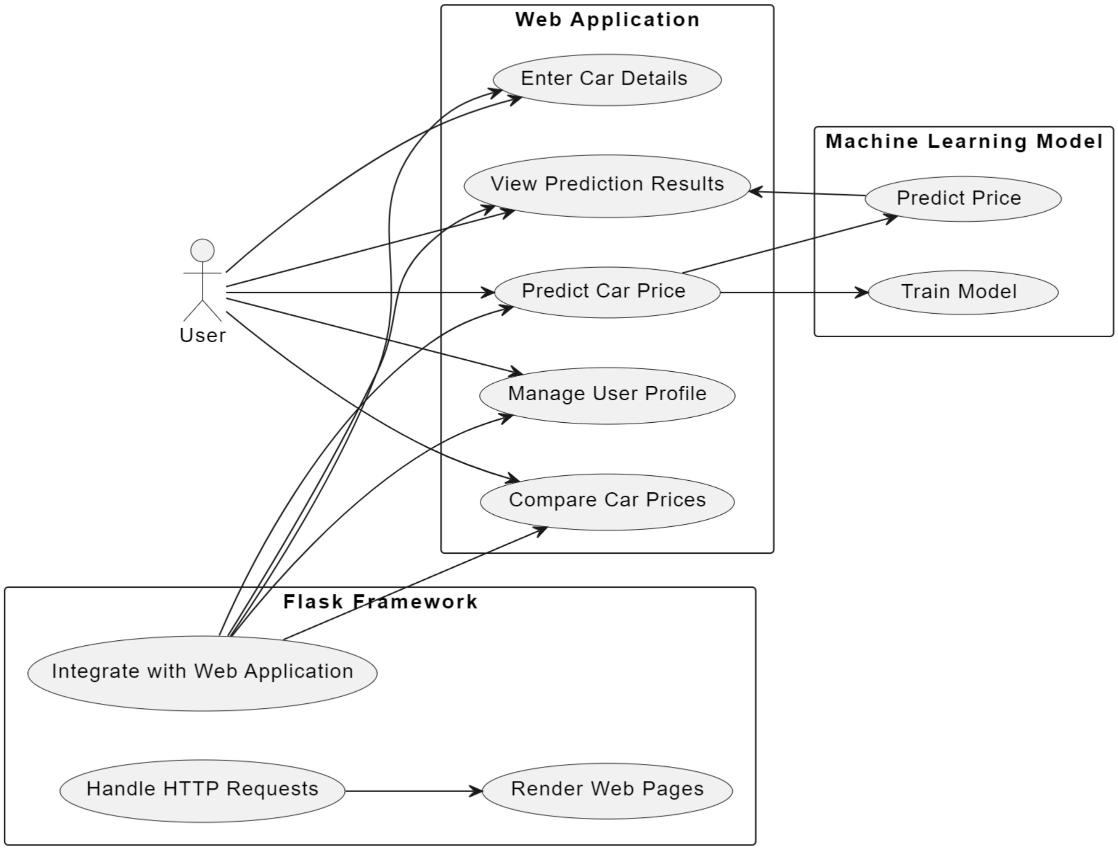
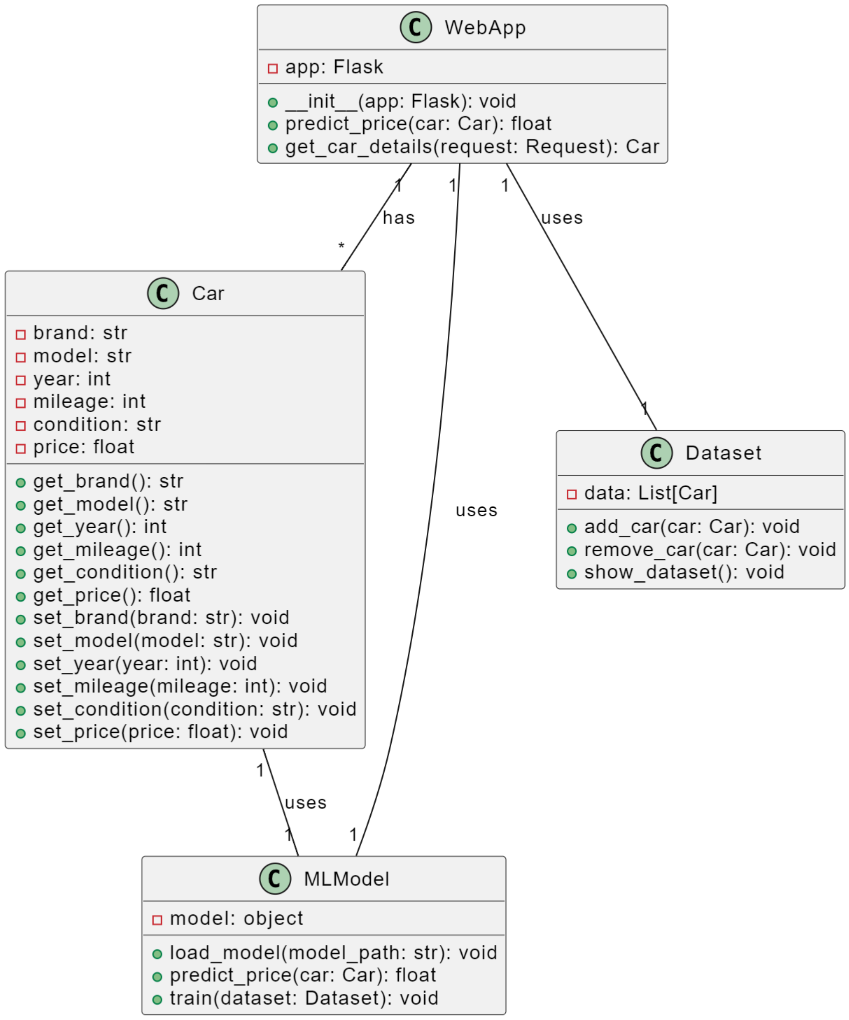


FIG 4.1.1 (Use Case Diagram for car price prediction)

EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

4.1.2 CLASS DIAGRAM:



WebApp
app: Flask
__init__(app: Flask): void
predict_price(car: Car): float
get_car_details(request: Request): Car
1
1
1
has
*
uses
Car
brand: str
model: str
year: int
mileage: int
condition: str
price: float
get_brand(): str
get_model(): str
get_year(): int
get_mileage(): int
get_condition(): str
get_price(): float
set_brand(brand: str): void
set_model(model: str): void
set_year(year: int): void
set_mileage(mileage: int): void
set_condition(condition: str): void
set_price(price: float): void
uses
1
Dataset
data: List[Car]
add_car(car: Car): void
remove_car(car: Car): void
show_dataset(): void
1
uses
1
1
MLModel
model: object
load_model(model_path: str): void
predict_price(car: Car): float
train(dataset: Dataset): void

FIG 4.1.2 (Class Diagram for car price prediction)

EXPLANATION:

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

4.1.3 OBJECT DIAGRAM:

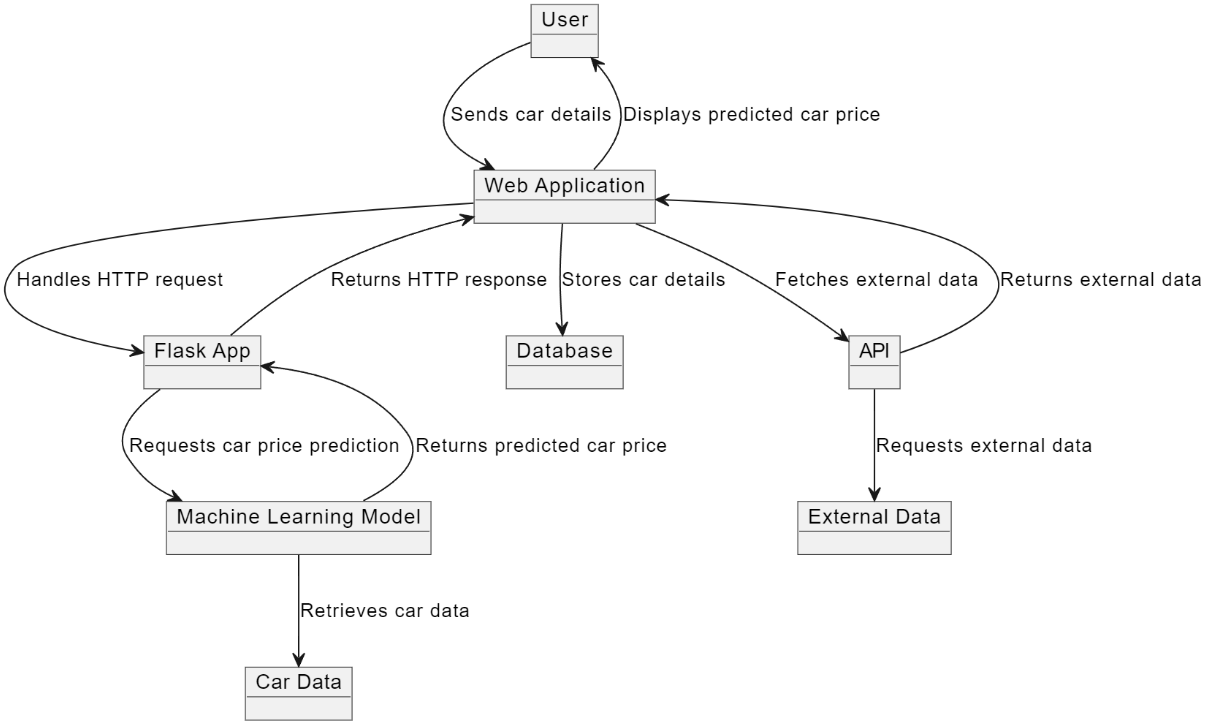


FIG 4.1.3 (Object Diagram for car price prediction)

EXPLANATION:

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

4.1.4 STATE DIAGRAM:

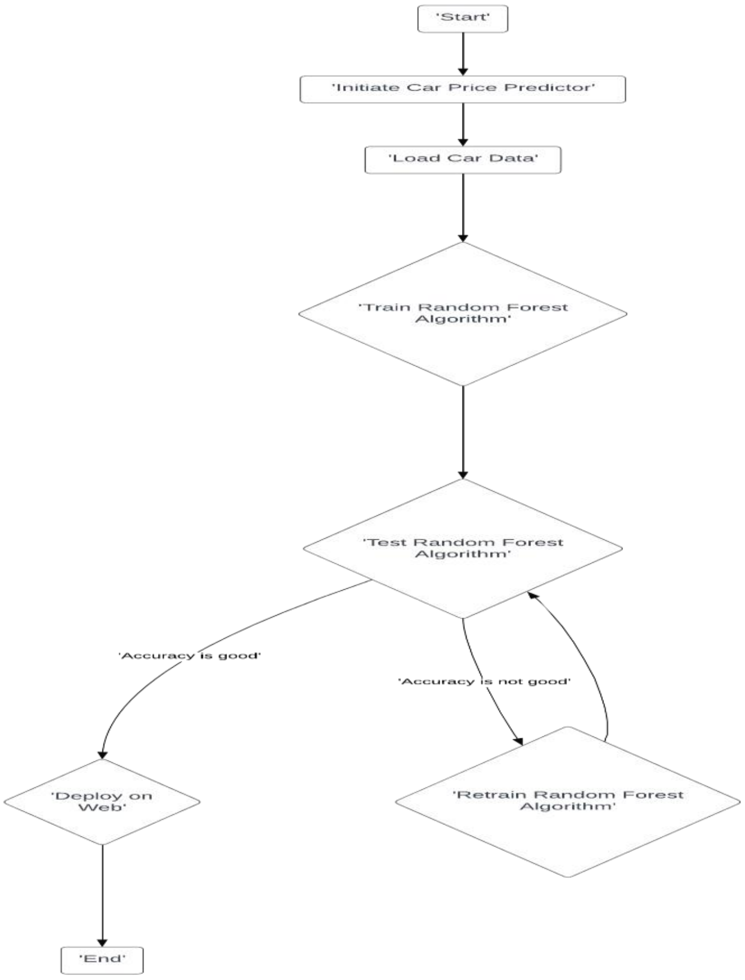

'Start'
'Initiate Car Price Predictor'
'Load Car Data'
'Train Random Forest Algorithm'
'Test Random Forest Algorithm'
'Accuracy is good'
'Accuracy is not good'
'Deploy on Web'
'Retrain Random Forest Algorithm'
'End'

FIG 4.1.4 (State Chart Diagram for car price prediction)

EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.

4.1.5 ACTIVITY DIAGRAM:

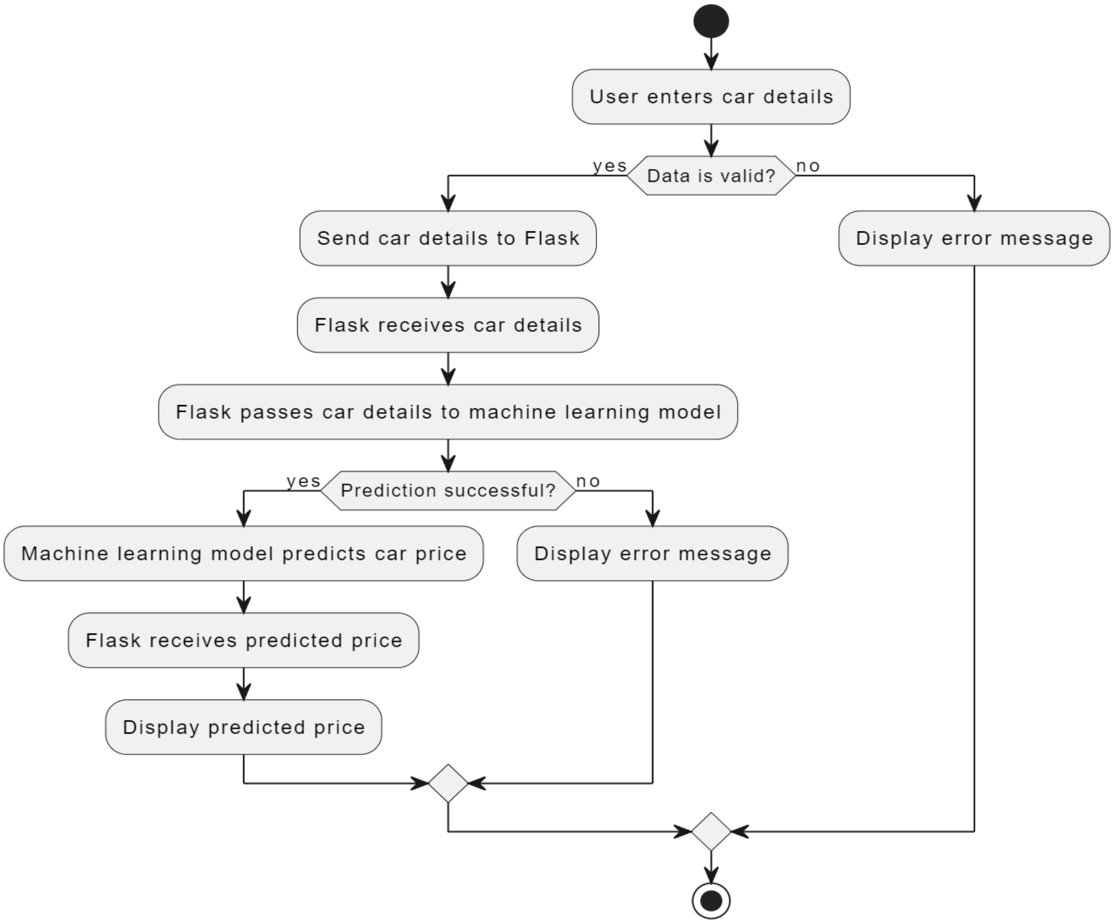

User enters car details
yes
Data is valid?
no
Send car details to Flask
Display error message
Flask receives car details
Flask passes car details to machine learning model
yes
Prediction successful?
no
Machine learning model predicts car price
Display error message
Flask receives predicted price
Display predicted price

FIG 4.1.5 (Activity Diagram for car price prediction)

EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

4.1.6 SEQUENCE DIAGRAM:

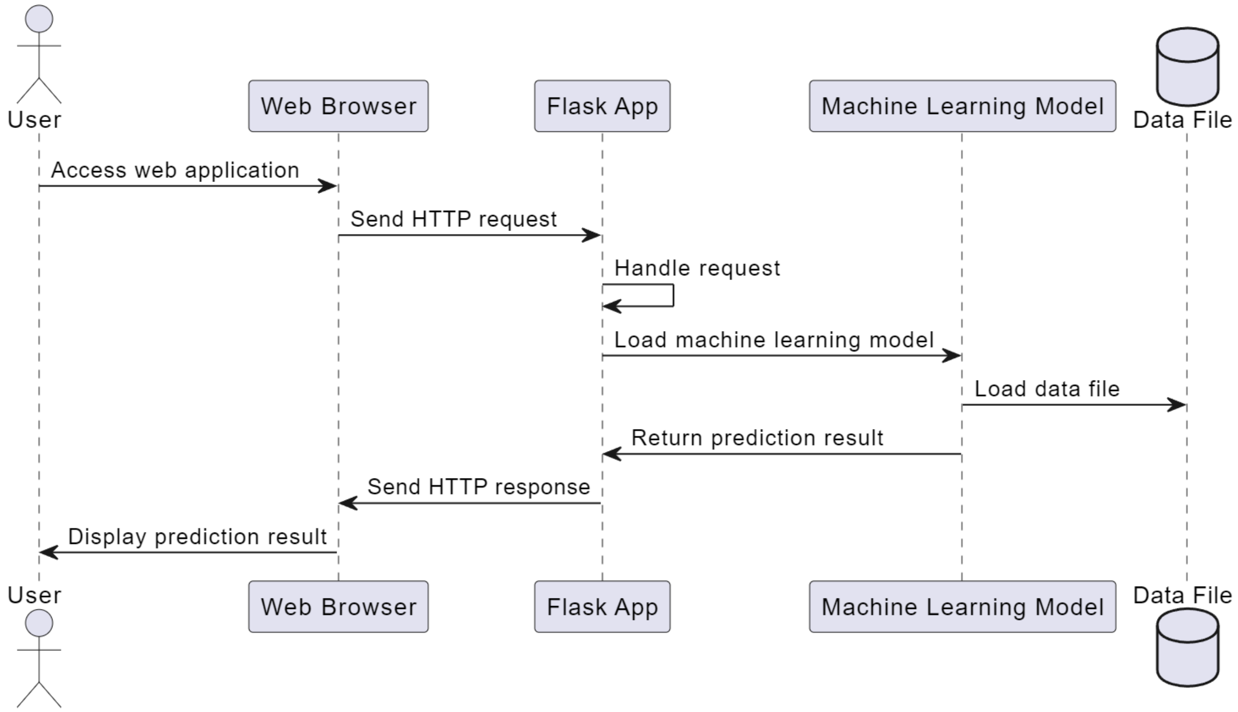


FIG 4.1.6 (Sequence Diagram for car price prediction)

EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

4.1.7 COLLABORATION DIAGRAM:

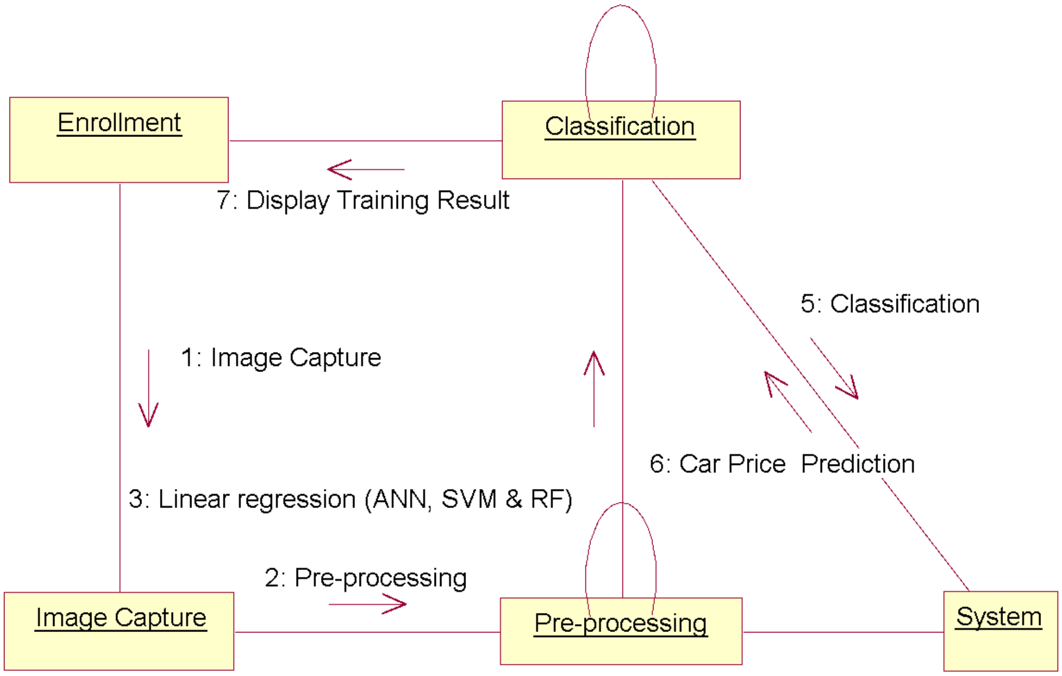


FIG 4.1.7(Collaboration Diagram for car price prediction)

EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

4.1.8 COMPONENT DIAGRAM:

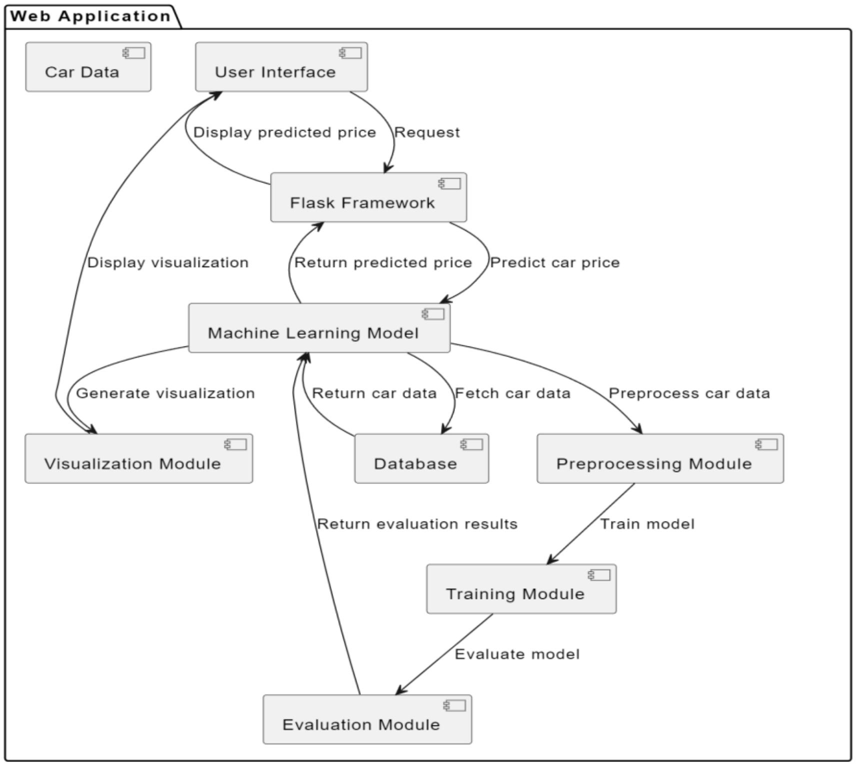


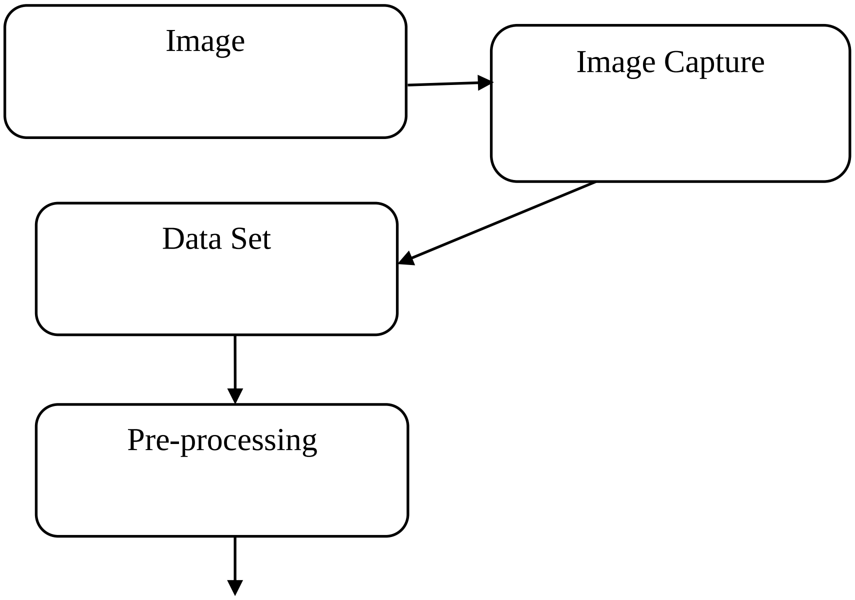
FIG 4.1.8 (Component Diagram for car price prediction)

EXPLANATION:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

4.1.9 DATA FLOW DIAGRAM:

Level-0:



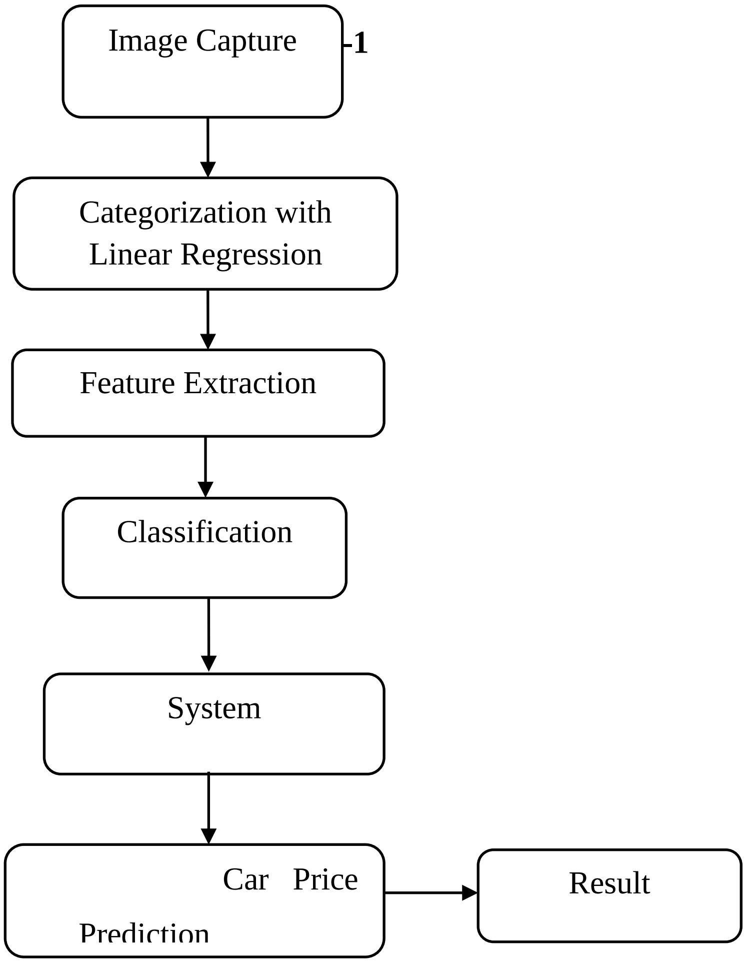


FIG 4.1.9 (Data Flow Diagram for car price prediction)

EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

4.1.10 Deployment Diagram

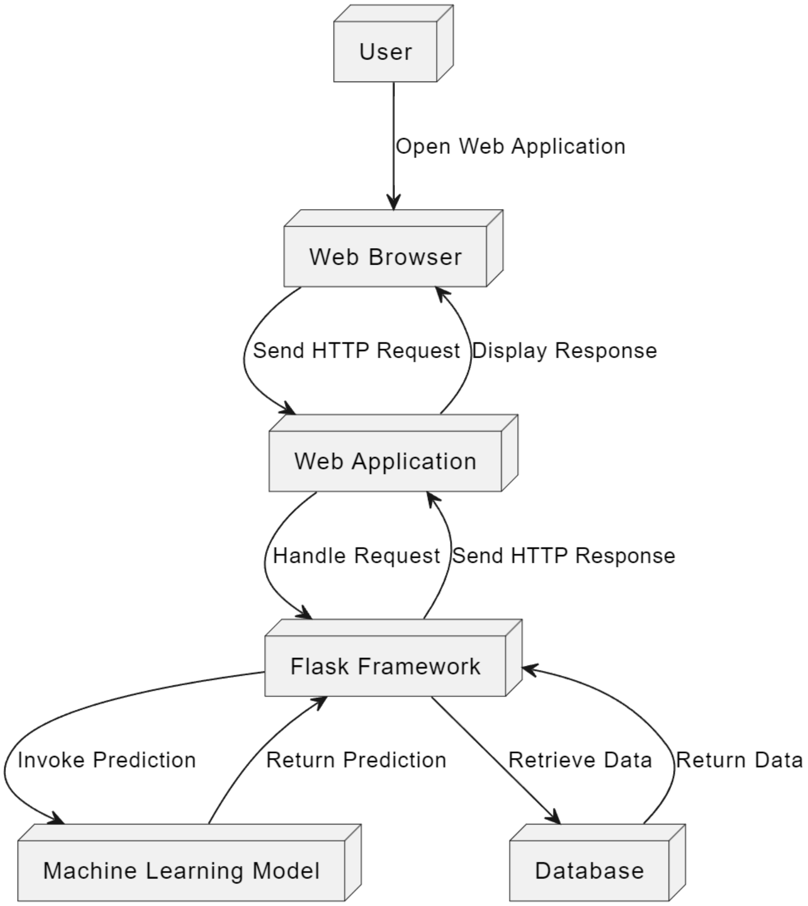


FIG 4.1.10(Deployment Diagram for car price prediction)

EXPLANATION:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

4.1.11 SYSTEM ARCHITECTURE:

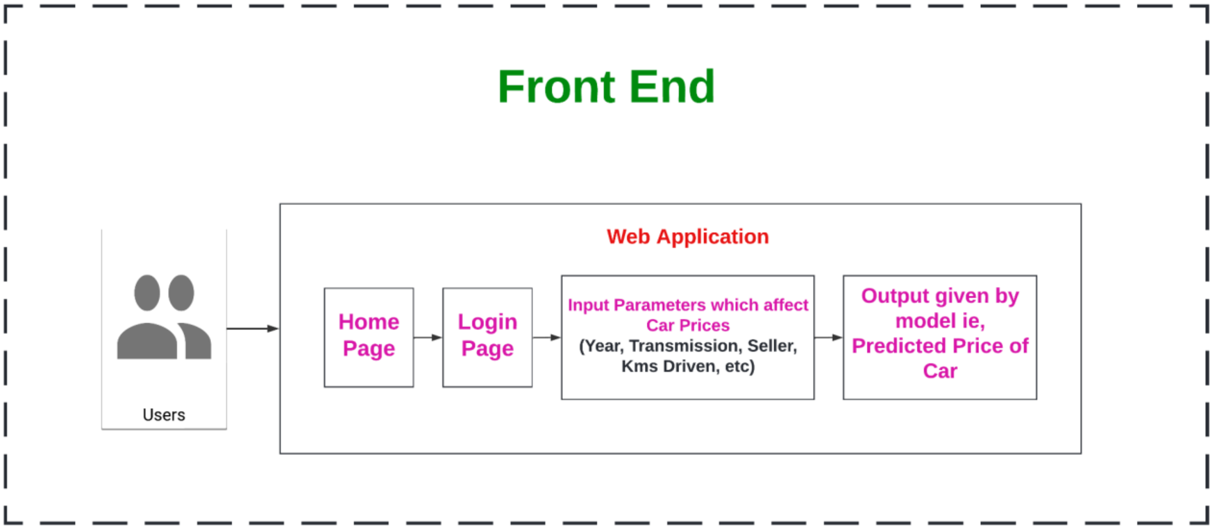

Front End
Users
Web Application
Home Page
Login Page
Input Parameters which affect Car Prices
(Year, Transmission, Seller, Kms Driven, etc)
Output given by model ie, Predicted Price of Car

FIG 4.1.11 (System Architecture Front End Diagram for car price prediction)

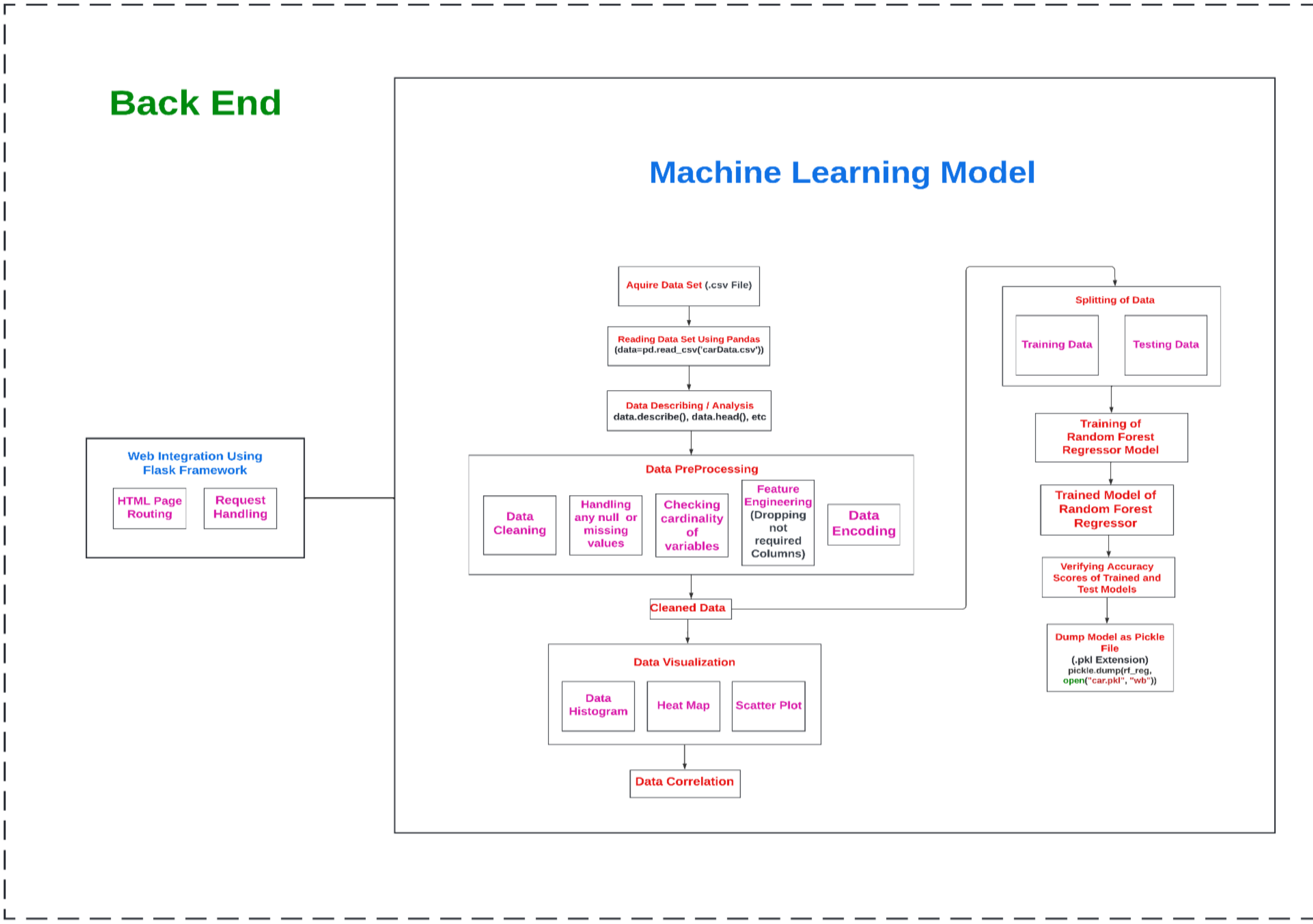

Back End
Web Integration Using Flask Framework
HTML Page Routing
Request Handling
Machine Learning Model
Aquire Data Set (.csv File)
Reading Data Set Using Pandas (data=pd.read_csv('carData.csv'))
Data Describing / Analysis data.describe(), data.head(), etc
Data PreProcessing
Data Cleaning
Handling any null or missing values
Checking cardinality of variables
Feature Engineering (Dropping not required Columns)
Data Encoding
Cleaned Data
Data Visualization
Data Histogram
Heat Map
Scatter Plot
Data Correlation
Splitting of Data
Training Data
Testing Data
Training of Random Forest Regressor Model
Trained Model of Random Forest Regressor
Verifying Accuracy Scores of Trained and Test Models
Dump Model as Pickle File (.pkl Extension) pickle.dump(rf_reg, open("car.pkl", "wb"))

FIG 4.1.11 (System Architecture Back End Diagram for car price prediction)

CHAPTER 5

DEVELOPMENT TOOLS

5.1 MACHINE LEARNING:-

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn patterns and make predictions or decisions without being explicitly programmed. It is based on the idea that systems can learn from data, identify patterns, and improve their performance over time.

There are several types of machine learning approaches:

1. SUPERVISED LEARNING: The algorithm is trained on a labeled dataset, where it learns the relationship between input features and corresponding output labels. The goal is to make accurate predictions on new, unseen data.

2.UNSUPERVISED LEARNING: The algorithm is given data without explicit instructions on what to do with it. It must discover patterns or relationships within the data, often through techniques like clustering or dimensionality reduction.

3.REINFORCEMENT LEARNING: The algorithm learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The goal is to learn the optimal actions to take in different situations to maximize cumulative reward.Machine learning is applied across various domains, including image and speech recognition, natural language processing, recommendation systems, autonomous vehicles, and predictive analytics. It has the potential to automate complex tasks, improve decision-making processes, and uncover insights from vast amounts of data. As technology continues to advance, machine learning plays a central role in driving innovation and transforming the way we approach problem-solving in diverse fields.

5.2 PYTHON LIBRARIES FOR MACHINE LEARNING:-

Python has become a dominant programming language in the field of machine learning (ML) due to its simplicity, versatility, and the availability of robust libraries specifically designed for ML tasks. Here are some key libraries in Python for machine learning:**1. NUMPY-**NumPy is a fundamental library for numerical operations in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is the foundation for many other machine learning libraries.

2. PANDAS-Pandas is widely used for data manipulation and analysis. It offers data structures like DataFrame, which is especially useful for handling structured data. Pandas simplifies tasks such as cleaning, transforming, and exploring datasets before feeding them into machine learning models.

3. SCIKIT-LEARN:-Scikit-learn is a comprehensive machine learning library that provides simple and efficient tools for data analysis and modeling. It includes various algorithms for classification, regression, clustering, and dimensionality reduction. Scikit-learn is designed to work seamlessly with other scientific libraries like NumPy and SciPy.4.

4. MATPLOTLIB AND SEABORN-Matplotlib and Seaborn are essential for data visualization in Python. Matplotlib offers a wide range of static, animated, and interactive plots, while Seaborn provides a high-level interface for creating attractive and informative statistical graphics.

5. TENSORFLOW AND PYTORCH-TensorFlow and PyTorch are deep learning frameworks that facilitate the creation of neural networks and deep learning models. They offer flexibility and scalability for building and training complex models, making them popular choices for deep learning tasks.

6. KERAS-Keras is a high-level neural networks API that runs on top of TensorFlow and Theano. It simplifies the process of building and training neural networks by providing a user-friendly interface, making it an excellent choice for beginners in deep learning.

7. NLTK AND SPACY- Natural Language Toolkit (NLTK) and SpaCy are libraries for natural language processing (NLP). NLTK is widely used for tasks such as tokenization, stemming, and part-of-speech tagging, while SpaCy focuses on providing efficient and production-ready tools for NLP.

8. STATSMODELS-Statsmodels is a library for estimating and testing statistical models. It is particularly useful for traditional statistical analyses, hypothesis testing, and regression modeling.

9. XGBOOST AND LIGHTGBM-XGBoost and LightGBM are gradient boosting frameworks that excel in handling tabular data and structured datasets. They are widely used for classification and regression tasks, offering high performance and efficiency.

10. OPENCV-OpenCV is a computer vision library that provides a wide range of tools for image and video processing. It is essential for tasks such as object detection, image recognition, and feature extraction.

These libraries collectively form a powerful ecosystem in Python for machine learning, covering a broad spectrum of tasks from data preprocessing and analysis to building and deploying complex machine learning models.

5.3 FEATURES OFFERED BY PYTHON FOR MACHINE LEARNING

Python and machine learning (ML) have become closely intertwined, and Python is now the de facto language for many aspects of ML development. Several factors contribute to this strong relationship:

1. EASE OF LEARNING AND USE:-Python's syntax is clear and readable, making it accessible for beginners. This ease of learning and use has contributed to Python's popularity in ML, allowing practitioners to focus on solving problems rather than wrestling with the complexities of the language.

2. EXTENSIVE LIBRARIES: - Python boasts a rich ecosystem of libraries and frameworks that cater specifically to machine learning and data science. Libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and PyTorch provide efficient tools for various ML tasks, streamlining development and research processes.

3. COMMUNITY SUPPORT: - Python has a large and active community of developers. This community support ensures that there is a wealth of resources, tutorials, and documentation available for anyone working on ML projects in Python. Collaboration and knowledge sharing are facilitated through platforms like GitHub and forums.

4. VERSATILITY:- Python is a versatile language, capable of handling diverse tasks. It seamlessly integrates with other languages and technologies, making it a preferred choice for end-to-end ML development—from data preprocessing and model training to deployment.

5. DATA SCIENCE AND VISUALIZATION: - Python excels in data science and visualization, critical components of the ML workflow. Libraries like Pandas and Matplotlib provide tools for handling and exploring datasets, and Jupyter Notebooks allow for interactive and iterative development, making it easy to visualize results and share findings.

6. DEEP LEARNING FRAMEWORKS:- Major deep learning frameworks, such as TensorFlow and PyTorch, have official Python APIs, making Python the primary language for developing and implementing complex neural network architectures. Keras, a high-level neural networks API, is also seamlessly integrated with Python.

7. RAPID PROTOTYPING: - Python's dynamic nature and interpreted paradigm facilitate rapid prototyping. ML researchers and developers can quickly experiment with different algorithms, models, and parameters, accelerating the iterative development process.

8. BIG DATA INTEGRATION:- Python integrates well with big data processing frameworks like Apache Spark. This enables ML practitioners to work with large-scale datasets and distributed computing environments seamlessly.

9. SUPPORT FOR MULTIPLE PARADIGMS: - Python supports various programming paradigms, including procedural, object-oriented, and functional programming. This flexibility allows developers to choose the paradigm that best suits their ML project requirements.

10. DEPLOYMENT AND INTEGRATION:-Python's flexibility extends to deployment and integration. Models developed in Python can be integrated into various production environments, and tools like Flask and Django facilitate the development of web-based applications with ML capabilities.

In summary, Python's simplicity, extensive libraries, community support, and adaptability make it an ideal language for machine learning development. Its role spans from initial data exploration to the implementation of complex models, and its integration with deep learning frameworks has solidified its position as the language of choice for the ML community.

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

6.2 IMPLEMENTATION & SOURCE CODE

```
#importing the libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

#reading the dataset

data=pd.read_csv('car data.csv')

print(data.shape)

data.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0

2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

Checking if there are any missing values

```
data.isnull().sum()
```

```
Car_Name        0

Year            0

Selling_Price    0

Present_Price    0

Kms_Driven       0

Fuel_Type        0

Seller_Type      0

Transmission     0

Owner            0

dtype: int64
```

Checking cardinality of independent categorical variables in the dataset

```
print('Unique elements in Seller_Type are',data['Seller_Type'].unique())
print('Unique elements in Fuel_Type are',data['Fuel_Type'].unique())
print('Unique elements in Transmission are',data['Transmission'].unique())
print('Unique elements in Owner are',data['Owner'].unique())
print('Unique elements in Year are',data['Year'].unique())
```

```
Unique elements in Seller_Type are ['Dealer' 'Individual']
Unique elements in Fuel_Type are ['Petrol' 'Diesel' 'CNG']
Unique elements in Transmission are ['Manual' 'Automatic']
Unique elements in Owner are [0 1 3]
Unique elements in Year are [2014 2013 2017 2011 2018 2015 2016 2009 2010 2012 2003 2008 2006 2005
 2004 2007]
```

```
print('Unique elements in Car_Name are',data['Car_Name'].nunique())
#98 unique elements
#so, rather than encoding it, we can just drop this columbn as it doesn' make sense
```

```
Unique elements in Car_Name are 98
```

```
data.describe()
```

Year	Selling_Price	Present_Price	Kms_Driven	Owner	
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000

75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

Dropping the Car_Name Column

```
dataset=data[['Year','Selling_Price','Present_Price','Kms_Driven','Fuel_Type','Seller_Type','Transmission','Owner']]
dataset.head()
```

Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

```
dataset['Present_Year']=2022
dataset['Number_of_Years_Old']=dataset['Present_Year']- dataset['Year']
dataset.head()
```

Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Present_Year	Number_of_Years_Old	
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0	2022	8

1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0	2022	9
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0	2022	5
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0	2022	11
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0	2022	8

So, we can now safely drop 'Year' and 'Present_Year' columns

```
dataset.drop(labels=['Year', 'Present_Year'],axis=1,inplace=True)
dataset.head()
```

Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Number_of_Years_Old	
0	3.35	5.59	27000	Petrol	Dealer	Manual	0	8
1	4.75	9.54	43000	Diesel	Dealer	Manual	0	9
2	7.25	9.85	6900	Petrol	Dealer	Manual	0	5
3	2.85	4.15	5200	Petrol	Dealer	Manual	0	11
4	4.60	6.87	42450	Diesel	Dealer	Manual	0	8

Encoding the Categorical Variables

```
#select categorical variables from then dataset, and then implement categorical encoding for nominal variables
Fuel_Type=dataset[['Fuel_Type']]
Fuel_Type=pd.get_dummies(Fuel_Type, drop_first=True)
```

```
Seller_Type=dataset[['Seller_Type']]
Seller_Type=pd.get_dummies(Seller_Type, drop_first=True)

Transmission=dataset[['Transmission']]
Transmission=pd.get_dummies(Transmission, drop_first=True)

dataset=pd.concat([dataset,Fuel_Type, Seller_Type, Transmission], axis=1)

dataset.drop(labels=['Fuel_Type', 'Seller_Type', 'Transmission'], axis=1, inplace=True)

dataset.head()
```

```
   Selling  Present  Kms_    Ow   Number_of_  Fuel_Typ  Fuel_Typ  Seller_Type_  Transmissio
   _Price   _Price   Driven  ner  Years_Old   e_Diesel  e_Petrol  Individual    n_Manual

0  3.35     5.59     27000   0    8           False     True      False         True

1  4.75     9.54     43000   0    9           True      False     False         True

2  7.25     9.85     6900    0    5           False     True      False         True

3  2.85     4.15     5200    0    11          False     True      False         True

4  4.60     6.87     42450   0    8           True      False     False         True
```

```
dataset.columns
```

```
Index(['Selling_Price', 'Present_Price', 'Kms_Driven', 'Owner',
       'Number_of_Years_Old', 'Fuel_Type_Diesel', 'Fuel_Type_Petrol',
       'Seller_Type_Individual', 'Transmission_Manual'],
      dtype='object')
```

```
# Dataset Correlation
dataset.corr()
```

```
         Sellin   Prese    Kms_           Number_of  Fuel_Ty   Fuel_Ty   Seller_Typ
         g_Pri    nt_Pri   Drive   Ow     _Years_Ol  pe_Dies   pe_Petr   e_Individu   Transmissi
         ce       ce       n       ner    d          el        ol        al           on_Manual
```

Selling_Price	1.000000	0.878983	0.029187	-0.088344	-0.236141	0.552339	-0.540571	-0.550724	-0.367128
Present_Price	0.878983	1.000000	0.203647	0.008057	0.047584	0.473306	-0.465244	-0.512030	-0.348715
Kms_Driven	0.029187	0.203647	1.000000	0.089216	0.524342	0.172515	-0.172874	-0.101419	-0.162510
Owner	-0.088344	0.008057	0.089216	1.000000	0.182104	-0.053469	0.055687	0.124269	-0.050316
Number_of_Years_Old	-0.236141	0.047584	0.524342	0.182104	1.000000	-0.064315	0.059959	0.039896	-0.000394
Fuel_Type_Diesel	0.552339	0.473306	0.172515	-0.053469	-0.064315	1.000000	-0.979648	-0.350467	-0.098643
Fuel_Type_Petrol	-0.540571	-0.465244	-0.172874	0.055687	0.059959	-0.979648	1.000000	0.358321	0.091013
Seller_Type_Individual	-0.550724	-0.512030	-0.101419	0.124269	0.039896	-0.350467	0.358321	1.000000	0.063240
Transmission_Manual	-0.367128	-0.348715	-0.162510	-0.050316	-0.000394	-0.098643	0.091013	0.063240	1.000000

```
sell=dataset['Selling_Price']
dataset.drop(['Selling_Price'], axis=1, inplace=True)
dataset=dataset.join(sell)
dataset.head()
X=dataset.iloc[:,:-1]
y=dataset.iloc[:,-1]
```

```
X=dataset.iloc[:,:-1].values
y=dataset.iloc[:,-1].values
X
X
```

```
from sklearn.model_selection import train_test_split
 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Random Forest Regressor

```
from sklearn.ensemble import RandomForestRegressor
rf_reg = RandomForestRegressor(random_state=0)
rf_reg.fit(X_train, y_train)
y_pred = rf_reg.predict(X_test)

print("Random forest Score on Training set is",rf_reg.score(X_train, y_train))#Training Accuracy
 print("Random forest Score on Test Set is",rf_reg.score(X_test, y_test))#Testing Accuracy
```

Dump the model selected as a Pickle File

```
import pickle
pickle.dump(rf_reg, open("car.pkl", "wb"))

# load model from file
model = pickle.load(open("car.pkl", "rb"))

model.predict([[5.59, 27000, 0, 3, 0, 1, 0, 1]])
```

FIRST.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Car price prediction</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta content="Free HTML Templates" name="keywords">
    <meta content="Free HTML Templates" name="description">
    <!-- Favicon -->
    <link href="img/favicon.ico" rel="icon">
    <!-- Google Web Fonts -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500;600;700&family=Open+Sans:wght@400;600&display=swap" rel="stylesheet">
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
    <!-- Libraries Stylesheet -->
    <link href="../static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
    <!-- Customized Bootstrap Stylesheet -->
    <link href="../static/css/style.css" rel="stylesheet">
</head>
```

```
<body>
    <!-- Topbar Start -->
    <!-- Topbar End -->
    <!-- Navbar Start -->
    <div class="container-fluid p-0">
        <nav class="navbar navbar-expand-lg bg-white navbar-light py-3 py-lg-0 px-lg-5">
            <a href="index.html" class="navbar-brand ml-lg-3">
                <h1 class="m-0 text-uppercase text-primary"> Car Quintessence</h1>
            </a>
            <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse justify-content-between px-lg-3" id="navbarCollapse">
                <div class="navbar-nav mx-auto py-0">
                    <a href="{{ url_for('first')}}" class="nav-item nav-link active">Home</a>
                    <a href="{{ url_for('login')}}" class="nav-item nav-link">Login</a>
                </div>
            </div>
        </nav>
    </div>
    <!-- Navbar End -->
    <!-- Header Start -->
```

```
   <div class="jumbotron jumbotron-fluid position-relative overlay-bottom" style="margin-bottom:
90px;">

       <div class="container text-center my-5 py-5">

                                                    <h1 class="text-white display-1 mb-2"> Car Price Estimation</h1>

           <h1 class="text-white mt-4 mb-4">Using

Machine Learning   </h1>

           <div class="mx-auto mb-5" style="width: 100%; max-width: 600px;">

           </div>

       </div>

   </div>

   <!-- Header End -->

   <!-- Footer End -->

   <!-- Back to Top -->

   <!-- JavaScript Libraries -->

   <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

   <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>

   <script src="../static/lib/easing/easing.min.js"></script>

   <script src="../static/lib/waypoints/waypoints.min.js"></script>

   <script src="../static/lib/counterup/counterup.min.js"></script>

   <script src="../static/lib/owlcarousel/owl.carousel.min.js"></script>

   <!-- Template Javascript -->

   <script src="../static/js/main.js"></script>

</body> </html>
```

INDEX.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="utf-8">

   <title>Car price prediction</title>

   <meta content="width=device-width, initial-scale=1.0" name="viewport">

   <meta content="Free HTML Templates" name="keywords">

   <meta content="Free HTML Templates" name="description">

   <!-- Favicon -->

   <link href="../img/favicon.ico" rel="icon">

   <!-- Google Web Fonts -->

   <link rel="preconnect" href="https://fonts.gstatic.com">

   <link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500;600;700&family=Open+Sans:wght@4
00;600&display=swap" rel="stylesheet">

   <!-- Font Awesome -->

   <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet

   <!-- Libraries Stylesheet -->

   <link href="../static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

   <!-- Customized Bootstrap Stylesheet -->

   <link href="../static/css/car1.css" rel="stylesheet">

</head>
```

```
<body>
    <!-- Topbar Start -->
    <!-- Topbar End -->
    <!-- Navbar Start -->
    <div class="container-fluid p-0">
        <nav class="navbar navbar-expand-lg bg-white navbar-light py-3 py-lg-0 px-lg-5">
            <a href="index.html" class="navbar-brand ml-lg-3">
               <h1 class="m-0 text-uppercase text-primary"> Car Prices</h1>
            </a>
            <button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse justify-content-between px-lg-3" id="navbarCollapse">
                <div class="navbar-nav mx-auto py-0">
                     <a href="{{ url_for('first')}}" class="nav-item nav-link active">Home</a>
                    <a href="{{ url_for('login')}}" class="nav-item nav-link">Login</a>
                                                                                              <a href="{{ url_for('upload')}}" class="nav-item nav-link">Upload</a>
                                                                                               <a href="{{ url_for('index')}}" class="nav-item nav-link">Prediction</a>
                                                                                                            <a href="{{ url_for('performance')}}" class="nav-item nav-link">Performance</a>
                </div>
            </div>

        </nav>

    </div>

    <!-- Navbar End -->

    <!-- Header Start -->

    <div class="jumbotron jumbotron-fluid page-header position-relative overlay-bottom"
style="margin-bottom: 90px;">

        <div class="container text-center py-5">

           <h1 class="text-white mt-4 mb-4">Car Prices Detection  </h1>

            <div class="d-inline-flex text-white mb-5">

                <h1 class="text-white mt-4 mb-4">Prediction  </h1>

            </div>

        </div>

    </div>

    <!-- Header End -->

    <!-- Contact Start -->

    <div class="container-fluid py-5">

        <div class="container py-5">

             <center>  <h1 class="text-black mt-4 mb-4">Prediction</h1></center>



                                                                       <div class="container">

                                                                                             <center>  <h1 class="text-black mt-4 mb-4">Please
fill the parameters below and click on Selling Price button to check car price</h1></center>
```

```
 <div class="row" style="margin-left: 400px">

    <form action="\predict" method="post">

      <h3>Year</h3><input id="first" name="Year" placeholder="eg. like the year '2010'
"type="number " >

    <br>

    <br>

      <h3>Showroom Price(lakhs)</h3><input id="second" name="Present_Price" placeholder="eg.
'10.45' lakhs"required="required">

    <br>

    <br>

      <h3>Kilometers Driven</h3><input id="third" name="Kms_Driven" placeholder="eg. 10000 Km
driven before"required="required">

    <br>

    <br>

      <h3>Previous Owners</h3><input id="fourth" name="Owner" placeholder="0 or 1 or 2 Previous
Owners" required="required">

    <br>

    <br>

      <h3>Fuel Type</h3><select name="Fuel_Type_Petrol" id="fuel" required="required">

        <option value="Petrol">Petrol</option>

        <option value="Diesel">Diesel</option>

        <option value="Diesel">CNG</option>

      </select>

    <br>

    <br>
```

```
      <h3>Seller_Type</h3><select name="Seller_Type_Individual" id="resea" required="required">

        <option value="Dealer">Dealer</option>

        <option value="Individual">Individual</option>

      </select>

    <br>

    <br>

      <h3>Transmission Type</h3><select name="Transmission_Mannual" id="research" required="required">

        <option value="Mannual">Manual Car</option>

        <option value="Automatic">Automatic Car</option>

      </select>

      <br><br><button id="sub" class="btn btn-primary" type="submit ">Selling Price</button>

    </form>

    <br>

                          </div>

    </div>

                          <h3>{{ prediction_text }}<h3>

  </div>

  <!-- Contact End -->

 </div>

  <!-- Footer Start -->

  <div class="container-fluid position-relative overlay-top bg-dark text-white-50 py-5" style="margin-top: 90px;">

    <div class="container mt-5 pt-5">
```

```
            <div class="row">

        </div>

    </div>

    <div class="container-fluid bg-dark text-white-50 border-top py-4" style="border-color: rgba(256, 256, 256, .1) !important;">

        <div class="container">

        </div>

    </div>

    <!-- Footer End -->

    <!-- Back to Top -->

    <a href="#" class="btn btn-lg btn-primary rounded-0 btn-lg-square back-to-top"><i class="fa fa-angle-double-up"></i></a>

    <!-- JavaScript Libraries -->

    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>

    <script src="../static/lib/easing/easing.min.js"></script>

    <script src="../static/lib/waypoints/waypoints.min.js"></script>

    <script src="../static/lib/counterup/counterup.min.js"></script>

    <script src="../static/lib/owlcarousel/owl.carousel.min.js"></script>

    <!-- Template Javascript -->

    <script src="js/main.js"></script>

</body>

</html>
```

LOGIN.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <title>Car price prediction</title>

    <meta content="width=device-width, initial-scale=1.0" name="viewport">

    <meta content="Free HTML Templates" name="keywords">

    <meta content="Free HTML Templates" name="description">

    <!-- Favicon -->

    <link href="../img/favicon.ico" rel="icon">

    <!-- Google Web Fonts -->

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500;600;700&family=Open+Sans:wght@4
00;600&display=swap" rel="stylesheet">

    <!-- Font Awesome -->

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">

    <!-- Libraries Stylesheet -->

    <link href="../static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

    <!-- Customized Bootstrap Stylesheet -->

    <link href="../static/css/car.css" rel="stylesheet">

</head>
```

```
<body>

   <!-- Topbar Start -->

   <!-- Topbar End -->

   <!-- Navbar Start -->

   <div class="container-fluid p-0">

      <nav class="navbar navbar-expand-lg bg-white navbar-light py-3 py-lg-0 px-lg-5">

         <a href="index.html" class="navbar-brand ml-lg-3">

            <h1 class="m-0 text-uppercase text-primary"> Car Prices</h1>

         </a>

         <button type="button" class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">

            <span class="navbar-toggler-icon"></span>

         </button>

         <div class="collapse navbar-collapse justify-content-between px-lg-3" id="navbarCollapse">

            <div class="navbar-nav mx-auto py-0">

            <a href="{{ url_for('first')}}" class="nav-item nav-link active">Home</a>

               <a href="{{ url_for('login')}}" class="nav-item nav-link">Login</a>

            </div>

         </div>

      </nav>

   </div>

   <!-- Navbar End -->
```

```
    <!-- Header Start -->

    <div class="jumbotron jumbotron-fluid page-header position-relative overlay-bottom"
style="margin-bottom: 90px;">

        <div class="container text-center py-5">

          <h1 class="text-white mt-4 mb-4">Car Prices Detection  </h1>

            <div class="d-inline-flex text-white mb-5">

                <h1 class="text-white mt-4 mb-4">Login  </h1>

            </div>

        </div>

    </div>

    <!-- Header End -->

    <!-- Contact Start -->

    <div class="container-fluid py-5">

        <div class="container py-5">

 <head>

   <center>  <h1 class="text-black mt-4 mb-4">Login</h1></center>

  <style>

body  {

background-image: url("../static/images/email3.jpg");

}
```

```
</style>
                         <script>
  addEventListener("load", function () {
    setTimeout(hideURLbar, 0);
  }, false);
  function hideURLbar() {
    window.scrollTo(0, 1);
  }
  function login(){
  var uname = document.getElementById("uname").value;
  var pwd = document.getElementById("pwd").value;

                        if(uname == "admin" && pwd == "admin")
                        {
                         alert("Login Success!");

                         window.location = "{{url_for('upload')}}";
                         return false;
                        }
                        else
                        {
                        alert("Invalid Credentials!")
                        }
}
   </script>
</head>
 <body id="page-top">
  <!-- Portfolio Section -->
  <section class="page-section portfolio" id="portfolio">
      <br>
  <br>
      <!-- Portfolio Section Heading -->
      <!-- Icon Divider -->
      <!-- Portfolio Grid Items -->
      <div class="row">
        <!-- Portfolio Item 1 -->
        <div class="col-md-6 col-lg-4" style="margin-left:380px">
                                             <div class="control-group">
                                                        <!-- Username -->
                                                        <label class="control-label"
for="username"><b>Username</b></label>
                                                        <div class="controls">
                                                          <input type="text" id="uname" name="uname"
placeholder="" class="form-control”
                                                                </div>
                                                       </div>
                                                       <div class="control-group">
```

```
                                    <!-- Password-->

                                         <br>

                                    <label class="control-label"
for="password"><b>Password</b></label>

                                    <div class="controls">

                                     <input type="password" id="pwd" name="pwd"
placeholder="" class="form-control">

                                         </div>

                                   </div>

                                        <div class="col-md-6 col-lg-4" style="margin-left:-150px">

                                   <div class="control-group">

                                    <!-- Button -->

                                         <br>

                                    <div class="controls">

                                     <input type="button" class="btn btn-primary" value="Login"
style="margin-left: 280px" onclick="login()">

                                         </div>

                                   </div>

                                        </div>

 </div>

 </div>

 </section>
```

```
   </body>

      </div>

   </div>

   <!-- Contact End -->

   <!-- Footer Start -->

   <div class="container-fluid position-relative overlay-top bg-dark text-white-50 py-5" style="margin-
top: 90px;">

      <div class="container mt-5 pt-5">

         <div class="row">

      </div>

   </div>

   <div class="container-fluid bg-dark text-white-50 border-top py-4" style="border-color: rgba(256,
256, 256, .1) !important;">

      <div class="container">

      </div>

   </div>

   <!-- Footer End -->

   <!-- Back to Top -->

   <a href="#" class="btn btn-lg btn-primary rounded-0 btn-lg-square back-to-top"><i class="fa fa-
angle-double-up"></i></a>

  <!-- JavaScript Libraries -->

   <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
```

```
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>

    <script src="../static/lib/easing/easing.min.js"></script>

    <script src="../static/lib/waypoints/waypoints.min.js"></script>

    <script src="../static/lib/counterup/counterup.min.js"></script>

    <script src="../static/lib/owlcarousel/owl.carousel.min.js"></script>

    <!-- Template Javascript -->

    <script src="js/main.js"></script>

</body>

</html>
```

PERFORMANCE.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Car price prediction</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta content="Free HTML Templates" name="keywords">
    <meta content="Free HTML Templates" name="description">
    <!-- Favicon -->
    <link href="../img/favicon.ico" rel="icon">
    <!-- Google Web Fonts -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500;600;700&family=Open+Sans:wght@400;600&display=swap" rel="stylesheet">
    <!-- Font Awesome -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
    <!-- Libraries Stylesheet -->
    <link href="../static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
    <!-- Customized Bootstrap Stylesheet -->
    <link href="../static/css/car5.css" rel="stylesheet">
</head>
```

```
<body>

    <!-- Topbar Start -->

    <!-- Topbar End -->

    <!-- Navbar Start -->

    <div class="container-fluid p-0">

        <nav class="navbar navbar-expand-lg bg-white navbar-light py-3 py-lg-0 px-lg-5">

            <a href="index.html" class="navbar-brand ml-lg-3">

                <h1 class="m-0 text-uppercase text-primary"> Car Prices</h1>

            </a>

            <button type="button" class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse justify-content-between px-lg-3" id="navbarCollapse">

                <div class="navbar-nav mx-auto py-0">

                   <a href="{{ url_for('first')}}" class="nav-item nav-link active">Home</a>

                    <a href="{{ url_for('login')}}" class="nav-item nav-link">Login</a>

                                                                                             <a href="{{ url_for('upload')}}" class="nav-item
nav-link">Upload</a>

                                                                                              <a href="{{ url_for('index')}}" class="nav-item nav-
link">Prediction</a>

                                                                                                           <a href="{{ url_for('performance')}}"
class="nav-item nav-link">Performance</a>

                                                                                                                         <a href="{{ url_for('first')}}"
class="nav-item nav-link">Logout</a>
```

```
        </div>

      </div>

    </nav>

  </div>

  <!-- Navbar End -->

  <!-- Header Start -->

  <div class="jumbotron jumbotron-fluid page-header position-relative overlay-bottom"
style="margin-bottom: 90px;">

    <div class="container text-center py-5">

     <h1 class="text-white mt-4 mb-4">Car Prices Detection  </h1>

      <div class="d-inline-flex text-white mb-5">

        <h1 class="text-white mt-4 mb-4">Performance  </h1>

      </div>

    </div>

  </div>

  <!-- Header End -->

  <!-- Contact Start -->

  <div class="container-fluid py-5">

    <div class="container py-5">

     <center>  <h1 class="text-black mt-4 mb-4"></h1></center>

                                  <div class="container pt-5 pb-3">

      <div class="text-center mb-3 pb-3">

       <h1>PERFORMANCE ANALYSIS</h1>
```

```
          </div>
 <div class="row gy-4">
  <div class="section-header">
        <h2> </h2>
      </div>
          <body>
                                        <div style="margin-left:150px">
 </br></br>
<div><span style="margin-left:180px;color:black"><font size="12">The r2_score:  </font>
<span style="margin-left:50px;color:black"><font size="12">0.92</font>
</div>
</br>
</br>
<div><span style="margin-left:180px;color:black"><font size="12">Mean Absolute Error:</font>
<span style="margin-left:60px;color:black"><font size="12">0.81  </font>
</div>
</br>
</br>
 <div><span style="margin-left:180px;color:black"><font size="12">RMSE:   </font>
<span style="margin-left:130px;color:black"><font size="12"> 1.54</font>
</div>
</br>
</br>
```

```
<div><span style="margin-left:180px;color:black"><font size="12">Standard Deviation:   </font>

<span style="margin-left:20px;color:black"><font size="12">9.22 %</font>

</div>

</br>

</br>

 </br>

</br>

</br>

</br>

</br>

 </div>

<div class="col-lg-8 mx-auto">

      <!-- To configure the contact form email address, go to mail/contact_me.php and update the
email address in the PHP file on line 19. -->

            <div><span style="margin-left:100px;color:red"><font size="8">Confusion
Matrix</font></div>

                                                  </br>

                                                  </br>

                                           <center>        <div><img style="margin-left:-
50px;width:800px;height:500px;" src="../static/download.png"</div></center>


     </div>

</body>

      </div>

  </div>
  </div>
    </div>
    <!-- Contact End -->
    <!-- Footer Start -->
    <div class="container-fluid position-relative overlay-top bg-dark text-white-50 py-5" style="margin-top: 90px;">
        <div class="container mt-5 pt-5">
            <div class="row">
        </div>
    </div>
    <div class="container-fluid bg-dark text-white-50 border-top py-4" style="border-color: rgba(256, 256, 256, .1) !important;">
        <div class="container">
        </div>
    </div>
    <!-- Footer End -->
    <!-- Back to Top -->
    <a href="#" class="btn btn-lg btn-primary rounded-0 btn-lg-square back-to-top"><i class="fa fa-angle-double-up"></i></a>
    <!-- JavaScript Libraries -->
    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.bundle.min.js"></script>
    <script src="../static/lib/easing/easing.min.js"></script>
    <script src="../static/lib/waypoints/waypoints.min.js"></script>
    <script src="../static/lib/counterup/counterup.min.js"></script>

    <script src="../static/lib/owlcarousel/owl.carousel.min.js"></script>

    <!-- Template Javascript -->

    <script src="js/main.js"></script>

</body>

</html>
```

CHAPTER 7

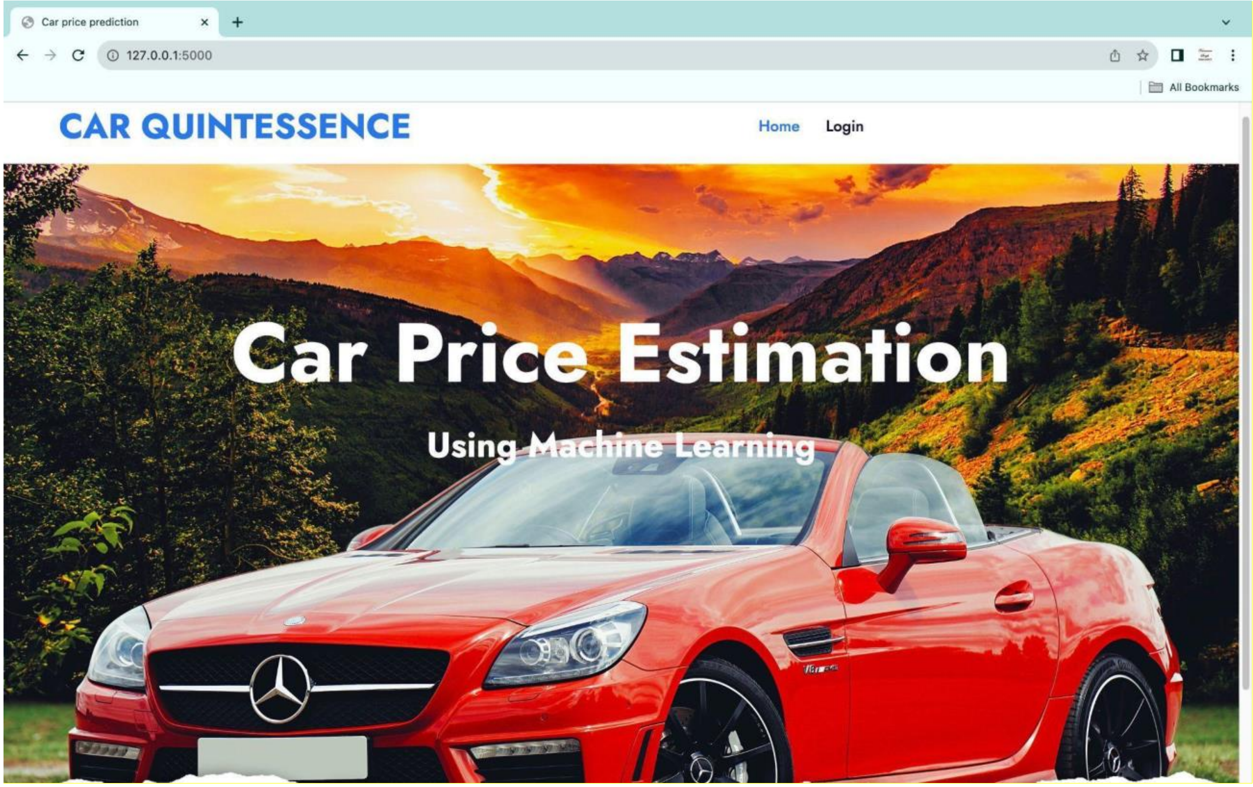
SNAPSHOTS

7.1 GENERAL

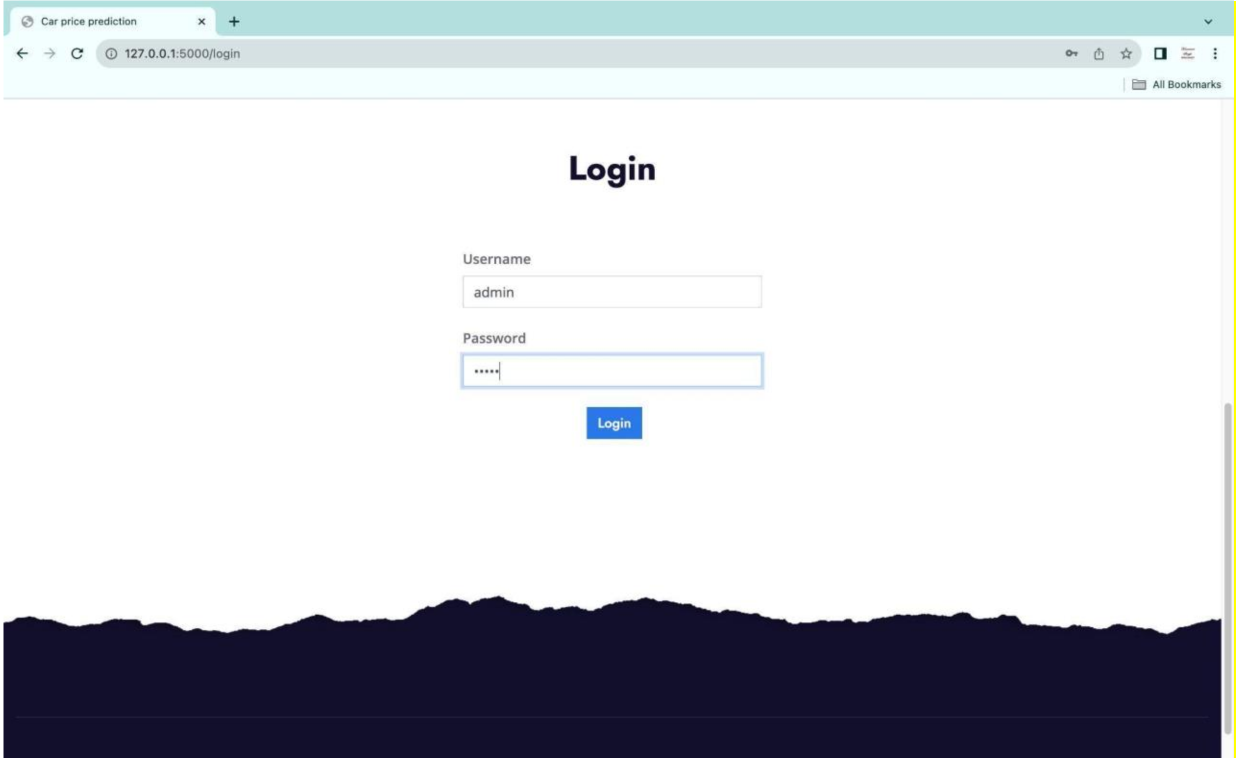
This project is implemented like an application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

7.2VARIOUS SNAPSHOTS

1. HOME PAGE



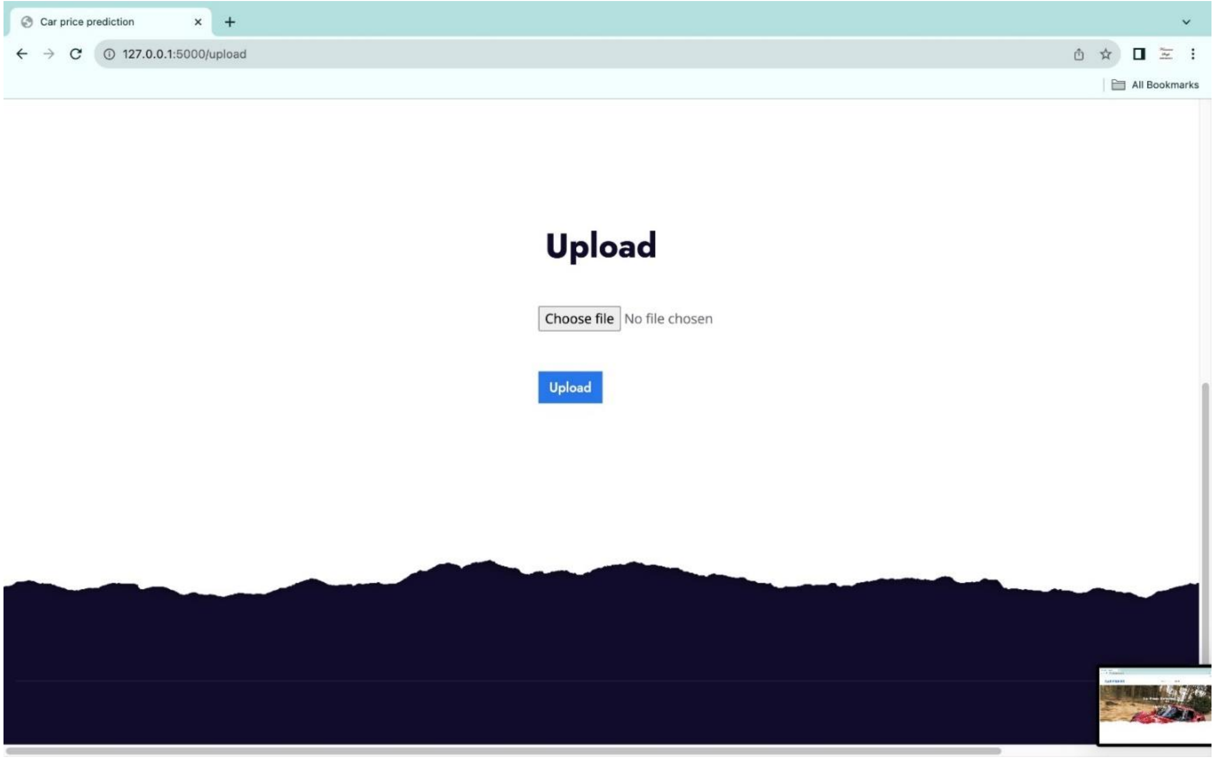
2.LOGIN PAGE



Username: admin

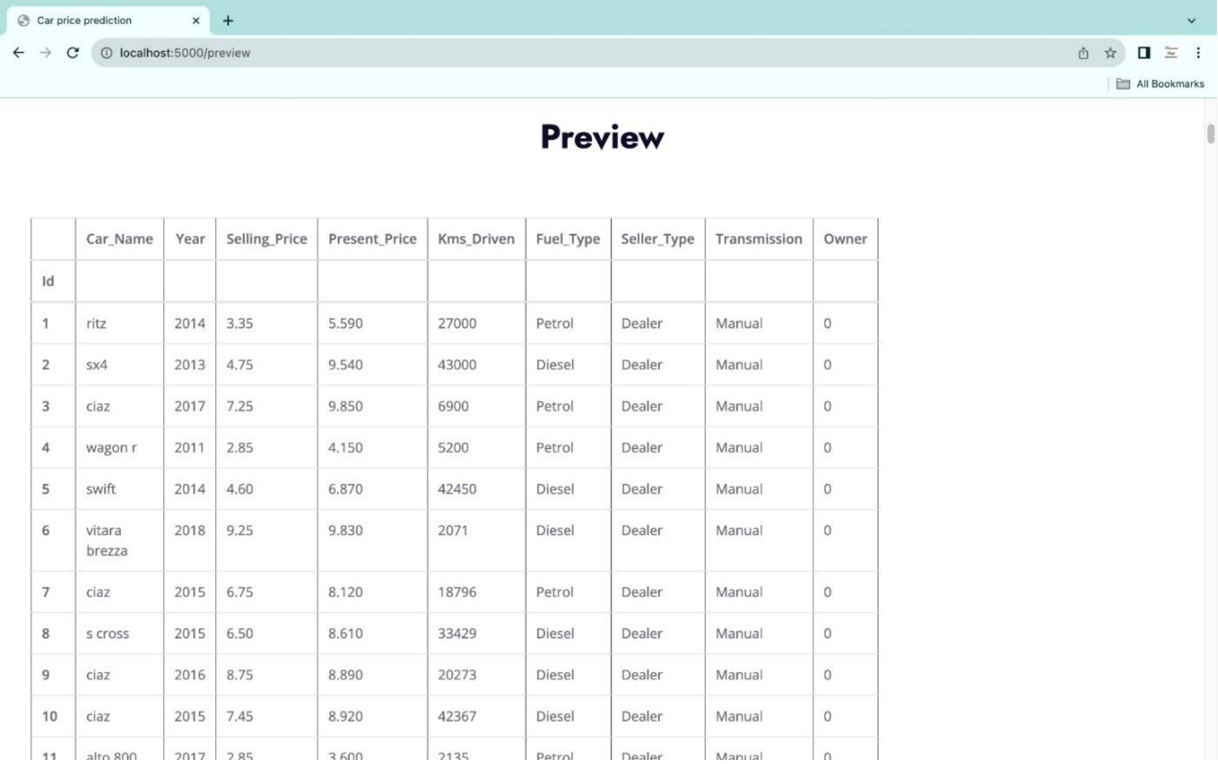
Password: •••••

3. UPLOADING DATASET .CSV FILE



No file chosen

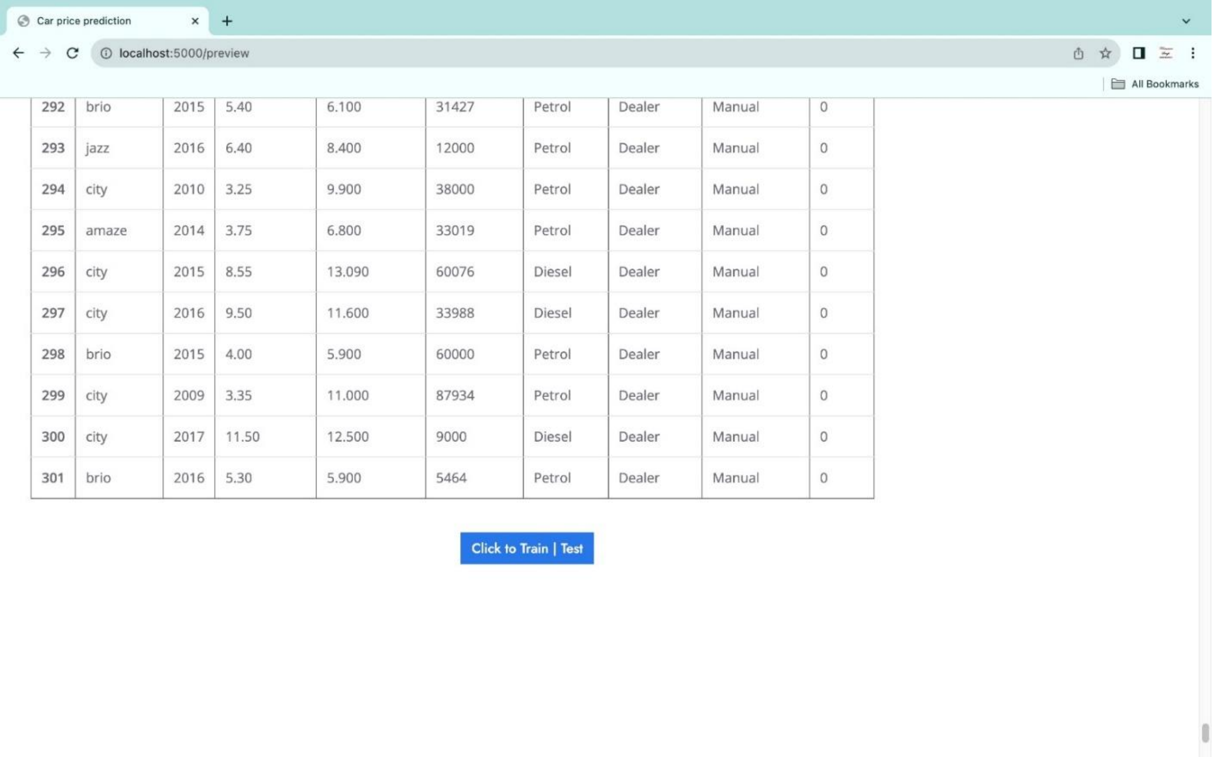
4. PREVIEW DATA FRAME



Preview

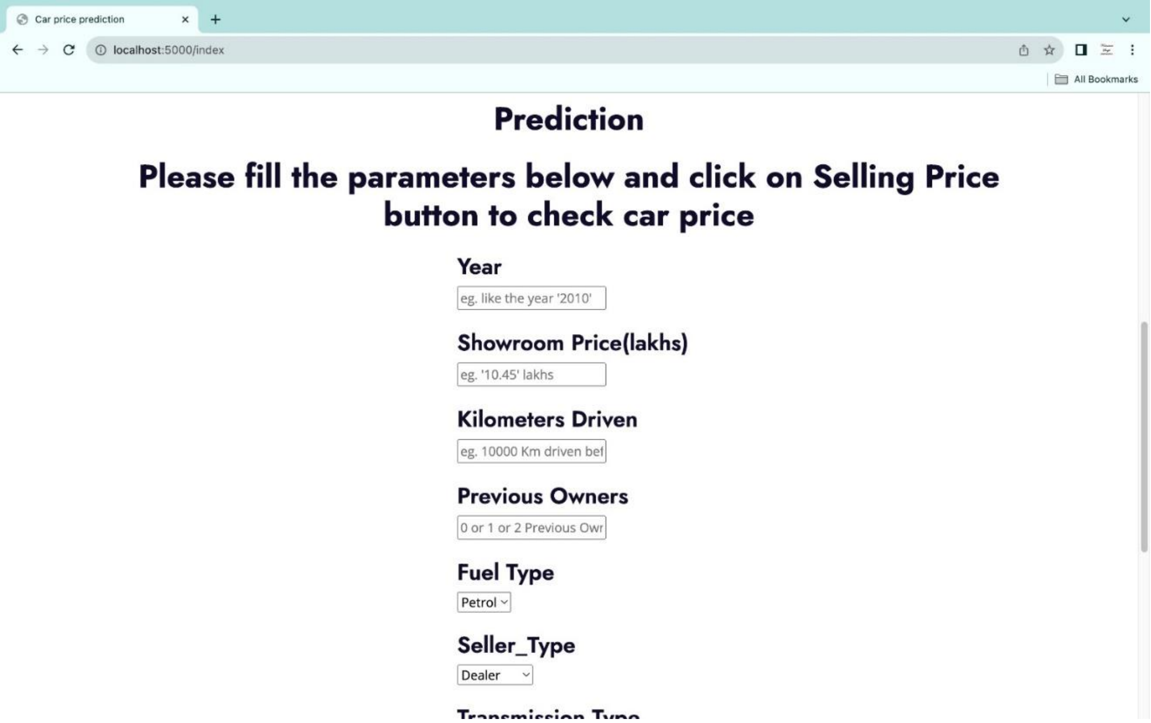
	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
Id									
1	ritz	2014	3.35	5.590	27000	Petrol	Dealer	Manual	0
2	sx4	2013	4.75	9.540	43000	Diesel	Dealer	Manual	0
3	ciaz	2017	7.25	9.850	6900	Petrol	Dealer	Manual	0
4	wagon r	2011	2.85	4.150	5200	Petrol	Dealer	Manual	0
5	swift	2014	4.60	6.870	42450	Diesel	Dealer	Manual	0
6	vitara brezza	2018	9.25	9.830	2071	Diesel	Dealer	Manual	0
7	ciaz	2015	6.75	8.120	18796	Petrol	Dealer	Manual	0
8	s cross	2015	6.50	8.610	33429	Diesel	Dealer	Manual	0
9	ciaz	2016	8.75	8.890	20273	Diesel	Dealer	Manual	0
10	ciaz	2015	7.45	8.920	42367	Diesel	Dealer	Manual	0
11	alto 800	2017	2.85	3.600	2135	Petrol	Dealer	Manual	0

5. TRAINING THE REGRESSION MODEL



292	brio	2015	5.40	6.100	31427	Petrol	Dealer	Manual	0
293	jazz	2016	6.40	8.400	12000	Petrol	Dealer	Manual	0
294	city	2010	3.25	9.900	38000	Petrol	Dealer	Manual	0
295	amaze	2014	3.75	6.800	33019	Petrol	Dealer	Manual	0
296	city	2015	8.55	13.090	60076	Diesel	Dealer	Manual	0
297	city	2016	9.50	11.600	33988	Diesel	Dealer	Manual	0
298	brio	2015	4.00	5.900	60000	Petrol	Dealer	Manual	0
299	city	2009	3.35	11.000	87934	Petrol	Dealer	Manual	0
300	city	2017	11.50	12.500	9000	Diesel	Dealer	Manual	0
301	brio	2016	5.30	5.900	5464	Petrol	Dealer	Manual	0

6. ENTERING CAR PARAMETERS



Prediction

Please fill the parameters below and click on Selling Price button to check car price

Year

eg. like the year '2010'

Showroom Price(lakhs)

eg. '10.45' lakhs

Kilometers Driven

eg. 10000 Km driven bef

Previous Owners

0 or 1 or 2 Previous Owr

Fuel Type

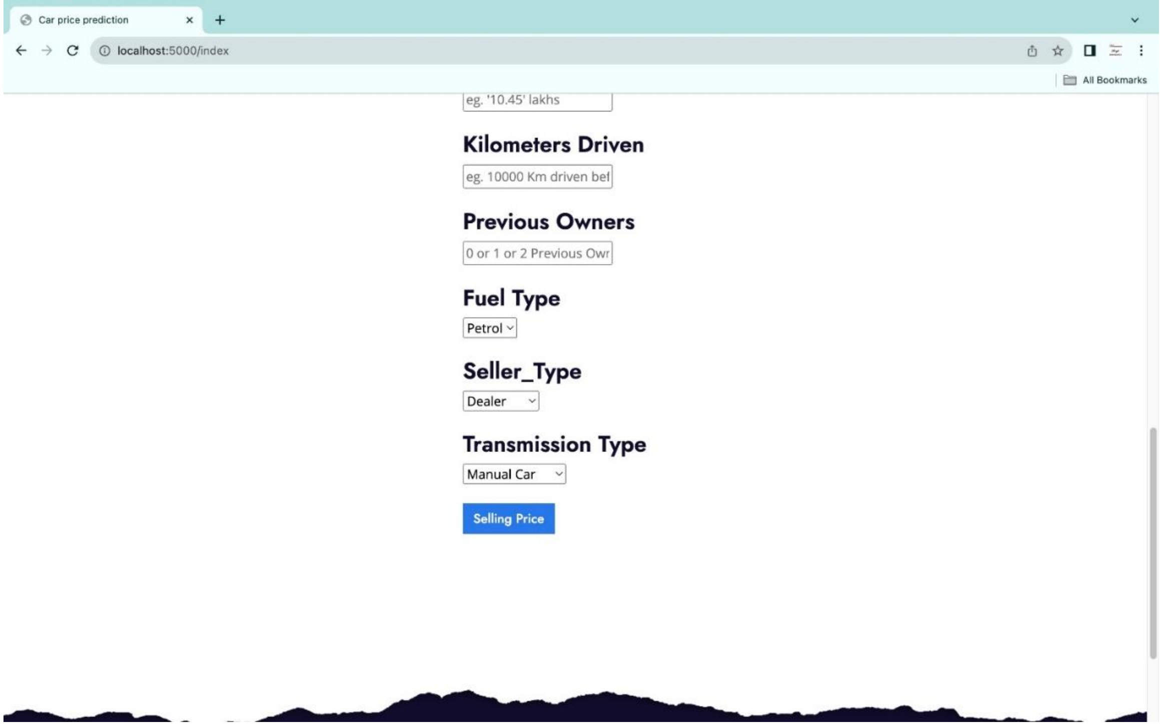
Petrol

Seller_Type

Dealer

Transmission Type

7. ENTERING CAR FEATURES



eg. '10.45' lakhs

Kilometers Driven

eg. 10000 Km driven bef

Previous Owners

0 or 1 or 2 Previous Owr

Fuel Type

Petrol

Seller_Type

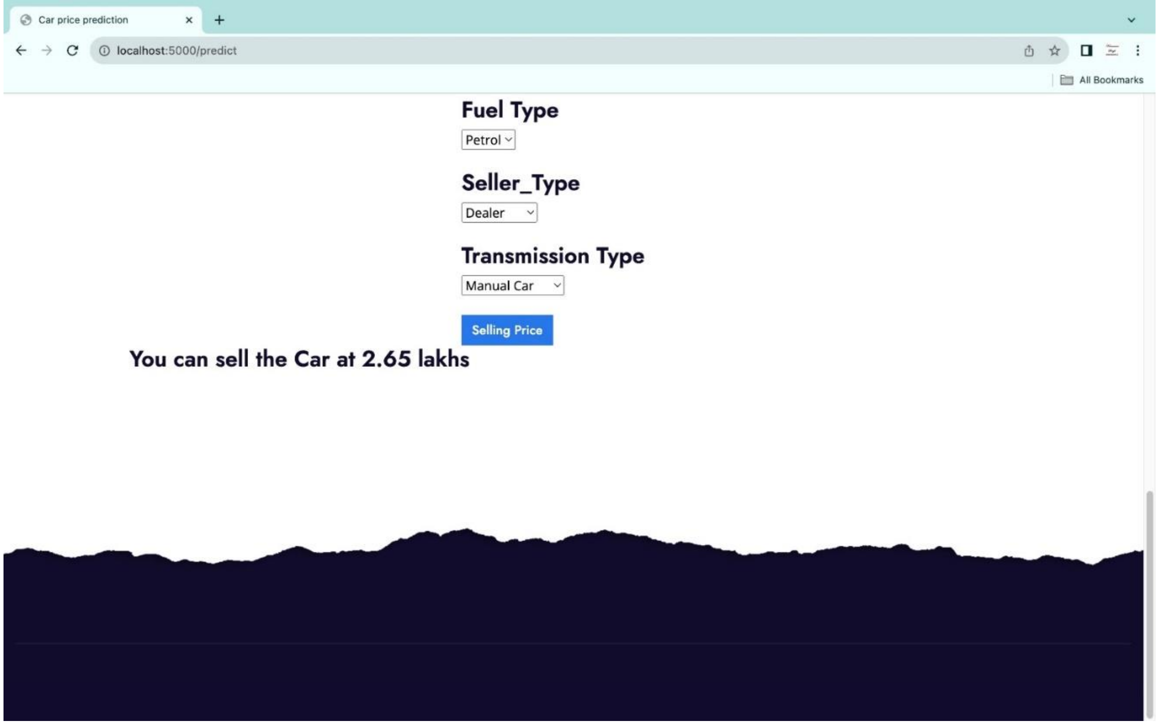
Dealer

Transmission Type

Manual Car

Selling Price

8. PREDICTED CAR PRICE (OUTPUT)



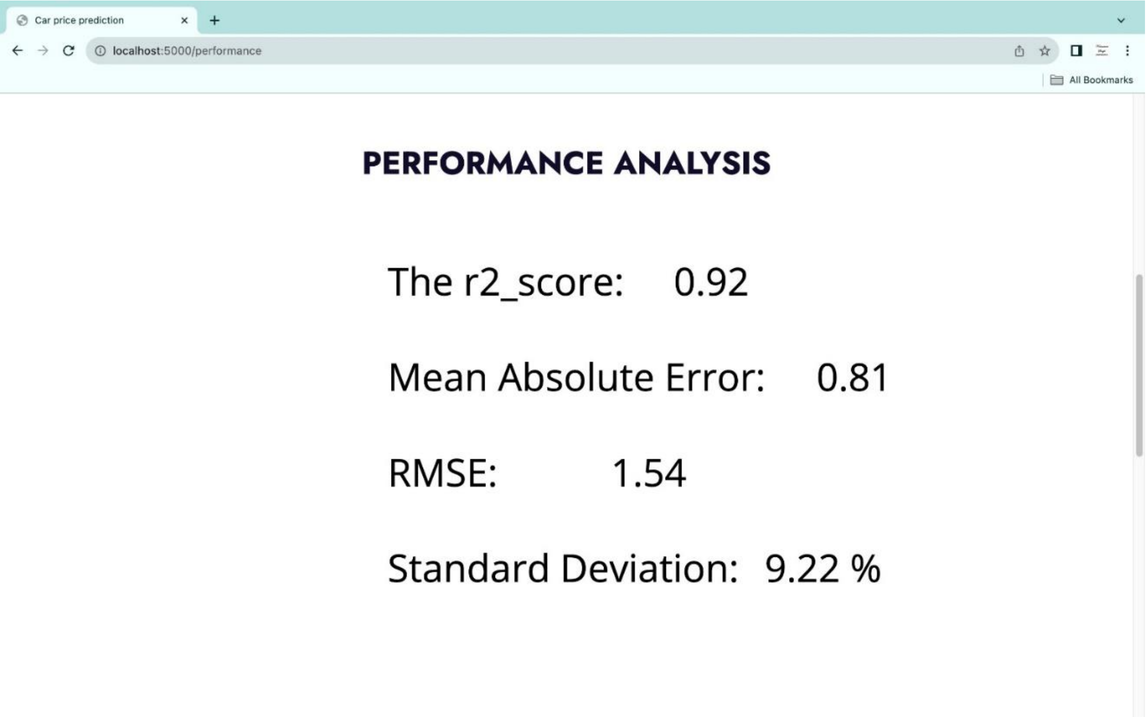
Fuel Type
Petrol

Seller_Type
Dealer

Transmission Type
Manual Car

You can sell the Car at 2.65 lakhs

9. PERFORMANCE OF THE RANDOM FOREST MODEL



PERFORMANCE ANALYSIS

The r2_score: 0.92

Mean Absolute Error: 0.81

RMSE: 1.54

Standard Deviation: 9.22 %

CHAPTER 8

SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 TYPES OF TESTS

8.3.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

8.3.3 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 PERFORMANCE TEST

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

ACCEPTANCE TESTING FOR DATA SYNCHRONIZATION:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 BUILD THE TEST PLAN:-Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9

CONCLUSION

9.1 CONCLUSION

Car price prediction poses a formidable challenge due to the multitude of attributes crucial for accurate estimations. A pivotal phase in this process involves data collection and preprocessing. In this research, data normalization, standardization, and cleaning were executed through a dedicated script, mitigating unnecessary noise for machine learning algorithms. Although data cleaning significantly enhances prediction performance, it proved insufficient for the complexities inherent in the dataset. Employing other machine learning algorithms resulted in an accuracy below 50%. Consequently, this project proposes the Random Forest Algorithm Regression approach, achieving a notable accuracy of ~92%. This marked improvement underscores the effectiveness of this approach. However, it is acknowledged that this system demands more computational resources than other machine learning algorithms.

CHAPTER 10

FUTURE ENHANCEMENTS

10.1 FUTURE ENHANCEMNETS AND APPLICATIONS

Although, this system has achieved astonishing performance in car price prediction problem our aim for the future research is the following:-

DYNAMIC PRICE PREDICTION MODELS:-Implement dynamic pricing models that consider real-time market trends, economic factors, and emerging technologies, providing users with even more accurate and up-to-date car price predictions.

USER CUSTOMIZATION AND PREFERENCES:-

Introduce user profiles where individuals can set preferences, allowing the system to tailor predictions based on specific criteria like brand loyalty, environmental consciousness, or desired features.

INCORPORATION OF EXTERNAL DATASETS:-

Enhance the dataset by integrating external sources such as automotive industry reports, consumer sentiment data, and environmental impact assessments to further refine predictions and provide a holistic view of the market.

INTERACTIVE DATA VISUALIZATIONS:-

Develop interactive visualizations that allow users to explore and interpret the factors influencing car prices, promoting a deeper understanding of the prediction model and fostering user engagement.

GEO-SPECIFIC PRICE INSIGHTS:-

Provide geographically specific insights into regional pricing variations, considering factors like local demand, supply chain dynamics, and economic conditions to offer users location-specific predictions.

Integration of Advanced Machine Learning Algorithms:--

Explore and integrate advanced machine learning algorithms beyond Random Forest, such as neural networks or gradient boosting, to evaluate their impact on prediction accuracy and enhance the system's capabilities.

MOBILE APPLICATION INTEGRATION-

Develop a mobile application to extend the accessibility of the car price prediction system, enabling users to input parameters and receive predictions conveniently on their smartphones.

HISTORICAL PRICE TRENDS ANALYSIS:-

Implement a feature that analyzes historical car price trends, allowing users to track how prices have evolved over time and providing valuable insights for predicting future trends.

REFERENCES

[1] A. Das Mou, P. K. Saha, S. A. Nisher and A. Saha, "A Comprehensive Study of Machine Learning algorithms for Predicting Car Purchase Based on Customers Demands," 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 2021,IEEE, pp. 180-184, doi: 10.1109/ICICT4SD50815.2021.9396868.

[2] J. C. Pope and J. Silva-Risso, “The psychological effect of weather on car purchases* meghan r. busse devin g. pope,” The Quarterly Journal of Economics, vol. 1, no. 44, p. 44, 2014.

[3] M. Jayakameswaraiah and S. Ramakrishna, “Development of data mining system to analyze cars using tknn clustering algorithm,” In- ternational Journal of Advanced Research in Computer Engineering Technology, vol. 3, no. 7, 2014.

[4] F.Harahap,A.Y.N.Harahap,E.Ekadiansyah,R.N.Sari,R.Adawiyah, and C. B. Harahap, “Implementation of na ̈ive bayes classification method for predicting purchase,” in 2018 6th International Conference on Cyber and IT Service Management (CITSM). IEEE, 2018, pp. 1–5.

[5] K. S. Durgesh and B. Lekha, “Data classification using support vector machine,” Journal of theoretical and applied information technology, vol. 12, no. 1, pp. 1–7, 2010.

[6] R. Ragupathy and L. Phaneendra Maguluri, “Comparative analysis of machine learning algorithms on social media test,” International Journal of Engineering and Technology(UAE), vol. 7, pp. 284–290, 03 2018

[7] N.Pal,P.Arora,P.Kohli,D.Sundararaman,andS.S.Palakurthy, “How much is my car worth? a methodology for predicting used cars prices using random forest,” in Future of Information and Communication Conference. Springer, 2018, pp. 413–422.

[8] S. Pudaruth, “Predicting the price of used cars using machine learning techniques,” Int. J. Inf. Comput. Technol, vol. 4, no. 7, pp. 753–764, 2014.

[9] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, “Supervised machine learning algorithms: classification and comparison,” International Journal of Computer Trends and Technology (IJCTT), vol. 48, no. 3, pp. 128–138, 2017.

[10] M. R. Busse, D. G. Pope, J. C. Pope, and J. Silva-Risso, “The psychological effect of weather on car purchases,” The Quarterly Journal of Economics, vol. 130, no. 1, pp. 371–414, 2015.

[11] S. Veni and A. Srinivasan, “Defect classification using na ̈ive bayes classification,” Interbational Journal of Applied Engineering Research, vol. 12, no. 22, pp. 12 693–12 700, 2017.

[12] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, “Car price prediction using machine learning techniques,” 2019.

[13] M. Jabbar, “Prediction of heart disease using k-nearest neighbor and particle swarm optimization,” Biomed. Res, vol. 28, no. 9, pp. 4154– 4158, 2017.