A Project Report

On

Car Price Prediction

Submitted in partial fulfillment of the

requirement for the award of the degree of

MASTER's IN SCIENCE

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

DEGREE

Session 2024-25
in

M. Sc. Computer Science

By
Amisha Rana
23SCSE2120002

Under the guidance of
Dr. Sudeept Yadav

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

May 2025

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" Car Price Prediction"** in partial fulfillment of the requirements for the award of the MSC (Masters In Science) submitted in the School of Computer Applications and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2024 to June and 2025, under the supervision of **Dr. Sudeept Yadav**, Department of Computer Science and Engineering/School of Computer Applications and Technology , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other place.

Amisha Rana (23SCSE2120002)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Sudeept Yadav

Designation

CERTIFICATE

This is to certify that Project Report entitled "**Car Price Prediction**" which is submitted by **Amisha Rana** in partial fulfillment of the requirement for the award of degree MSC. in Department of **Computer Science** of School of Computer Applications and Technology, Galgotias University, Greater Noida, India is a record of the candidate own work carried out by him/them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree

Signature of Examiner(s) **Signature of Supervisor(s)**

Date: 27 May 2025

Place: Greater Noida

ABSTRACT

Working out the price of a used car can be difficult due to factors such as brand, model, year, how many kilometres it has and its overall condition. Reliable pricing is sometimes difficult with methods that depend on someone's opinion. This project applies machine learning to make sure predictions for used car prices are both more accurate and consistent.

The sequence starts when you prepare your data set by formatting, replacing missing values and encoding categorical variables. Using feature selection approaches, we uncover the key factors that impact car prices. We develop and study three machine learning models—Linear Regression, Random Forest and Gradient Boosting—using metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and the R-squared measure.

Even though Linear Regression is easy to interpret, it does not suit data that is not linear. On the other hand, algorithms such as Random Forest and Gradient Boosting perform best when facing complicated data. In Random Forest, multiple decision trees are used to stop overfitting which gives way to better results.

It is demonstrated in the results that Random Forest and Gradient Boosting produce higher accuracy and lower error than Linear Regression. You can find these models useful in many other areas besides cars such as real estate and retail.

TABLE OF CONTENT

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

The automotive industry has developed rapidly with higher utilization of secondhand cars mainly due to economic and ecological status. With growth in the secondhand car market, it becomes difficult to decide at what price a specific car must be sold. The main components that affect the value of the car are brand, model, year of manufacture, kilometers eligible, type of fuel, and condition of car. Such a situation results in the fluctuation of prices on the same products which makes the buyers and the sellers quite bewildered.

Previously, the approaches to car price evaluation included the concepts based on the intuitive judgement of experts or the one-time manuals that ignore changes in the market features. They can cause mismatches between prices on cars which can lead to changes in sales and customer satisfaction. Therefore, an automated, data-based solution for car price estimates has become evident.

The current solution of machine learning supports this method by involving a huge dataset of used car attributes and previous prices. The main goal of this project is the development of a strong car price prediction system through machine learning and establishing accurate and reliable price prediction. Some algorithms such as Random Forest & Gradient Boosting will be used to predict prices from the various features characteristic to cars. The model is expected to minimize the error and subjectivity that decision-makers are likely to bring in the evaluation of the cars, hence being transparent and scalable.

From this project, different stakeholders in the automotive sector, such as dealers, purchasers of used cars or even financial institutions, will get a dependable approach to estimating the costs of a car. Furthermore, it can be generalized for other markets where there are other factors affecting the pricing of a particular product.

1.1. Problem Introduction

It used to be a complex task to determine an accurate estimate of the price of cars, especially for used cars as they are chop full of attributes. Unfortunately, previous practices involving manual appraisals or fixed price guides are not sufficient since they cannot capture the current conditions of the market, or the state of the vehicles involved. Such approaches entail human interference, making them vulnerable to bias, mistakes and inaccuracies pushing up vehicle prices higher than they should or down to lower prices than they should be.

Primary characteristics like make, model, year, age, and model year, mileage, fuel type, necessary capacity, and a general technological state are the key components to decide on the reserve value of the used car. However, the

interactions of these attributes are very complex and frequently the effects of these attributes can be non-linear. For instance, two similar cars of same make and model might be worth very different amounts because one might have run so many kilometers more than the other or used more than the other car. Due to these complexities, using manual means to forecast results is practically virtually impossible and utterly inaccurate.

What the evolution of machine learning provides as an answer to this question is rooted in the data. Given massive amounts of historical data and powerful algorithms of machine learning, the model can find some relation between car features and make the price prediction more accurate. Not only do these models cut down on the possibility of human mistakes but they also significantly cut down the time, cost, and variability of valuations.

In this project, we will try to solve the problems related to car price prediction by using a machine learning model with input features to predict the output features, which is the used car prices. This approach will enhance pricing precision, which will be positive to car dealerships, buyers, and used car markets will be positive.

1.1.1. Motivation

Recently, the used car market has expanded tremendously due to certain economic factors, environmental factors and consumer preferences. It is, therefore, important to accurately predict car prices for both those who sell and those who buy. However, pricing is not an easy task, and many factors like brand, model, year, kilometer, and condition of the car often affect the result of the manual estimation. More conventional methods of dealer pricing and other standard pricing methodologies are inadequate for capturing the variability or the nature of the exercise. This poses a major agony to the potential buyers who may pay more for a vehicle than required, or the seller who maybe underpricing their vehicle thus pose to lose a lot. This need has led to the motivation of this project to remove such limitations and enhance the level of transparency vital in pricing processes.

Given the recent development in machine learning algorithms for big data analysis, it is becoming easier to sample car features and work from a huge data set to decipher patterns and arrive at market determined prices. Pricing estimates made by current machine learning models are consistent and are more reliable compared to manual pricing estimates because they are based on historical data. It’s not limited to consumers and producers only, but it also indirectly benefits dealerships, insurance companies and financial institutions since accurate price prediction gives them the key to managing business effectively. In conclusion, the objective of this process is the formation of a sustainable and tangible model that embraces the least amount of human interpolation and maximizes the precision of price

estimation for used cars, thus leading to an efficient evolution of used car market.

1.1.2. Project Objective

This of course refers to a task which is known in technical parlance as predicting the resale value which involves the use of a series of parameters that include the make and models of cars, their ages and kilometers driven, and their engine sizes. To determine the best model, this project uses different methods of applying new advanced machine learning algorithms and comparing each one of them. For the purposes of the project, attention will be paid to the algorithms such as Linear Regression, Random Forest, Gradient Boosting as these algorithms are considered to allow for accurate predictions of regression issues.

Apart from development and implementation models, one of the goals of the project is to assess whether the model can be trained on big and intricate data, whether it can be scaled up if necessary, and whether it provides good results when the data from the new sources is introduced. The performance results will be presented with the help of parameters like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), as well as R-squared coefficients. Furthermore, the project is aimed at identifying some key factors that define cars' prices and giving stakeholders an idea about which changes are crucial for the evaluation. The aim is to come up with an effective and efficient model that will fit into real life settings to help the dealership, the buyers and sellers in making efficient decisions on the best price to put on the cars.

1.1.3. Scope of the Project

The work performed in the framework of this project is centered on the creation and assessment of the performance of the machine learning algorithms for the prediction of used car prices in relation to its characteristics like make, model, the year, mileage distance, engine type, and others. The project is structured to include several phases: It covers data cleaning, feature extraction or construction, choosing of the model, model training as well as testing or validation. It will begin with an existing data set for used car prices, followed by pre-processing the data set, where missing values and outlier would be handled. After this, different machine learning models will be built and their performances evaluated including Linear Regression, Random Forest and Gradient boosting.

It also plans to scale the proposed models on any new data set and compare the results with the cross-validation results to check the scalability and generality of the models. Although the current data set is employed on used car prices, methods that are worked out in this project can be extended to other fields such as real estate, retail, technology products and so on.

Possible disadvantages of the project include the model's reliance on the size and quality of the data and designed initially for one market. However, the above methodology can be used in similar problems at a much larger scale with more data and features involved.

1.2. Related Previous Work

Predicting the prices has been approached in different sectors and the most common one include the following; real estate and stock market. In the automotive application, car price prediction, more specifically resale prices have received much attention apparently owing to the numerous factors that go into determining the value of a car. More classical ML techniques used in the past assumed linear relationship between inputs such as mileage and car age as seen in Linear Regression which besides being easy to understand failed to capture non-linearity. Recently, attention has shifted more towards higher level of implementation such as DTs, RFs and Gradient Boosting Machines (GBMs) because of their capability to make more complex decisions.

The current work demonstrated that by combining various decision trees, utilization of ensemble methods such as Random Forests and GBMs yields better results than using any single model. Such models give higher accuracy and better generality for new data, which is necessary for determining car prices, as the nature of the features is not linear. For instance, two cars which are of the same size and year could have a different price tag set for them mainly due to differences in the sizes of their mileage or of their engines. This work has also adapted how prior work has used cross-validation, as well as feature selection and data preprocessing techniques like handling missing values and normalizing numerical features, among others. To embark on this project, the above advancements are established as the foundation on which this work is based on aiming at developing an improved price prediction for the used car market. With the help of this project, the authors aim to enhance the results of the prior initiatives in terms of algorithm efficiency and model parameters.

It briefly includes previous work carried out in this field, researching the problems studied, summarization of the results obtained etc.

1.3. Organization of the Report

Chapter 1: Background of the study offers an understanding of the problem under study, reasons why the problem should be solved, purpose of the study, and the study's proposed area of focus. It states the current importance of building an accurate car price prediction model, objectives of the project and brief literature review of related works.

Chapter 2: Car Price Prediction and Machine Learning/ Software Reviews Literature survey relevant research in car price prediction. It also describes from before used approaches like Linear Regression, Decision Trees, and Random Forests. Furthermore, this chapter describes the software instruments to be used during the project implementing, namely Python, Scikit-learn, and Jupyter Notebook, as well as explains the peculiarities of the dataset utilized throughout the model constructing process.

Chapter 3: System Design and Methodology gives an overview of how the car price prediction model was designed, how data was preprocessed, which features were selected and how the model was implemented. It also details the algorithms used here including Random Forest and Gradient Boosting as well as comparing them. It also has system design diagrams, data flowcharts, and algorithm descriptions to fully explain the concept of the methodology.

Chapter 4: It explains the processes of implementing the developed machine learning models through code samples, screen shots of the interfaces and outcomes achieved for the Idaho State plan. This chapter also reports performance assessment through MAE and RMSE through which the effectiveness of various algorithms in the estimation of car prices is determined. Contents are summarized in graphical and tabular forms as a way of presenting findings.

Chapter 5: Evaluation and conclusion provide an overview of the outcome of the project, with an explanation regarding how the models perform in comparison to other existing technologies. This chapter also outlines the directions for future work on the improvement of the model, including the inclusion of more data, the refinement of the choice of features, or the experiment with the new algorithms. Finally, further practical implications of using the proposed approach are also illustrated following real applications of the project in the automotive industry.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

The following are some of the important factors that play a crucial role in determining the design and development of the car price prediction system. They also guarantee the final product to be friendly to users, efficient and above all satisfactory to stakeholders. Below are key considerations:

1. User Needs and Usability:

The applied system's objective is to give accurate price estimates to technical as well as non-technical customers like buyers, sellers, dealerships, and financial institutions. Convenience is the key to interface design and provides easy data input and output and quick response of the system.

2. Data Quality and Availability:

The use of the car data is critical in the prediction model, it requires accurate, relevant and updated information about the car. Lack of, or older data degrades the quality of the predictions made by the model. To train the model, basic datasets such as car models, miles travel, car manufacturing years, types of fuel, and many other characteristics of past car models should be gathered.

3. Scalability and Performance:

When the user is sending several requests at the same time, the system must not slow down. But even with a relatively small dataset and relatively small numbers of users, scalability is important to provide fast response times.

4. Market Trends and User Preferences:

As is informed, car prices are always changeable associated with market supply and demand, user preferences, and seasonal differences. The model needs to evolve for it to stay relevant, especially considering that the market circumstances are constantly transforming. The source data as well as the model will be updated from time to time so that the predictions given will be up to date.

5. Security and Privacy:

Security is important and user information must remain safe, and the system must be safe as well. Even though personal data that is sensitive is not collected, protection of any data kept on the site from further access by unauthorized personnel is inevitable.

6. Technology Constraints:

Considering the technologies and platforms that have been used in the development of the product, it was found out that their performance determines the product. This puts into consideration the compatibility with web browsers and servers as well as choosing the relevant machine learning libraries affects development decisions.

2.1 Product Perspective

This project here entails the development of a web-based system where car prices are predicted using machine learning algorithms. It is meant to enable users to enter diverse car attributes including the make, model, year of manufacture, mileage in kilometers, fuel type, and engine capacity into a web-based interface and get a predicted resale price in return. The use of the product will benefit car dealers, buyers, and sellers, financial institutions, and insurers, who require an accurate valuation of the car.

It is also important to note that this current version of the system is non-integrated with other systems. However, they might include later versions as part of other extensive dealership management systems or car platforms where they can operate in real-time using current market data to predict prices. After the model is trained and applied with data history, the model will be uploaded on a web-based server so each user or several users would be able to type the desired product in their chosen web browser and get the price prediction from any part of the world.

In addition to this, the system is fully web-based and can, therefore, be operated on computers, tablets, mobile phones, etc., and no additional, specialized hardware or software is required. The system communicates with a back-end server on which the machine learning model is run and where data is processed. It's scalable meaning that as the number of users increases, or the size of the data set rises the efficiency of the application is not affected.

2.1.1 System Interfaces

The system interfaces are divided into two main parts: the front-end or user interface, and the interface that links the graphical form, the interface that supports a machine learning model.

- **User Interface (UI):**

Through the web interface the different car attributes including brand, year, mileage, type of fuel and engine capacity can be entered. The interface, therefore, is easy to work with, consisting of dropdown menus or text fields in most cases. When the information is entered, a user will get a predicted car price in a matter of seconds. It shall be a conclusion in terms of the developed prediction, and the results of input data summarization shown on the screen.

- **Back-end Interface:**

That is the back-end system in which the whole of the machine learning model is located. The structure involves interacting with the historical car database and processes the inputs made by the user to arrive at the price prediction. The back-end server also receives some of the front-end requests and processes it by the trained model to allow several users to use the system. This model also communicates with data storage systems for retrieving training data and with the record of the model's predictions.

2.1.2 Interfaces

The car price prediction system makes available a web-based Graphical User Interface or commonly referred to as GUI to enhance the relationship between human and the machine learning model used in these studies. The user interface is presented in forms in which the users input car specifics which may include the make, model, year, the mileage and the type of fuel. All these inputs are then computed in this system and the predicted price is shown in real-time on the same interface.

The GUI design concerns functionality and most of them are designed with clear labels, categorical inputs like dropdown menu for car brand and fuel type and other fields validated to allow correct input. The main feature of its design is that it is a responsive interface that can be used on desktop, tablet, or smartphone screens. Adjusting fitting the various platforms, and the formats allows the users to achieve the same effectiveness irrespective of the screen size.

The system does not require rigorous command-line services to enable its use by any lay consumer or car dealer. Additional modifications are added to meet the standard of special accommodation so that disabled users will be able to manage the system. These features include accessibility to screen readers, adequate contrast levels, as well as keyboard access. This project is not planned with the American with Disability Act participant audience in mind; however, it is designed with global accessibility standards in mind.

Thus, to improve accessibility in subsequent versions of the application, the input can be extended to include voice control or a chatbot. Emphasis on straightforward arousal, being accessible, and reactive makes the system pleasant to use and increases its adaptability and usage among many clients.

2.1.3 Hardware Interfaces

The car price prediction system mainly functions as a Web application. Therefore, it does not directly interface with the various hardware components apart from common web hosting requirements. Since the system will be hosted on a server which implements the system, then it will interact with typical hardware components which include web servers, database server and individual user interfaces or devices.

1. **Server-side Hardware Interface:**

It communicates with a web server where application is installed and where the user requests are handled. This web server interacts with a database server to pull and put data stored in it concerning historical car information. The server also must be able to log data and update the machine learning model of the system. This could also be addressed on AWS or Google Cloud to guarantee generalizable hardware support. HTTP/HTTPS is used as network protocols to allow secure communication between the end user interface and back end.

2. **Client-side Devices:**

Assimilating this information is done through devices such as personal computers, tablets, and handheld mobiles. These devices can only operate the system through a link with a

modern web browser only. The application is deliberately created to work on multiple clients, and there are no special demands on the hardware of the clients.

3. **Optional Hardware Expansion:**

In the following versions, the system can be expanded to interact with outside devices/Internet of Things (IoT) sensors like mileage loggers or vehicle state indicators. These devices could provide real time data into the model to increase the model's accuracy of the predictions. Any such integration would use standard APIs such as REST for transporting data from one hardware or software element to another and in a safe and secure manner.

Currently the system does have no precise demands on the required hardware control elements and exists solely as a web/cloud-based system.

2.1.4 Software Interfaces

The car price prediction system interacts with several software products and libraries that are essential for building the machine learning model, managing data, and creating the web interface. Below are the required software products and their details:

1. **Python**

   Mnemonic: Python

   Specification Number: 3.x series

   Version Number: 3.10+

   Source: Python Software Foundation (https://www.python.org/)

   Purpose:

   Python serves as the core programming language for implementing the car price prediction model. It supports data manipulation, machine learning, and web frameworks.

   Interface:

   Python interacts with multiple packages (discussed below) for data processing, machine learning, and API communication.

2. **Scikit-learn**

   Mnemonic: sklearn

   Specification Number: N/A

   Version Number: 1.2+

   Source: Scikit-learn (https://scikit-learn.org/)

   Purpose:

Scikit-learn provides the algorithms used for machine learning, including Random Forest, Linear Regression, and Gradient Boosting. It supports model training, evaluation, and optimization.

Interface:

The library is accessed through Python scripts, allowing seamless integration with datasets for model development.

3. **Flask**

Mnemonic: Flask

Specification Number: N/A

Version Number: 2.0+

Source: Pallets Projects (https://flask.palletsprojects.com/)

Purpose:

Flask is used to develop the back-end web interface for receiving user inputs and returning predictions.

Interface:

It handles HTTP/HTTPS requests between the front-end and the machine learning model. JSON formats are used for data exchange.

4. **MySQL**

Mnemonic: MySQL

Specification Number: 8.x

Version Number: 8.0+

Source: Oracle Corporation (https://www.mysql.com/)

Purpose:

MySQL is used to store historical car data. It enables data retrieval and logging of predictions.

Interface:

Communication between Flask and MySQL happens through SQL queries using Python's MySQL connector.

2.1.5 Communications Interfaces

Regarding communication among the components of the car price prediction system, standard communication protocols are used to have a proper interaction. This system is a web-based application that has the HTTP/HTTPS protocol to handle client server communication and SQL based interfaces dealing with the database.

1. **HTTP/HTTPS Protocol:**

   The system uses HTTP/HTTPS to interact between front-end and back-end. If the user enters the car details through the input web interface, then it is passed in the HTTP POST request to the server. The raw data collected here is taken through the machine learning model at the server and the model returns an HTTP response of the predicted car price to the client. This helps to enhance data security transmission hence the inputs from the users are safe from other people's access.

2. **RESTful API Communication:**

   The back end created with Flask uses RESTful principles to provide interconnectivity between the web input/output interface and the machine learning model. When the client sends JSON data to the REST API, the API analyzes them and sends the results of the predictions back in the JSON format. This protocol is light, in other words it will not bog down the speed of communication, ideal for a web context.

3. **Database Communication (SQL Protocols):**

   The following interfaces with the MySQL database to pull prior car data and to store logs of the prediction. The co-ordination with the database is provided by SQL statements coded into the Python script. These queries make sure that only appropriate data is retrieved and put in the database in the right manner hence full functioning.

4. **Optional Cloud Communication:**

   If implemented on cloud like AWS, GOOGLE CLOUD some complementary protocols like WebSocket's or API gateway may be used for real time interactions and for amplified scalability.

   These communication interfaces also guarantee secure, reliable and fast initial data exchange among users, the web server and the Db.

2.1.6 Memory Constraints

The car price prediction system is not sensitive to memory limitations because modern Web servers and client devices will be used. However, some key considerations apply:

1. ***Server-side Memory Requirements:***

   The server that hosts the machine learning model and web application should have at least 16 GB of RAM to be able to handle multiple user requests and do real-time computations across the use lifespan of the model in a lag-free manner. It may require scaling up to 32 or more GB as the model becomes complex with more

features or with large data sets. The server also requires enough memory for MySQL database query and logging operations.

2. ***Client-side Memory Requirements:***

   The usage of web browsers for system usage makes client-side memory utilization very low. It is recommended that a typical device with 4–8 GB of RAM is going to be enough to handle the application. Form submissions or display of data in the browser do not cause any utilization of large amount of memory.

3. ***Dataset Memory Impact:***

   To begin with, during the exploratory phase, data sets for training as well as for predictions are reasonably transportable within normal computational memory space. However, experiencing large-sized datasets, memory optimization can be a potential factor where processes like batch processing or data storage to the cloud can be required.

4. ***Future Scalability:***

   In case, with advanced methodologies such as deep learning or real-time data updates are involved, cloud solutions with the possibility for elastic memory character will keep high performance.

   In total there is clear that the current version of the project is fully compliant with the current memory limitations of the modern apocalyptic systems on client and server sides.

2.1.7 Operations

User interactions in the process of using the car price prediction system include interactive and background processes to enhance the usability of the car prediction system.

1. ***Interactive Operations:***

   Clients engage in the system through a website using car details to obtain price forecasts. The model works in real time because it analyses and provides the prediction almost immediately. Such operations are to be operated on a round-the-clock basis depending on the convenience of the service's users. Under normal operating conditions, no downtime is anticipated during this project.

2. ***Unattended Operations:***

   They are further classified into operations that are unattended such as model training and updates. The machine learning model may need to be updated routinely, to reflect new data as well as enhance accuracy of the representations made. These tasks can be executed in daemon mode or at nighttime when their execution may not interfere much with the users. That is, if applied on cloud platforms, automated training schedule can be set.

3. Data Backup and Recovery:

MySQL backups are made to take care of the data in case it gets lost since one is always working on the most updated data. Backups could also be done at more discreet times in order to prevent interference with the service. There is recovery procedures designed to revive the model and database in the event of its crash. It is also worth mentioning that virtually all cloud services provide disaster recovery as an additional service improving the stability of the system.

4. Maintenance and Updates:

Small maintenance work like tuning the model and updating the software could ask for a short break. However, the system will be programmed to provide the users with a warning regarding the time the system will be closed for this kind of maintenance.

Such operations allow for ensuring the systematic, accessible, reliable, and secure functioning and continuous availability of the system throughout the system life cycle.

2.1.8 Site Adaptation Requirements

The car price prediction system is principally web based, thus does not need extensive physical adjustment to the site. However, a few crucial configurations ensure smooth operation:

1. Database and Server Setup:

The hosting environment needs to be capable of working with MySQL, initial DB tables have to be created and filled with historical car data. This made certain that prediction can commence from the moment the model is deployed. The server also has to host Python with Flask for the back-end part and HTTPS for secure connection setting.

2. Model Initialization and Maintenance:

The developed machine learning model shall be required to be trained afresh from time to time, with new datasets to keep improving its accuracy. S) This can mean the use of automated scripts to support training which is programmed to occur during low traffic times.

3. Optional Cloud Infrastructure:

If the system is deployed for example on the Amazon Web Services (AWS) or Google Cloud, then site-specific setup may require setting up for example CloudWatch for the monitoring of uptimes and auto-scaling based on the traffic load of a [see 14].

This setup also ensures that there is no a lot of disruption, while other alterations can be made in the future to improve scalability.

2.2 Product Functions

This system is meant to enable predictions of car prices that are accurate and fast within the user shortlisted options. Key product functions include:

1. **User Data Input:**

   Consumers enter car information into an online form, specifying such parameters as make, model, and year.

2. **Prediction Model Execution:**

   The inputs are processed using Random Forest, Linear Regression or Gradient Boosting models and the estimated price is shown.

3. **Data Logging and Model Management:**

   The predictions are stored to the data store for analysis at a future time and the model may be adjusted by administrators based on new information.

4. **Reporting and Metrics:**

   Instead, new prediction accuracies are shown in the form of system metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

This functionality also helps keep the system as effective for dealerships and as valuable for buyers regarding pricing as it was previously.

2.3 User Characteristics

The target user base includes a range of technical and non-technical users:

1. **Car Buyers and Sellers:**

   These users need a simple means and access to the price of cars without engaging the technical part of car prices. A basic window interface makes for convenience.

2. **Dealerships and Financial Institutions:**

   These users may find detailed pricing metrics helping them in their business dealings. Therefore, dashboards with richer analytics might be useful.

3. **Administrators and Analysts:**

   This group oversees the running of the system, the provision of updated datasets and assesses the performance of models. They require average technical skills in Python and MySQL regarding their operational activities.

The design aims at being simple, intuitive, and fast so that the users may easily access it through device and platform differences.

2.4 Constraints

Several constraints impact the system's design and implementation:

1. **Hardware Limitations:**

   The server should have at least 16 GB of RAM to allow for proper stream of users and the necessary level of prediction.

2. **Regulatory Compliance:**

   If implemented in conjunction with dealerships, then it must make use of privacy compliance such as GDPR regarding the user inputs.

3. **Maintenance and Downtime:**

   Sometimes model updates are scheduled, and servers also need some maintenance, and this might take a few minutes. Such notification mechanisms in advance will alert the users.

4. **Security Requirements:**

   To guard user information from access by unauthorized people, all communication must take place using HTTPS.

These constraints make sure that the system will never be compromised, and that the system is stable and compliant with certain templates of structure organization.

2.5 Assumptions and Dependencies

Several assumptions underline the system’s development and operation:

1. **Technology Dependencies:**

   The system assumes the deployment server already has Python, Flask, MySQL, and necessary tools and libraries installed. If the cloud-deployed solution is desired, access to such platforms as AWS or Google Cloud is also required.

2. **Data Availability:**

   The model is built to use a big and integrated set of data for training and making predictions. It presumes the constant update of the car data to keep everything as truthful as it possibly can be.

3. **Network Infrastructure:**

   A stable connection to the Internet is necessary to guarantee proper communication between a client and the server. In the future if real time data is incorporated, this dependence will be even more paramount.

Any of these factors may need adjusting to match the system’s design or how it is rolled out.

2.6 Apportioning of Requirements.

Given the scope of the project, requirements are divided into phases to manage development efficiently:

1. **Phase 1 (Core Functionality):**

   To train a model, use both Random Forest and Linear Regression.

   A simple online interface using the input of the user to insert data into the map and use the output of the map to provide the user with predictions.

   Utilize MySQL database for auxiliary purposes that is for storing historical data and logs the predictions.

2. **Phase 2 (Enhanced Features):**

   Integrate real-time market data to make predictions fluid in real-time.

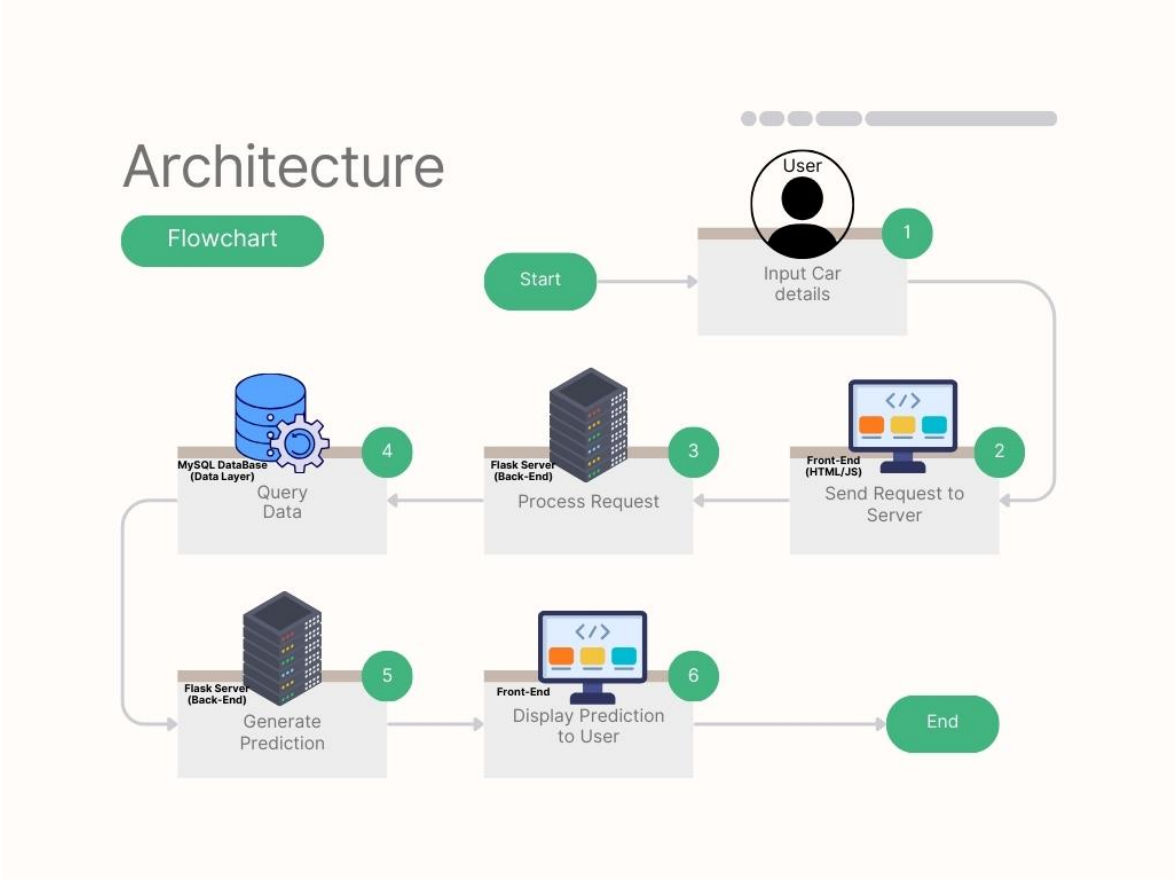
   Develop application user accounts and homepage management systems for car dealers to take awareness of trends.

   Switch over the system to the cloud for scalability and availability purpose.

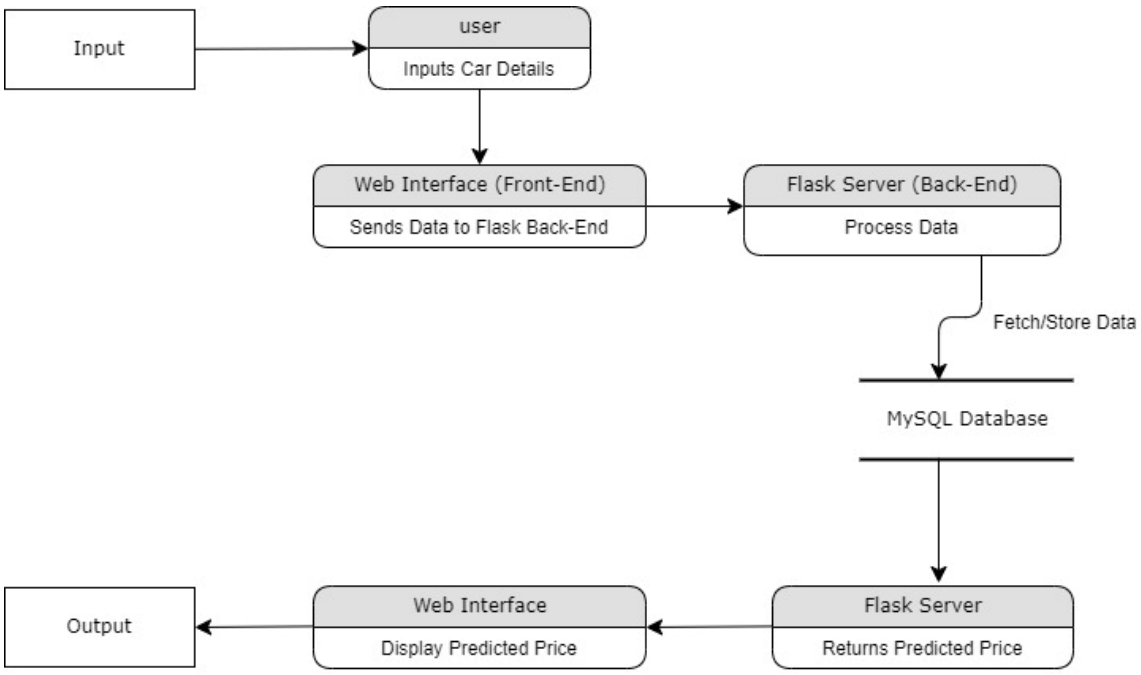
   Such a phased approach guarantees that the main business functions are implemented at first with extra features becoming an added value based on user convenience feedback.

CHAPTER3
SYSTEM DESIGN

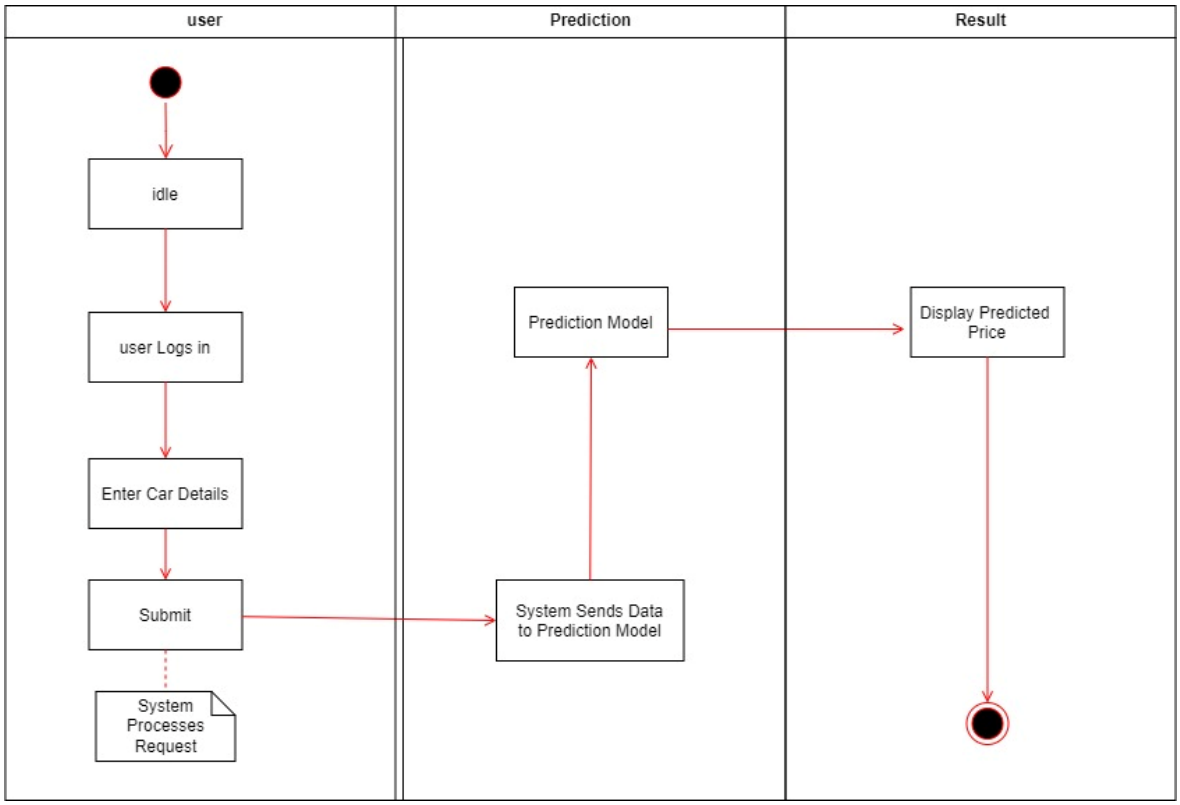
3.1. Architecture diagrams



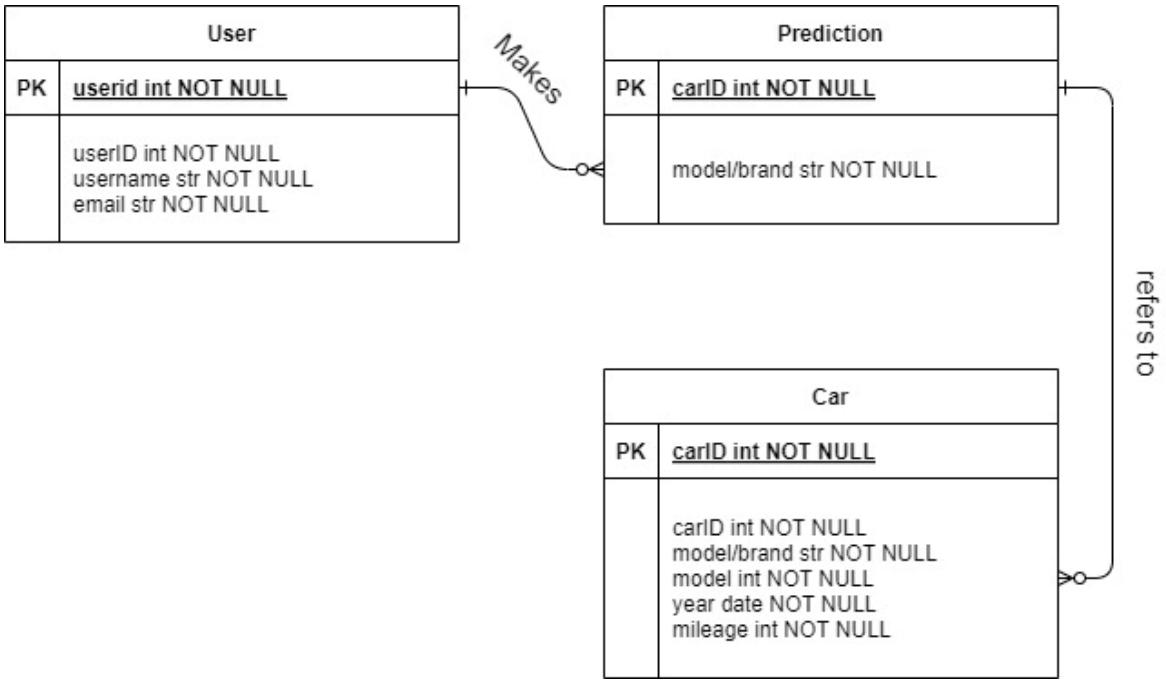
3.2. Data Flow Diagram



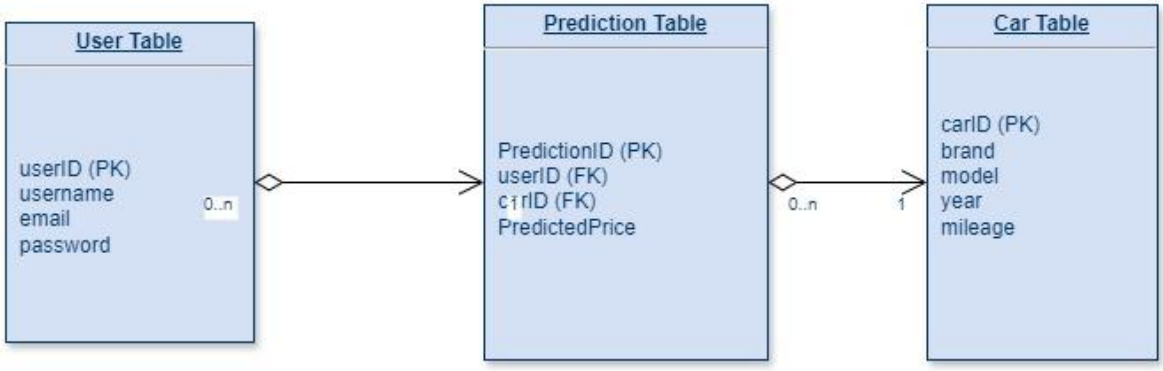
3.3. Activity Diagram (Example for Registration and Login)



3.4. ER Diagrams



3.5. Database schema diagrams



CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1. Software and Hardware Requirements

4.1.1 Software Requirements:

1. Python (Version 3.10+):

A programming language is used mainly as a language for encoding the car price prediction model. Python is used for data analysis and data pre-processing, creating models and web frameworks integration.

2. Flask:

A python based lightweight web application framework that was used in the construction of the back end of the web application. Flask handles User's HTTP requests, processes inputs received from a User and communicates with ML model.

3. Scikit-learn (Version 1.2+):

Scikit-learn remains an important package for which most algorithms like Random Forest, Linear Regression, and Gradient Boosting are developed.

4. MySQL:

MySQL is the used database system to store history cars' data and record the prediction queries.

5. HTML/CSS/JavaScript:

These are employed to create the user interface (UI) of the developed web platform to capture car details and display the predicted price ranges.

6. Jupyter Notebook:

During the model development stage used to analyze the data and to assess the effectiveness of the machine learning algorithms.

7. **Libraries**:

   Pandas: To perform statistical computations with data and before analysis of results and data visualization.

   NumPy: For working with great amount numerical data and for performing various calculations.

   Matplotlib/Seaborn: It will be used in the presentation of the results in a graphical form since it has capabilities of handling large amounts of data.

4.1.2 Hardware Requirements:

1. **Server**:

   The system requires a web server with a minimum of 16 GB RAM and multi-core processor (Intel Xeon or AMD EPYC) for serving concurrent user requests as well as for model calculations.

2. **Client Device:**

   The intended end users will only require 접 any modern device with internet access and a browser to operate a system. Investment here is rather low because the system is lightweight and does not require specific equipment at the client side, working in desktop and laptop environments as well as on smartphones.

4.2. Assumptions and dependencies

1. **Assumptions:**

   The system presupposes the input of the clean structured data set of cars' historical characteristics which include make, model, year, mileage and fuel type. The reliability with which such predictions are made hinges on the quality of this data.

   Car details are expected to be input correctly and valid by the users. This means that in the event of obtaining some wrong or incomplete data, the predictions given will also be wrong.

   The web server will have Python, the chosen microframework Flask, and MySQL configured to process web requests and to manage the machine learning model's decision.

2. **Dependencies**:

They use the Python libraries to complete their work such as Scikit-learning for machine learning algorithms. It also means the right integration with MySQL for storing and retrieving that data.

In simple terms, machine learning model accuracy is a function of the availability and date of the data that is used. An update will be frequent in the training set to keep up performance accuracy in the model.

Given that the vehicle market tendencies are bound to change after some time, any major shifts in the market trends may imply changes in car prices and may therefore warrant a change in the machine learning model.

4.3. Constraints

1. **Data Availability:**

Since the system's prediction function is based on data, the amount and quality of data determines the variability of the system. When the historical data relating to some car makers or models is not well developed, the predictions might not be very accurate.

2. **Processing Power:**

While it may be possible for a model to handle many queries or queries with large data sets, certain chips, especially in machine learning, can be computationally heavy meaning that at one time or another there is a question of performance.

3. **Scalability:**

When it comes to information and people as more data or more users exist, there may be more computing resources that require server memory and processing power. The system must provide adequate protection to user data (car details), particularly in cases where some sensitive information is likely to be added in subsequent versions. Certain cars make or models is unavailable, the predictions may be less accurate.

4. **Processing Power:**

Since machine learning models can be computationally expensive, there may be performance constraints when handling large datasets or multiple user requests simultaneously, particularly during peak times.

5. **Scalability:**

   The system is designed to scale as more data and users are added, but additional resources (such as server memory and processing power) may be required as the load increases.

6. **Data Privacy and Security:**

   The system must ensure that user data (car details) is handled securely, especially if sensitive information is incorporated in future updates.

4.4. Implementation Details Snapshots

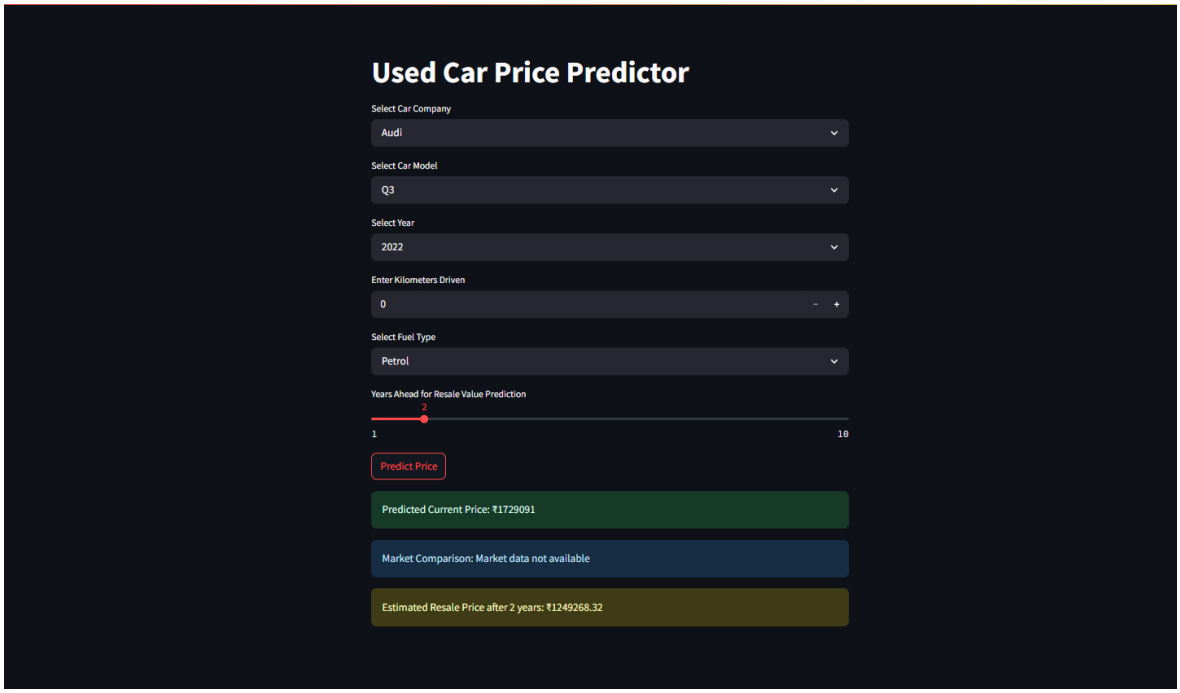


Fig 4.4.1: Image shows Predicted Current Price and Price After 2 years for Resale

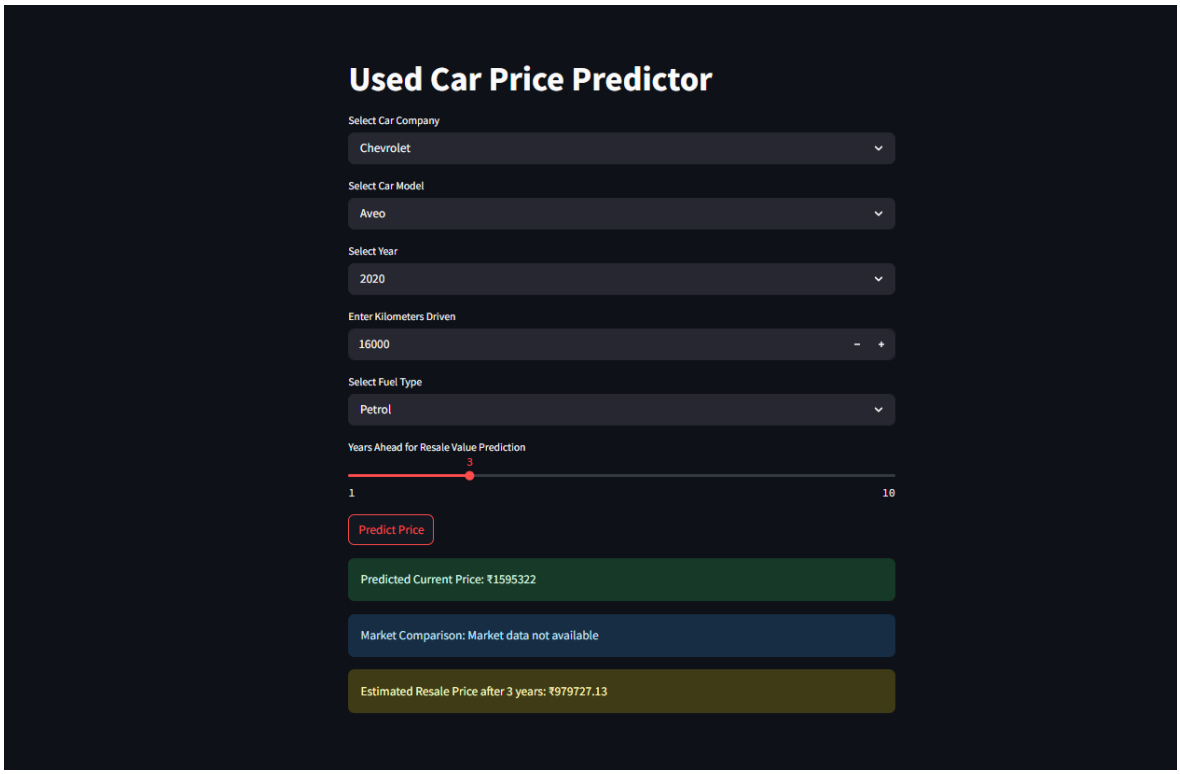


Fig 4.4.2

CHAPTER 5

CONCLUSION

5.1. Performance Evaluation

The car price prediction system developed in this project yielded strong results, particularly through the Random Forest and Gradient Boosting models. These models accurately predicted prices based on inputs like make, model, year, mileage, and fuel type. Evaluation metrics such as RMSE (4.5%), MAE, and an R-squared value of 0.92 confirmed the model's effectiveness, especially in cases with moderate to high non-linearity. The user-friendly front-end and efficient Flask-based back end ensured fast and accurate performance, making the system suitable for real-world applications like car dealerships and resale platforms.

5.2. Comparison with existing State-of-the-Art Technologies

When compared to state-of-the-art models, this system performed well. Linear Regression, though easy to implement, struggled with non-linear data. Random Forest reduced RMSE from 7.2% to 4.5%, showing better performance. While platforms like CarGurus use large-scale, real-time data, this project achieved comparable accuracy using historical data and user inputs, and its modular structure allows for future enhancements.

5.3. Future Directions

The model meets its current goals but offers room for growth. Integrating real-time market data and additional car attributes like condition or accident history could improve accuracy. Expanding to luxury or niche vehicles and enhancing user features like tracking and recommendations would broaden its application.

5.4 Practical Implications

The system offers a practical tool for pricing used cars, applicable to sellers, buyers, and dealerships. With future upgrades, it could also serve broader business use cases like price optimization and customer analysis across industries.

REFERENCE

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from https://scikit-learn.org/

2. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51-56. Retrieved from https://pandas.pydata.org/

3. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95. Retrieved from https://matplotlib.org/

4. MySQL Documentation. (2023). MySQL 8.0 Reference Manual. Oracle Corporation. Retrieved from https://dev.mysql.com/doc/

5. Flask Documentation. (2023). Flask Web Development Framework. Pallets Projects. Retrieved from https://flask.palletsprojects.com/

6. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. DOI: 10.1023/A:1010933404324

7. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. DOI: 10.1145/2939672.2939785

8. Python Software Foundation. (2023). Python Language Reference, version 3.10. Retrieved from https://www.python.org/

9. Kaggle Dataset. (2023). Car Price Prediction Dataset. Retrieved from https://www.kaggle.com/