A

MINI PROJECT REPORT

On

PREDICTION OF PRE-OWNED CAR PRICES USING MACHINE LEARNING

Submitted on the partial fulfilment of the Award of Degree in Bachelor of Technology in Computer Science and Engineering during academic year 2024-2025.

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD, TELANGANA.

Submitted by

Md. SALMA BEGAM	**-**	**21UC1A0546**
T. ARAVINDA SWAMY	**-**	**21UC1A0568**
G. PUJITHA	**-**	**21UC1A0525**
G. SHASHANK	**-**	**21UC1A0524**

Under the Guidance of

Mrs. J. SHILPA

Assistant Professor

Head of The Department of CSE

Department of Computer Science and Engineering

TALLA PADMAVATHI COLLEGE OF ENGINEERING

Accredited by NAAC

Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana.

TALLA PADMAVATHI COLLEGE OF ENGINERING

Accredited by NAAC

Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana.

BONAFIDE CERTIFICATE

This is to certify that this Project Report titled **"PREDICTION OF PRE-OWNED CAR PRICES USING MACHINE LEARNING"** is a bonafide work carried out by **Md. SALMA BEGAM (21UC1A0546), T. ARAVINDA SWAMY (21UC1A0568), G.PUJITHA (21UC1A0525), G.SHASHANK (21UC1A0524)** under the Supervision of **Mrs . J. SHILPA** Assistant Professor, HOD of Computer Science and Engineering in the partial fulfilment of the award of Bachelor of Technology in Computer Science and Engineering from Talla Padmavathi College of Engineering, Hanumakonda, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad, Telangana during the academic year 2024-2025.

This Project work does not constitute in part or full of any other works that have been earlier Submitted to this university or any other institutions for the award of any degree/diploma.

Internal Guide **Head of the Department**

External Examiner **Principal**

DECLARATION

We, **Md. SALMA BEGAM (21UC1A0546), T. ARAVINDA SWAMY (21UC1A0568), G.PUJITHA (21UC1A0525),G.SHASHANK (21UC1A0524)** final year students from Talla Padmavati College of Engineering, Warangal, affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana, solemnly declare that this Project titled " **PREDICTION PRE - OWNED CAR PRICES USING MACHINE LEARNING"** is a bona - fide work carried out by us under the Supervision of **Mrs. J. SHILPA,** Assistant Professor, HOD of CSE for the Award of Bachelor Of Technology in Computer Science and Engineering.

We also declare that this Project Work does not constitute in part or full of any other works that Have been earlier submitted to this University or any other institutions for the award of any Degree/diploma.

SI.NO	**SIGNATURE**	**NAME**	**ROLL NUMBER**
1.		Md. SALMA BEGAM	21UC1A0546
2.		T. ARAVINDA SWAMY	21UC1A0568
3.		G. PUJITHA	21UC1A0525
4.		G. SHASHANK	21UC1A0524

ACKNOWLEDGEMENT

We are grateful to our Chairman, **Mr. Talla Mallesham,** for providing us ambient learning Experience at our institution. We are greatly thankful to our Director, **Dr. Talla Vamshi**, and Directrix **Mrs. Chaitanya Talla Vamshi,** for their encouragement and valuable academic support in all aspects.We are thankful to our Principal, **Dr. R. Velu,** for his patronage towards our project and Standing as a support in the need of the hour. We would like to acknowledge and express our sincere thanks to our Guide **Mrs. J. Shilpa** Assistant Professor , HOD of Computer Science and Engineering for introducing the present Topic and for the inspiring Guidance, constructive criticism and valuable suggestions through Out our Project Work which have helped us in bringing out this proficient project. We also Thank all the faculty members of our institution for their kind and sustained support throughout Our program of study.

We thank our parents for their confidence that they have on us to be potential and useful technological graduates to serve the society at large.

Md. SALMA BEGAM	**- 21UC1A0546**
T. ARAVINDA SWAMY	**- 21UC1A0568**
G. PUJITHA	**- 21UC1A0525**
G. SHASHANK	**- 21UC1A0524**

ABSTRACT

The pre-owned car market is complex and dynamic, making price estimation challenging. Due to the unprecedented number of cars being purchased and sold, used car price prediction is a topic of high interest. Because of the affordability of used cars in developing countries, people tend more purchase used cars. Our project proposes a machine learning based approach to predict pre-owned car prices accurately. We collect a comprehensive dataset of historical car sales, including features such as manufacturer, model, year, mileage, condition and location. Our model leverages regression algorithms to identify key factors influencing prices. Hyperparameter tuning and feature engineering techniques optimize performance. An important qualification of a price prediction tool is that depreciation can be represented to better utilize past data for current price prediction. cars of a particular features start out with a price set by the manufacturer. As they age and resold as used, they are subject to supply-and-demand pricing for particular set of features, in addition to their unique history. The more this sets them apart from comparable cars, the harder they become to evaluate with traditional methods. Using Machine Learning algorithms to better utilize data on all the less common features of a car can more accurately assess the value of a vehicle. For this we have to develop a predictive model for pre-owned car prices and identify the significant factors affecting price of a car.

INDEX

CONTENTS Page No

CHAPTER -1

INTRODUCTION

The pre-owned car market has experienced significant growth in recent years, driven by increasing demand for affordable and reliable transportation.

However, determining the fair market value of a used car remains a challenging task. The complexity of the market, coupled with the numerous factors influencing car prices, accurate price estimation a daunting endeavor.

Traditional methods, such as relying on dealer expertise or using simplistic pricing guides, often yield inaccurate results. This opacity can lead to unfair pricing, dissatisfaction among buyers and sellers, and decreased market efficiency.

Machine Learning offers a promising solution to this problem. Our aim is to develop a predictive model using machine learning techniques to estimate pre-owned car prices. By analyzing historical sales data and incorporating relevant factors, our model will provide the Accurate price prediction for buyers, sellers and dealers, identification of key factors influencing prices and Data-driven insights for informed decision making.

The used car market is generally divided into two categories, retail and wholesale. The retail price is the higher of the two prices and is what an individual should expect when buying a car at a dealership.

The wholesale price is the lower price which dealers will pay. whether the dealer has sourced the car from a trade-in, auction, or another dealer, this price is considerably lower to ensure that the dealer will make a profit on the vehicle.

A difficulty for both parties to agree on a fair price. There is a need for a valuation method which can make use of more of the features particular to each car and extract information from all other previous sales of cars with shared features.

Regression algorithms are often used because they can predict continuous values like car prices. Some models that can be used include Linear Regression, Random Forest and Decision Tree.

The project can include a used interface that allows users to input information about a car and get an estimated price.

CHAPTER - 2

LITERATURE SURVEY

1. Prediction of used prices Using Machine Learning Techniques by M.I.A Miah, M.S Arefin and M.S. Islam. This paper represents the comparative study of different machine learning algorithms, including artificial neural networks, for predicting used car prices. The study uses a dataset of more than 100,000 used cars listing and compares the performance of different models in terms of accuracy, speed and computational efficiency (2022).
2. "Predicting Used car prices Using Neural networks" by A.F. Rahman, MA Uddin and M.M Islam. This paper presents a neural networks-based approach for predicting used car prices. The study uses a dataset of more than 10,000 used car listings and evaluates the performance of the model in terms of accuracy and computational efficiency (2020).
3. "Predicting Used car Prices Using machine Learning" by S.K. Singh (2020)- compared performance of Linear Regression, Decision Trees and Random Forest.
4. "Car price Prediction using Machine Learning Techniques" by V.K. Gupta. (2018)- Evaluated performance of support Vector Regression, Gradient Boosting, and Random Forest.
5. "Used Car Price Prediction using regression analysis and Machine learning Techniques" by A.K. Singh. (2019) - Explored Linear Regression, Ridge Regression, and Lasso Regression.
6. Comparative studies have evaluated the effectiveness of different algorithms, highlighting the superiority of ensemble methods and deep learning models over traditional statistical approaches (Bhatia et al., 2022). Future research directions include incorporating real-time data, exploring transfer learning across markets, and developing more interpretable models.

CHAPER - 3

SYSTEM ANALYSIS

Purpose of Project

The Purpose of the project is to provide Accurate Price Estimations for Buyers, Sellers and Dealers, Buyers can make better choices when buying a pre-owned car, Sellers can have a competitive price for their listings, Car manufacturers can benefit from the model. Create a model that can reliable estimate automobile prices based on many features such as brand, model, mileage, year, condition etc.

3.1 EXISTING SYSTEM

Considering the demand for private car all around the world, the demand of second-hand car market has been rising and creating a chance in business for both buyers and seller. In several countries, buying a used car is the best choice for customer because its price is reasonable and affordable by buyer. After few years of using them, it may get a profit from resell again. However, various factors influence the price of a used car such as how old of those vehicles and the condition in current scenario of them. Normally, the price of used cars in the market is not constant. Thus, car price evaluation model is required for helping in trading.

Predicting the price of used cars has not received much attention from academia despite its huge importance for the society. Bharambe and Dharamdhikhari (2015) used artificial neural networks (ANN) to analysis the stock market and predict market behavior. They claimed that their proposed approach is more accurate than existing ones by 25%. Pudaruth (2014) used four different supervised machine learning techniques namely KNN (K Nearest Neighbour), naïve bayes, linear regression and decision trees to predict the price of used cars. The best result was obtained using KNN which had a mean error of 27,000 rupees.

Ahangar (2010) also compared the use of neural networks with linear regression in order to predict the stock prices of companies in Iran. They also found that neural networks had superior performance both in terms of accuracy and speed compared to linear regression. Listiani (2009) used support vector machines (SVM) to predict the price of leased cars.

Disadvantages

Integration with Existing System: Integration ML models with legacy systems can be challenging.

Data Quality Problems: Noisy, missing or inconsistent data affects model accuracy.

Overfitting: Models may not accurately predict prices for uncommon configuration.

Model Maintenance: Continuously updating and refining models to maintain accuracy.

3.2 PROPOSED SYSTEM

The goal of our project is to predict the costs of used cars to enable the buyers to make informed purchase using the data collected form the various sources and distributed across various locations.

Our Project is about helping buyers to make an informed purchase by predicting the price of pre-owned cars.

The Proposed research work shows that, the predictive analytical models will be a great add-on to business mainly for assisting the decision-making process. Predictive Analytical is a process, where the businesses use statistical methods and technologies to analyze their historical data for delivering new insights and plan the future accordingly.

The major objective of our Project is to build a prediction model i.e., a fair price mechanism to predict the cars selling price based on their features like the car model, the number of years that a car is old, the type of fuel it uses, the type of seller, the type of transmission and the number of kilometers that the car has driven so far. Our project will help to get an approximation about selling price of a used car based on its features and reduces the seller and consumer risk in business

In order to carry out our Project, data have been obtained from different car websites and from the small found in daily newspapers like L ‘Express and Le Defi. The data was collected in less than one month interval because like other goods, the price of cars also changes with time and also data is collected from online marketplaces and dealership websites, including its features.

Next, the data undergoes preprocessing to clean it, handle missing values, and encode categorical variables.

We designed an interactive Web application for model deployment using the Flask framework. The proposed model utilizes the machine learning algorithms and regression techniques of statistics like liner, decision tree and random forest regressions to achieve this task.

Once the best-performance model is identified, it is deployed in a user-friendly application that allows users to input car details and receive price predictions. Finally the system is continuously improved by updating it with new data and incorporating user feedback to enhance its accuracy and usability.

Advantages

Improved Accuracy: Advanced machine learning algorithms and ensemble methods.

Real-Time Predictions: Integration with real-time market data for dynamic pricing.

Fair Pricing: Accurate valuations ensure fair prices for buyers and sellers.

Transparency: Clear and explainable pricing builds trust among customers.

Cost Reduction: Optimized pricing and inventory management reduce losses.

3.3 ALGORITHMS

1. Linear Regression
2. Random Forest
3. K-nearest Neighbours

Linear Regression Algorithm

Linear Regression is a statistical model and machine learning technique that estimates the relationships between a dependent variable and one or more independent variables. Linear Regression uses a linear equation to model the relationship between the variables. It uses the least squares to find the line that best fits the data by minimizing the distance between the line and the actual data points. Linear Regression can be used to predict the value of a dependent variable based on the value of an independent variable. Linear Regression uses a known data value to predict the value of an unknown data value. It estimates the coefficients the value of the dependent variable.

Random Forest Algorithm

The Random Forest Algorithm is a machine learning algorithm that uses multiple decision trees to make predictions. It's a popular choice for classification and regression problems, and is known for being flexible, easy to use, and robust to overfitting.

This algorithm creates many decision trees using a random selection of data points and features. When it is time to make a prediction, it outputs the most heavily weighted answer. The more trees in the forest, the higher its accuracy and problem-solving ability. The random forest algorithm is a machine learning technique that combines the results of multiple decision trees to produce a single outcome.

K-nearest Neighbours Algorithm

The K-nearest Neighbours algorithm is a supervised machine learning method employed to tackle classification and regression problem. KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection. Which uses proximity to make classifications or predictions about the grouping of an individual data point. It is one of the popular and simplest classification and regression classifiers used in machine learning. It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data. It is commonly associated with classification tasks, KNN can also be used for regression.

3.4 SYSTEM SPECIFICATIONS

Software Requirements:

Operating system: Min. Windows 7

Coding Language: Python

Hardware Requirements:

Processor: Intel core i5

RAM: 4GB

Hard Disk: 20GB

CHAPTER – 4

SYSTEM DESIGN

UML Diagrams

UML diagrams, or Unified Modeling Language diagrams, are a way to visualize software systems and processes using standardized modeling language. They can help with a variety of tasks. UML diagrams is general purpose, developmental, modeling language in the field of Software Engineering that is intended to provide a standard wat to visualize the design of a system. They are used to model software systems, workflows, and business processes.

UML diagrams break down systems into components and subcomponents, making it easier to visualize and plan projects, help with maintaining and documenting information about a system, can help different stakeholders understand a system at different levels of detail.

UML diagrams are created using a standard notation for many types a standard notation for many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams.

UML diagram is a general-purpose modeling language. The main aim of UML diagrams is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in another fields of engineering. UML is not a programming language, it is rather a visual language

Understanding and effectively using UML can significantly improve the quality and clarity of your software designs. UML diagrams are categorized into structural diagrams, behavioral diagrams, and also interaction overview diagrams. UML is used to specify, visualize, construct, and document the major elements of the software system.

It helps in designing and characterizing, especially those software systems that incorporate the concept of object orientation. It describes the working of both the software and hardware systems.

Here there are five types of UML diagrams for this Project namely:

1.Use Case Diagram

2.Class Diagram

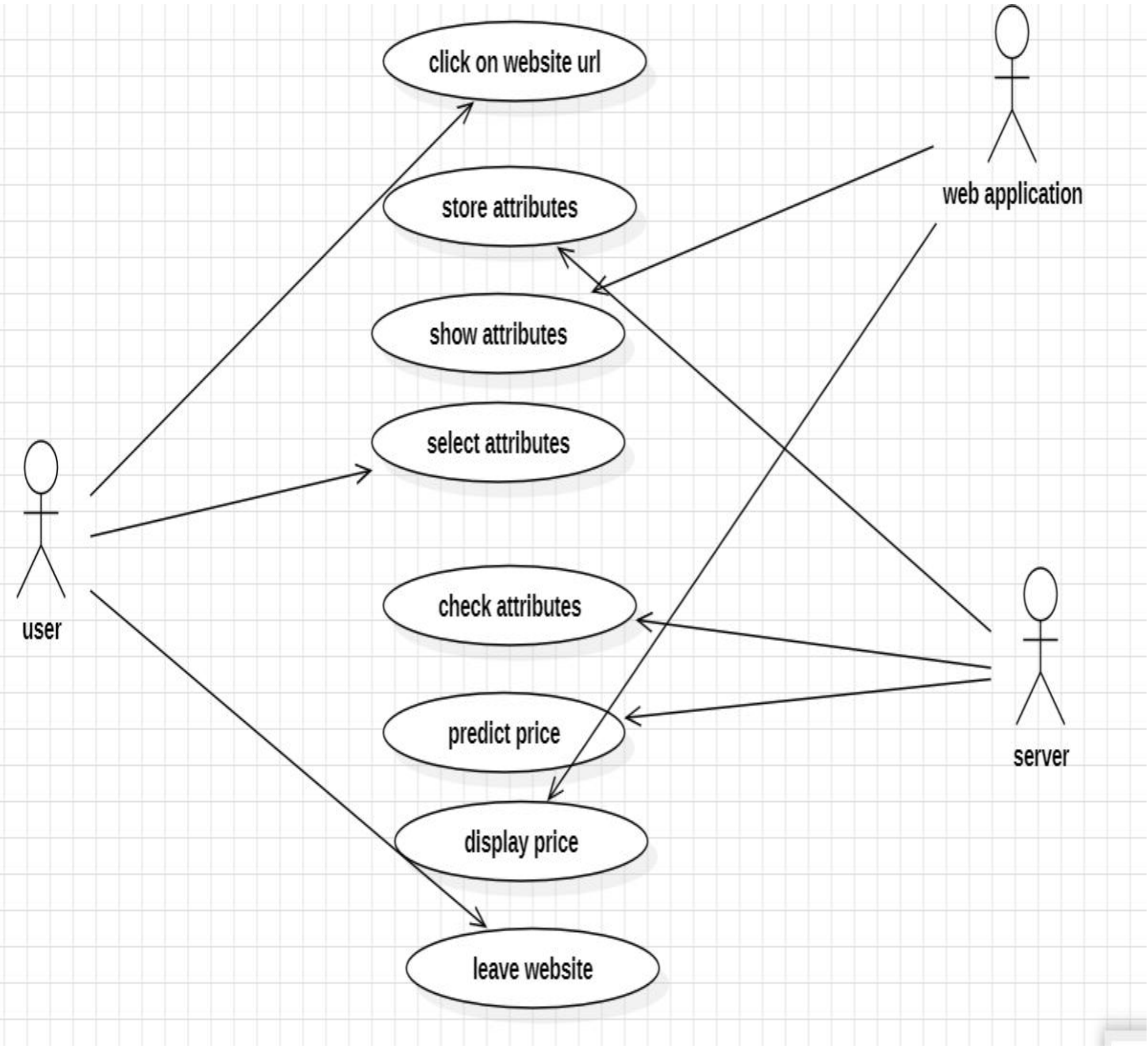
3.Sequence Diagram

4.State Chart Diagram

5.Collaboration Diagram

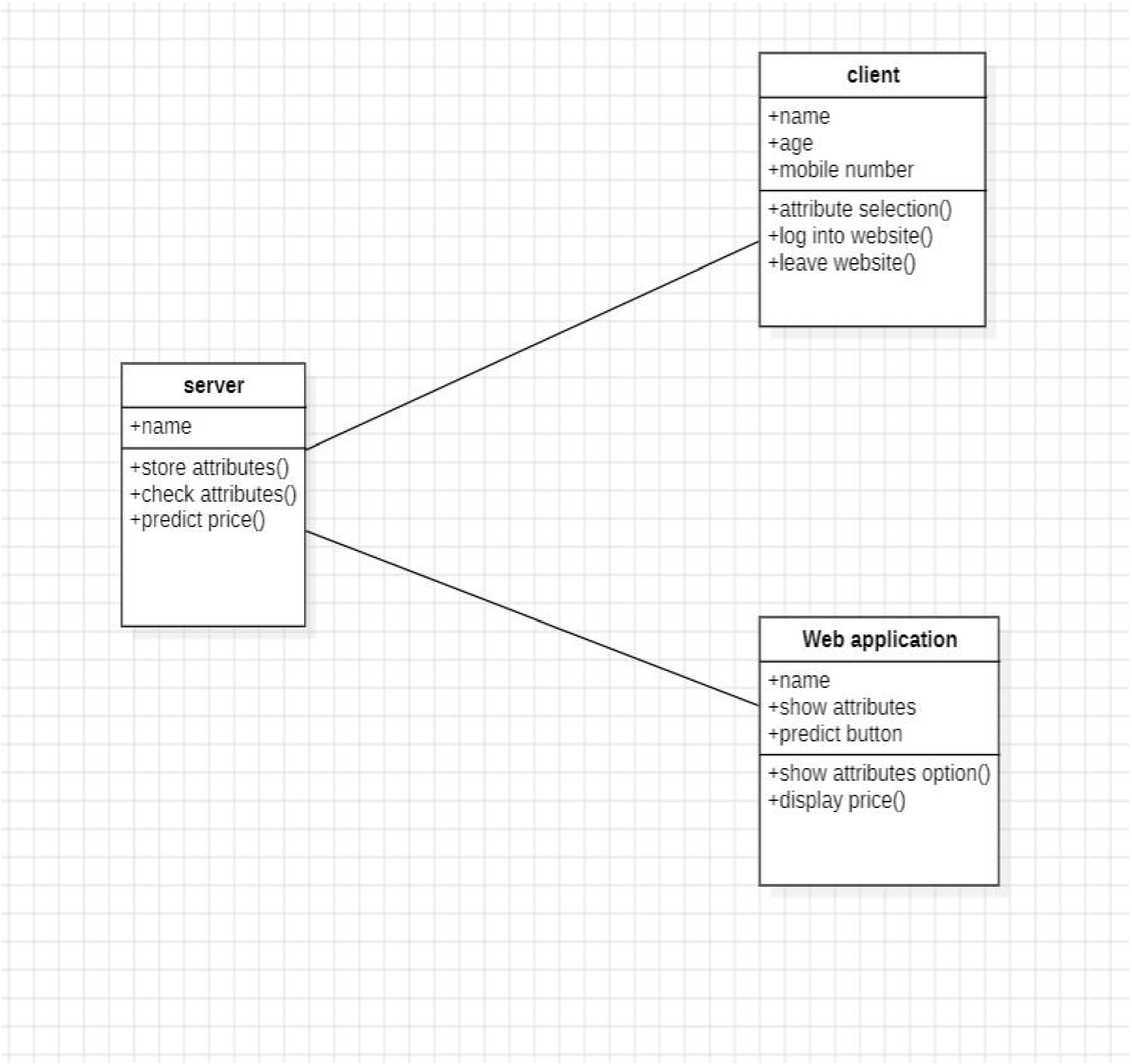
4.1 Use Case Diagram

A use case diagram is used to represent the d mic behavior of a system. It encapsulates the systems functionality. It depicts the high-leve nctionality of a system and also tells how the user handles a system Use Case diagrams are a set of Use Cases, actors and their relationships. They represent the use case view of a system. A Use Case represents a particular functionality of a system. So, Use Case diagram is used to describe the relationship among the functionalities and the internal external controllers. These controllers are known as actors. The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the systems requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.



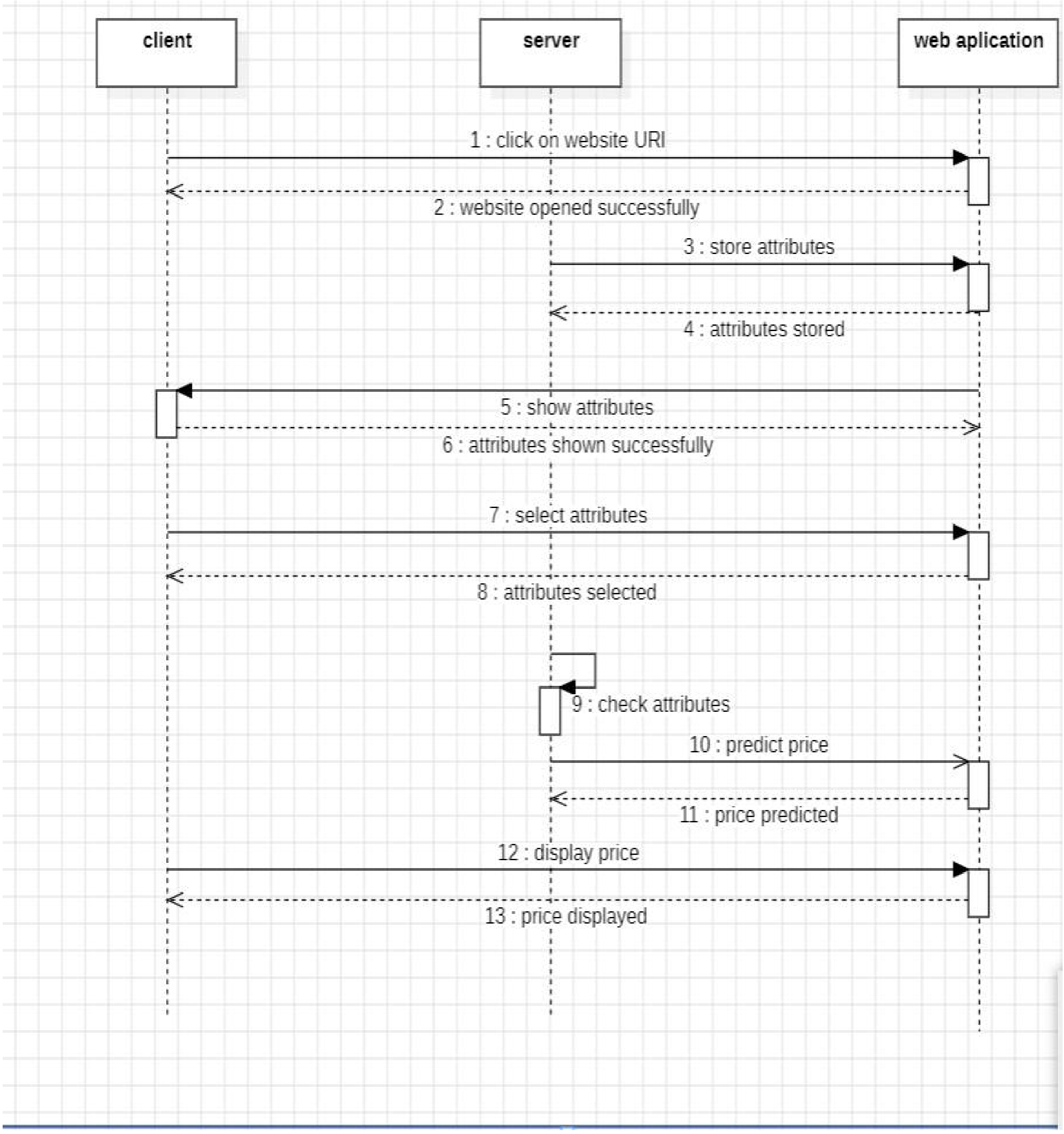
4.2 Class Diagram

Class diagrams are the most common used in U 8 lass diagrams model the static structure of a system, showing classes, attributes, operations, and relationships. Class diagram consist of classes, interface, associations and collaborations. Class diagrams basically represent the object view of a system which in static in nature. The class diagram depicts a static view of an application, it represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.



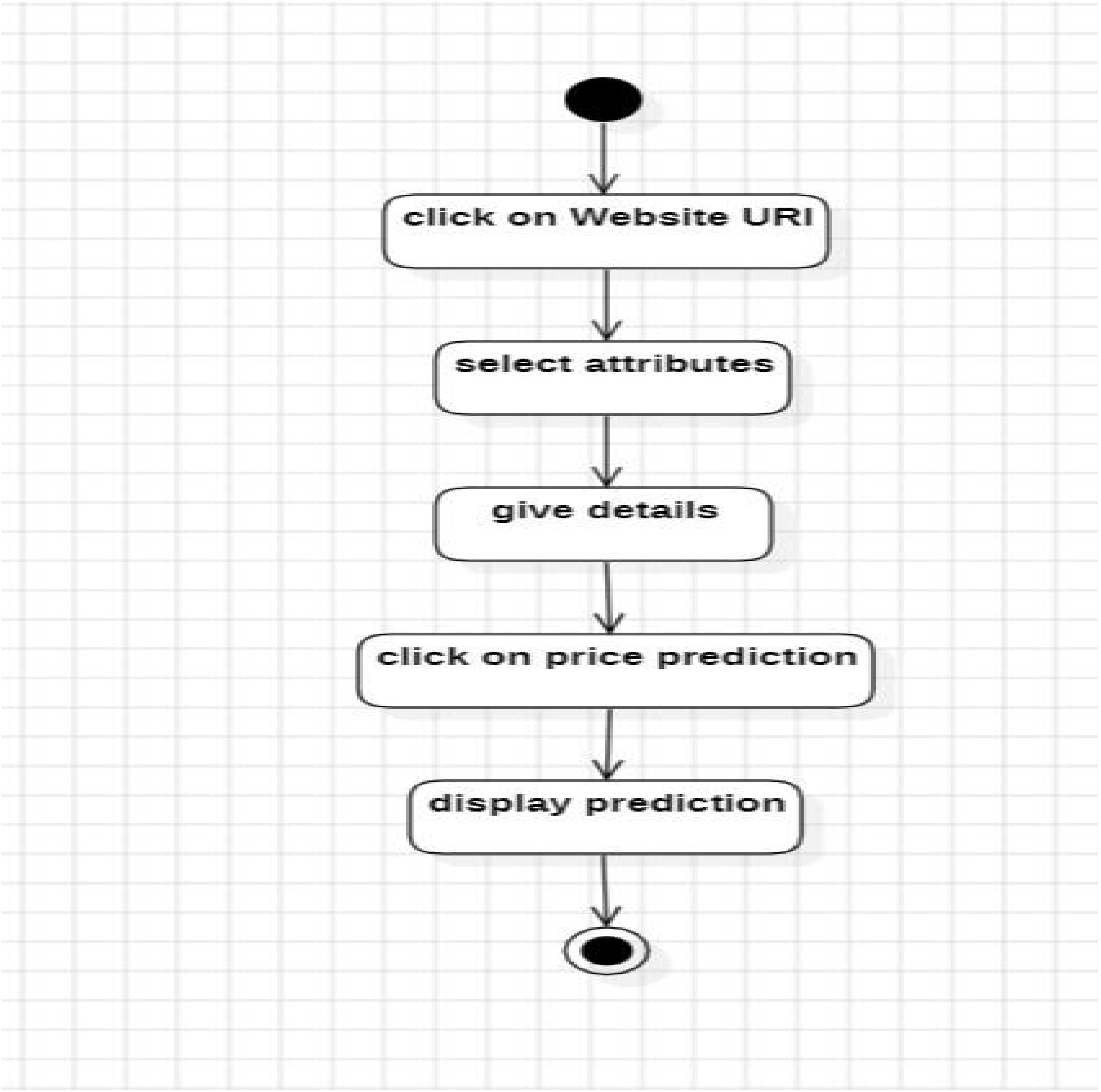
4.3 Sequence Diagram

Sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. Sequence diagrams model the dynamic behavior of a system, showing the interactions between objects over time. A Sequence diagram is an interaction diagram. From the name it clear that the diagram deals with some sequences, which are the sequence of messaging flowing from one object to another.



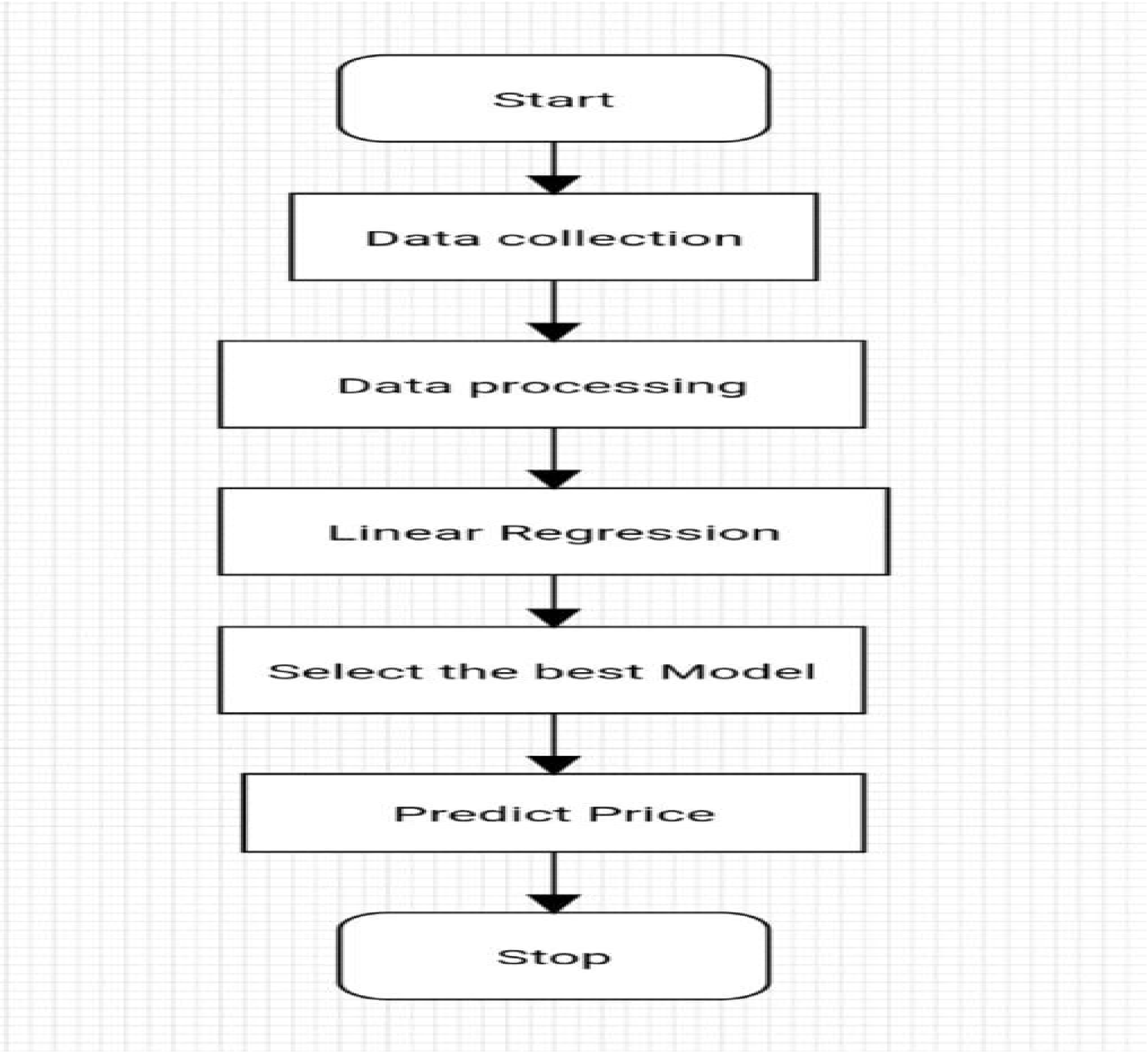
4.4 State chart Diagram

A Unified Modeling Language (UML) sta t diagram also known as a State Machine diagram, illustrates the sequence of states a nsitions. State represents an object's state at a specific moment in time. States are usually depicted as round-cornered rectangles with the state name inside. Transition indicates when an object will move from one state to another in response to an event. Events that are trigger state changes are written across the Transition arrows. State chart diagram is useful for Modeling the behavior of a system, class or interface and modelling reactive systems. A state chart diagram illustrates the dynamic view of a system.



4.5 Data-Flow diagram

A Data-flow diagram is a way of representing a flow of data through a process or a system. The Data-flow diagram also provides inforı about the outputs and inputs of each entity and the process itself. A data-flow diagram nas no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by flowchart. A data-flow diagram maps out the flow of information for any process or system. It uses defined symbols like rectangle, circles, and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flow diagram maps out the flow of information for any process or system.



CHAPTER-5

SOFTWARE EN ONMENT

5.1 Python Technology

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources newsgroup in 1991, and version 1.0 was released in 1994. Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2.

But in general, they remain not quite compatible. To be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do. Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator for Life) by the Python community.

The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation. Python is a general-purpose programming language, which is another way to say that it can be used for nearly everything. Most importantly, it is an interpreted language, which means that the written code is not actually translated to a computer-readable format at runtime. Whereas, most programming languages do this conversion before the program is even run. This type of language is also referred to as a "scripting language" because it was initially meant to be used for trivial projects.

On the down side, Python isn't easy to maintain. One command can have multiple meanings depending on context because Python is a dynamically typed language. And, maintaining a Python app as it grows in size and complexity can be increasingly difficult, especially finding and fixing errors. Users will need experience to design code or write unit tests that make maintenance easier.

Speed is another weakness in Python. Its flexibility, because it is dynamically typed, requires a significant amount of referencing to land on a correct definition, slowing performance. This can be mitigated by using alternative implementation of Python.

5.2 Python Libraries

Machine Learning, as the name suggests, is ence of programming a computer by which they are able to learn from different kinds o A more general definition given by Arthur Samuel is – "Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." They are typically used to solve various types of life problems. In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Python libraries that used in Machine Learning are: Numpy, Pandas,

NumPy: is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

Pandas: is a popular Python library for data analysis. It is not directly related to Machine Learning. Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics.

Flask: Flask is a lightweight and modular Python web framework that enables rapid development of web applications, APIs, and web services. Developed by Armin Ronacher, Flask is known for its flexibility, ease of use, and extensive libraries. It follows the Model-View-Controller (MVC) architecture and supports RESTful API development, unit testing, internationalization, and localization. Flask's core components include the Application Object, Routes, Views, Templates, and Request and Response Objects. With extensions like Flask-SQL Alchemy for database management, Flask-Login for authentication, and Flask-RESTful for API building, Flask streamlines web development. Ideal for small to medium-sized projects, prototyping, and data-driven applications, Flask powers various web applications, including blogs, portfolios, and machine learning model deployments. With a large community and extensive documentation, Flask remains a popular choice among developers.

CHAPTER 6

IMPLEMENTATION AND ANALYSIS

6.1 System Implementation

Data Collection for Machine Learning

Collecting Dataset, Pre-processing, Data cleaning, Data transformation, Data selection, Data input, Result Collecting Dataset. Data Collection is one of the most important tasks in building a machine learning model. We collect the specific dataset based on requirements from internet. The dataset contains some unwanted data also. So first we need to pre-process the data and obtain perfect data set for algorithm. It is the gathering of task related information based on some targeted variables to analyze and produce some valuable outcome. However, some of the data may be noisy, i.e. may contain inaccurate values, incomplete values or incorrect values. Hence, it is must to process the data before analyzing it and coming to the results. Data preprocessing can be done by data cleaning, data transformation, data selection. Data cleaning includes Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies Data transformation may include smoothing, aggregation, generalization, transformation which improves the quality of the data. Data selection includes some methods or functions which allow us to select the useful data for our system. Data input Dataset values converted into array values which is going to give to the algorithm to find accuracy. Select the algorithm based on the accuracy and analyze the data by using the algorithm. Result Based on that dataset we can get the result used our machine learning algorithm to predict the result. It will show the future values of particular stock. Network attacks are fortunately less frequent than normal traffic. This class imbalance can skew the model's learning. Techniques like oversampling (duplicating rare attack data) or under sampling (reducing normal traffic data) can be used to create a more balanced dataset. By collecting a comprehensive and well-prepared dataset, you can train a machine learning model to effectively detect cyberattacks and protect your network.

Data collection means pooling data by scraping, capturing, and loading it from multiple sources, including offline and online sources. High volumes of data collection or data creation can be the hardest part of a machine learning project, especially at scale. Data collection is a methodical practice aimed at acquiring meaningful information to build a consistent and complete dataset for a specific business purpose such as decision-making, answering research questions, or strategic planning. It's the first and essential stage of data-related activities and projects, including business intelligence, machine learning, and big data analytics. Data gathering also plays a key role in different steps of product management, from product discovery to product marketing. Yet it employs techniques and procedures different from those in machine learning and, thus, lies beyond the scope of this post.

6.2 Modules

- Data Collection
- Data Processing
- Data Splitting
- Applying ML Algorithms
- Comparing Performance
- Prediction

Data Collection

Data collection is the process of gathering and compiling data from various sources to build a dataset for machine learning model development.

It collects data from online marketplaces, dealer databases, government records

Data Processing

Data processing is the transformation of raw data into meaningful and useful information. It involves several steps to convert data into a format that can be analyzed, visualized, and user for decision-making. it cleans the data and preprocesses data, extract relevant features from data and checks the quality of data.

Data Splitting

Data splitting is a crucial step in machine learning that involves dividing a dataset into two or more subsets for training, validation, and testing purposes.

It uses the techniques like Random splitting, Stratified splitting, Time-series splitting.

Applying ML algorithms

Applying Machine Learning algorithms involves using various techniques to train and evaluate models on data.

In our project we Use Linear egression, Decision Trees, Random Forest algorithms.

Comparing Performance

Evaluate and compare the performance of different machine learning models.

It identifies the best-performing model, analyze the strengths and weaknesses of each model, and it Determines the most important features contributing to prediction accuracy.

Prediction

Generates predicted prices for pre-owned cars based on trained machine learning models, it uses the car attributes, market data.

6.3 CODE IMPLEMENTATION

FRONT – END:

```
import pandas as pd

import numpy as np

import pickle as pk

import streamlit as st
```

```
model = pk.load(open('model.pkl','rb'))
st.header('Car Price Prediction ML Model')
cars_data = pd.read_csv('Cardetails.csv')

def get_brand_name(car_name):
    car_name = car_name.split(' ')[0]
    return car_name.strip()

cars_data['name'] = cars_data['name'].apply(get_brand_name)

name = st.selectbox('Select Car Brand', cars_data['name'].unique())
year = st.slider('Car Manufactured Year', 1994,2024)
km_driven = st.slider('No of kms Driven', 11,200000)
fuel = st.selectbox('Fuel type', cars_data['fuel'].unique())
seller_type = st.selectbox('Seller  type', cars_data['seller_type'].unique())
transmission = st.selectbox('Transmission type', cars_data['transmission'].unique())
owner = st.selectbox('Seller  type', cars_data['owner'].unique())
mileage = st.slider('Car Mileage', 10,40)
engine = st.slider('Engine CC', 800,5000)
max_power = st.slider('Max Power', 0,300)
seats = st.slider('No of Seats', 5,10)

if st.button("Predict"):
    input_data_model = pd.DataFrame
([[name,year,km_driven,fuel,seller_type,transmıssıon,owner,mileage,engine,max_power,seats]
],columns=['name','year','km_driven','fuel','seller_type','transmission','owner','mileage','engine
','max_power','seats'])

    input_data_model['owner'].replace(['First Owner', 'Second Owner', 'Third Owner',
    'Fourth & Above Owner', 'Test Drive Car'],[1,2,3,4,5], inplace=True)

   input_data_model['fuel'].replace(['Diesel', 'Petrol', 'LPG', 'CNG'],[1,2,3,4], inplace=True)

   input_data_model['seller_type'].replace(['Individual', 'Dealer', 'Trustmark Dealer'],[1,2,3],
inplace=True)

   input_data_model['transmission'].replace(['Manual', 'Automatic'],[1,2], inplace=True)

   input_data_model['name'].replace(['Maruti', 'Skoda', 'Honda', 'Hyundai', 'Toyota', 'Ford',
'Renault', 'Mahindra', 'Tata', 'Chevrolet', 'Datsun', 'Jeep', 'Mercedes-Benz', 'Mitsubishi', 'Audi',
'Volkswagen', 'BMW', 'Nissan', 'Lexus', 'Jaguar', 'Land', 'MG', 'Volvo', 'Daewoo', 'Kia', 'Fiat',
'Force', 'Ambassador', 'Ashok', 'Isuzu',
'Opel'],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31],
inplace=True)

car_price = model.predict(input_data_model)

st.markdown('Car Price is going to be '+ str(car_price[0]))
```

BACK – END

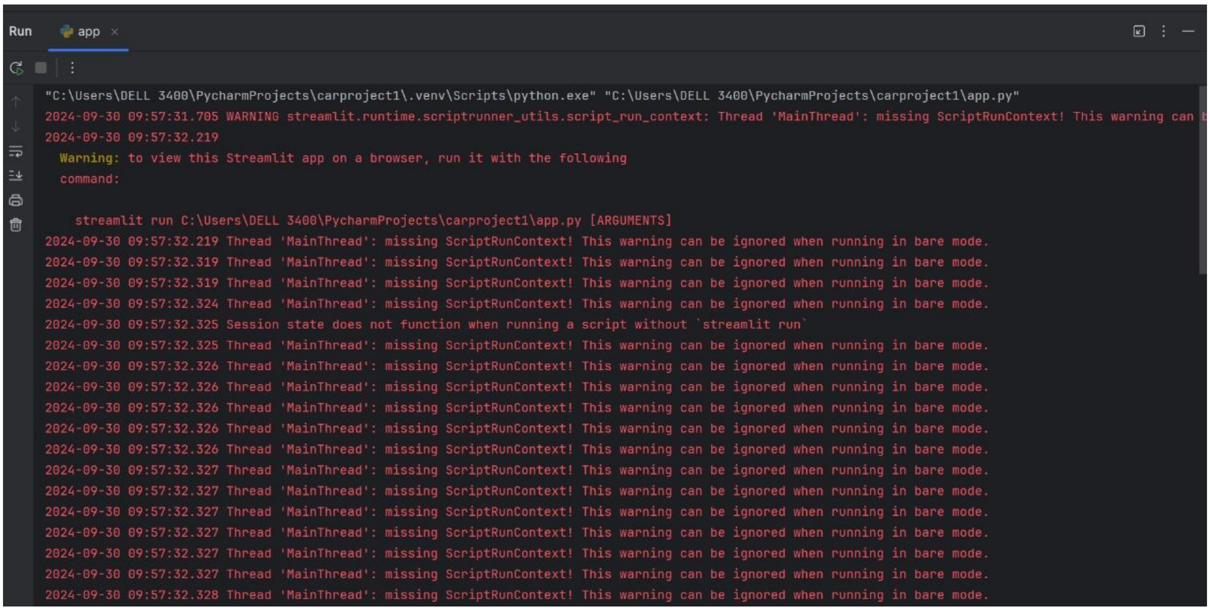
```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

cars_data = pd.read_csv('Cardetails.csv')

cars_data.head()

cars_data.drop(columns=['torque'], inplace=True)

cars_data.head()

cars_data.shape

#preprocessing

#NULL ChecK

cars_data.isnull().sum()

cars_data.dropna(inplace=True)

cars_data.shape

#Duplicate Check
```

```
cars_data.duplicated().sum()
cars_data.drop_duplicates(inplace=True)
cars_data.shape
cars_data
cars_data.info()
#Data Analysis
for col in cars_data.columns:
    print('Unique values of ' + col)
    print(cars_data[col].unique())
    print("=====================")
def get_brand_name(car_name):
    car_name = car_name.split(' ')[0]
return car_name.strip()
def clean_data(value):
    value = value.split(' ')[0]
    value = value.strip()
    if value == '':
        value = 0
    return float(value)
get_brand_name('Maruti  Swift Dzire VDI')
cars_data['name'] = cars_data['name'].apply(get_brand_name)
cars_data['name'].unique()
cars_data['mileage'] = cars_data['mileage'].apply(clean_data)
cars_data['max_power'] = cars_data['max_power'].apply(clean_data)
cars_data['engine'] = cars_data['engine'].ap [illegible] n_data)
for col in cars_data.columns:
    print('Unique values of ' + col)
    print(cars_data[col].unique())
    print("=====================")
```

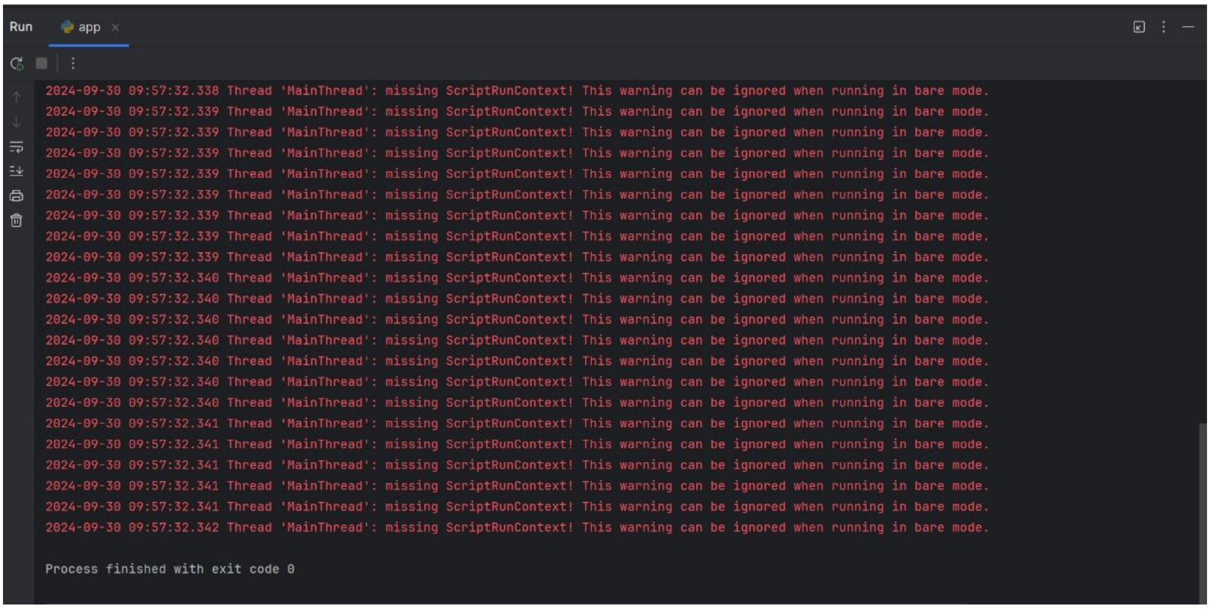
```
cars_data['name'].replace(['Maruti', 'Skoda', 'Honda', 'Hyundai', 'Toyota', 'Ford', 'Renault',
       'Mahindra', 'Tata', 'Chevrolet', 'Datsun', 'Jeep', 'Mercedes-Benz',
       'Mitsubishi', 'Audi', 'Volkswagen', 'BMW', 'Nissan', 'Lexus',
       'Jaguar', 'Land', 'MG', 'Volvo', 'Daewoo', 'Kia', 'Fiat', 'Force',
       'Ambassador', 'Ashok', 'Isuzu', 'Opel'],
   [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]
   ,inplace=True)
cars_data['transmission'].unique()
cars_data['transmission'].replace(['Manual', 'Automatic'],[1,2], inplace=True)
cars_data['seller_type'].unique()
cars_data['seller_type'].replace(['Individual', 'Dealer', 'Trustmark Dealer'],[1,2,3], inplace=True)
cars_data.info()
cars_data['fuel'].unique()
cars_data['fuel'].replace(['Diesel', 'Petrol', 'LPG', 'CNG'],[1,2,3,4], inplace=True)
cars_data.info()
cars_data.reset_index(inplace=True)
cars_data
cars_data['owner'].unique()
cars_data['owner'].replace(['First Owner', 'Second Owner', 'Third Owner','Fourth & Above Owner', 'Test Drive Car'], [1,2,3,4,5], inplace=True)
cars_data.drop(columns=['index'], inplace=True)
for col in cars_data.columns:
    print('------------')
    print(col)
    print(cars_data[col].unique())
cars_data.isnull().sum()
input_data = cars_data.drop(columns=['selling_price'])
output_data =cars_data['selling_price']
x_train, x_test, y_train, y_test = train_test_split(input_data, output_data, test_size=0.2)
```

```
#model Creation

model = LinearRegression()

#Train MOdel

model.fit(x_train, y_train)

predict = model.predict(x_test)

predict

x_train.head(1)

input_data_model = pd.DataFrame( [[5,2022,12000,1,1,1,1,12.99,2494.0,100.6,5.0]],

columns=['name','year','km_driven','fuel','seller_type','transmission','owner','mileage','engine','max_power','seats'])

input_data_model

model.predict(input_data_model)

import pickle as pk

pk.dump(model,open('model.pkl','wb'))
```

OUTPUT SCREENS

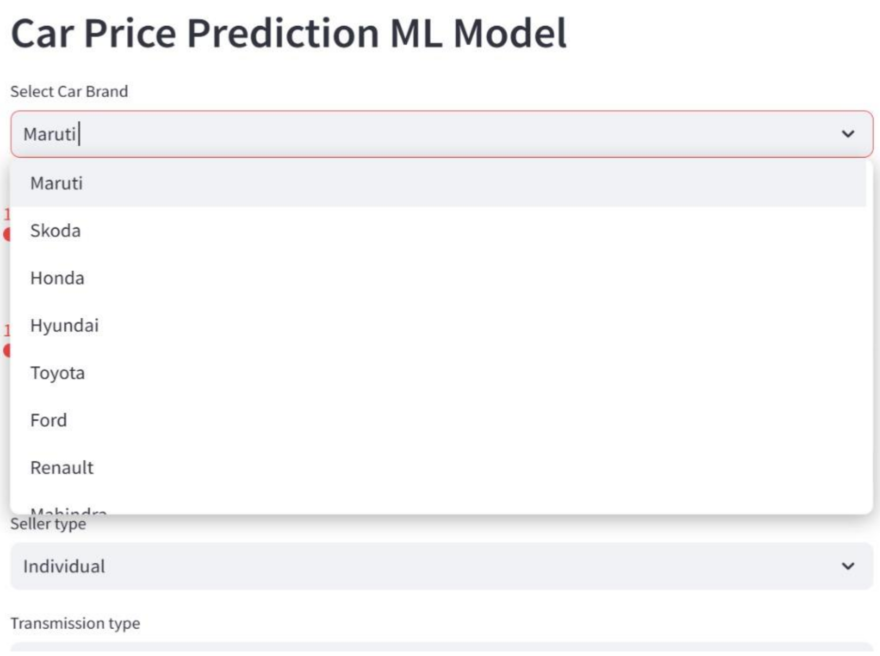


```
"C:\Users\DELL 3400\PycharmProjects\carproject1\.venv\Scripts\python.exe" "C:\Users\DELL 3400\PycharmProjects\carproject1\app.py"
2024-09-30 09:57:31.705 WARNING streamlit.runtime.scriptrunner_utils.script_run_context: Thread 'MainThread': missing ScriptRunContext! This warning can b
2024-09-30 09:57:32.219
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run C:\Users\DELL 3400\PycharmProjects\carproject1\app.py [ARGUMENTS]
2024-09-30 09:57:32.219 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.319 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.319 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.324 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.325 Session state does not function when running a script without `streamlit run`
2024-09-30 09:57:32.325 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.326 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.326 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.326 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.326 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.326 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.327 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.328 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
```



```
2024-09-30 09:57:32.338 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.339 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.340 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.341 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.341 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.341 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.341 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.341 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2024-09-30 09:57:32.342 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.

Process finished with exit code 0
```

Selecting Attributes from Website:

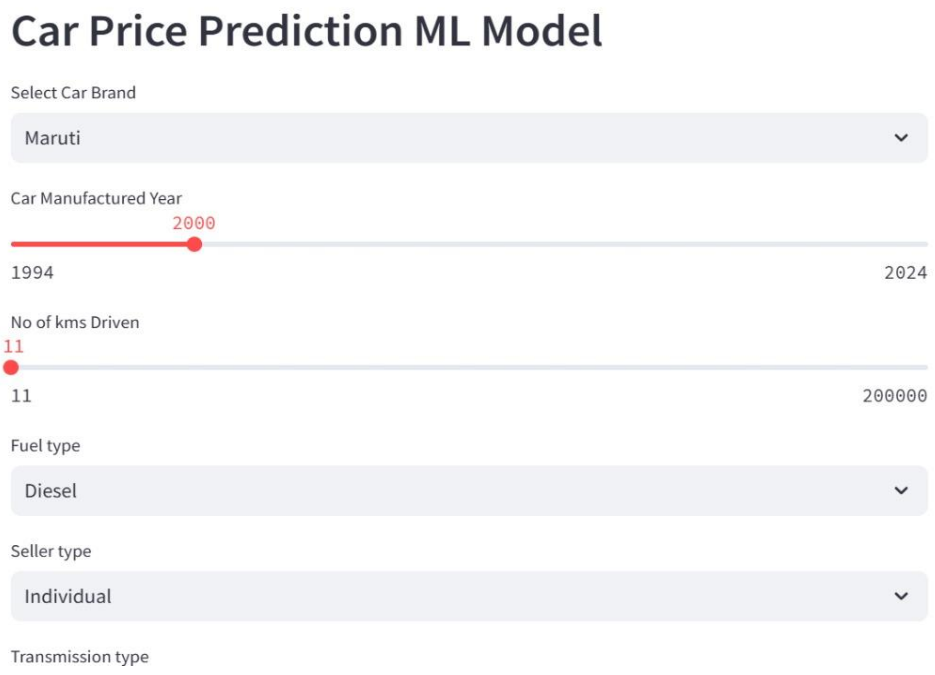


Car Price Prediction ML Model

Select Car Brand: Maruti

Seller type: Individual

Transmission type



Car Price Prediction ML Model

Select Car Brand: Maruti

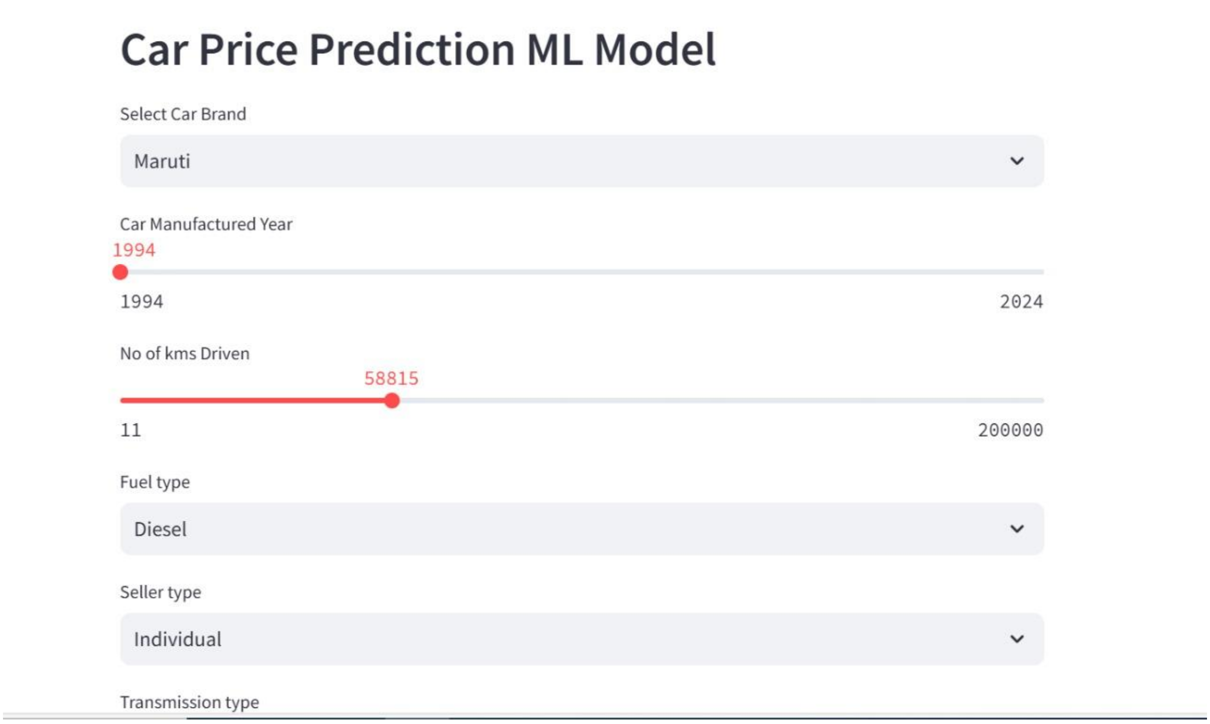
Car Manufactured Year: 2000

No of kms Driven: 11

Fuel type: Diesel

Seller type: Individual

Transmission type



Car Price Prediction ML Model

Select Car Brand

Maruti

Car Manufactured Year

1994

1994 2024

No of kms Driven

58815

11 200000

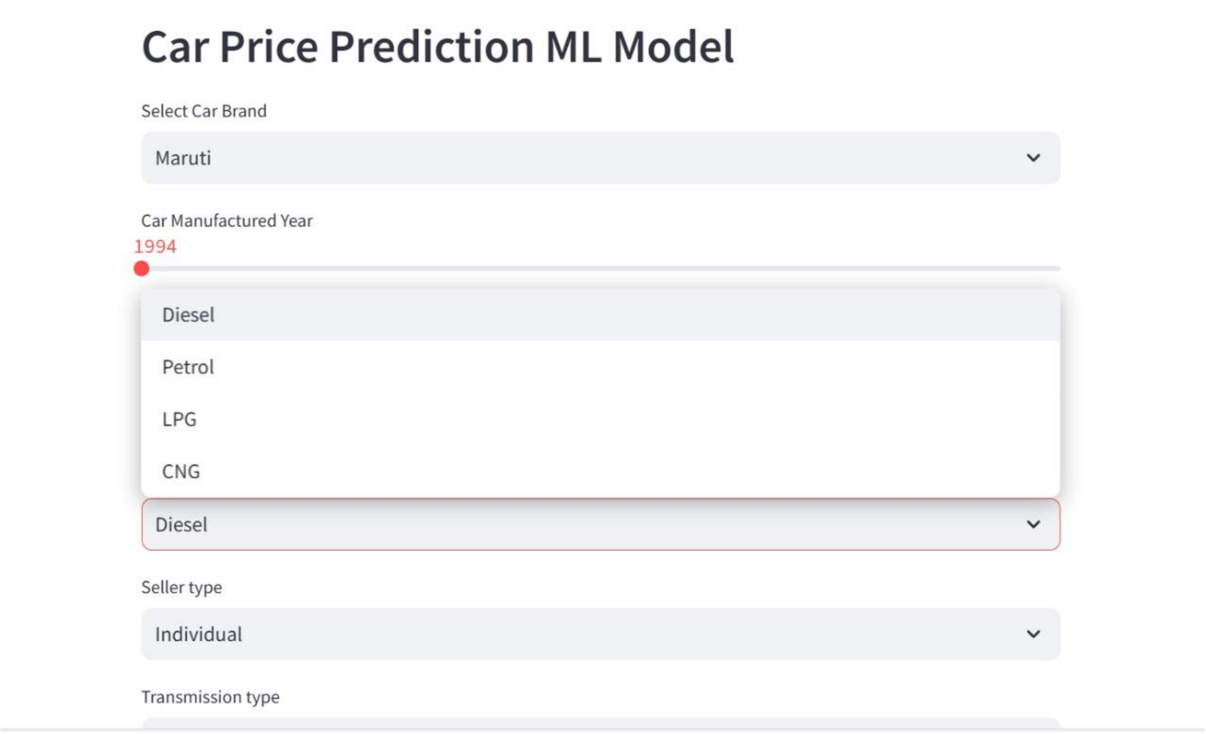
Fuel type

Diesel

Seller type

Individual

Transmission type



Car Price Prediction ML Model

Select Car Brand

Maruti

Car Manufactured Year

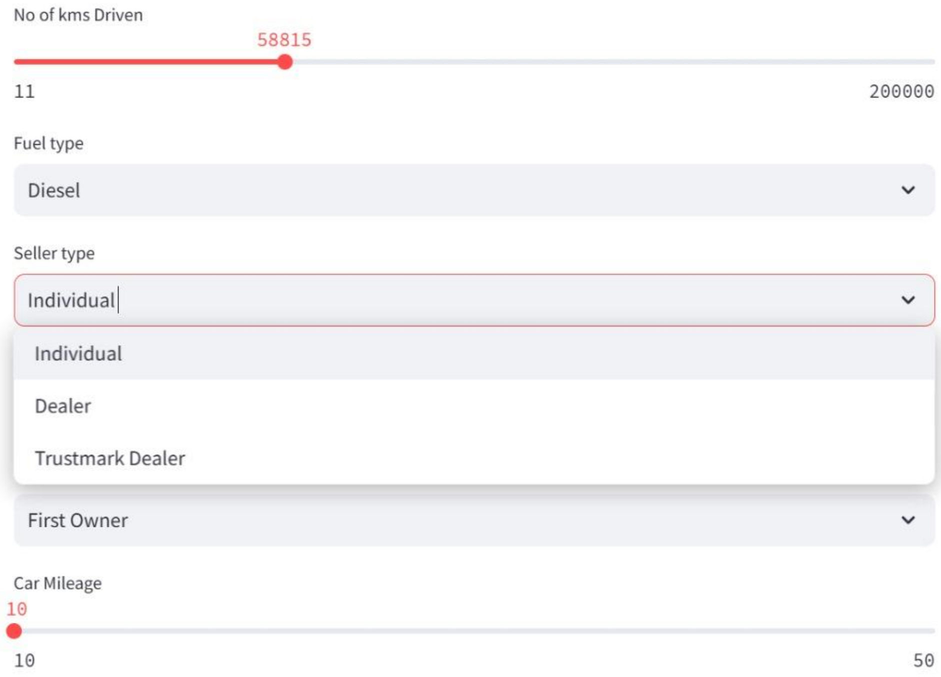
1994

Diesel

Seller type

Individual

Transmission type



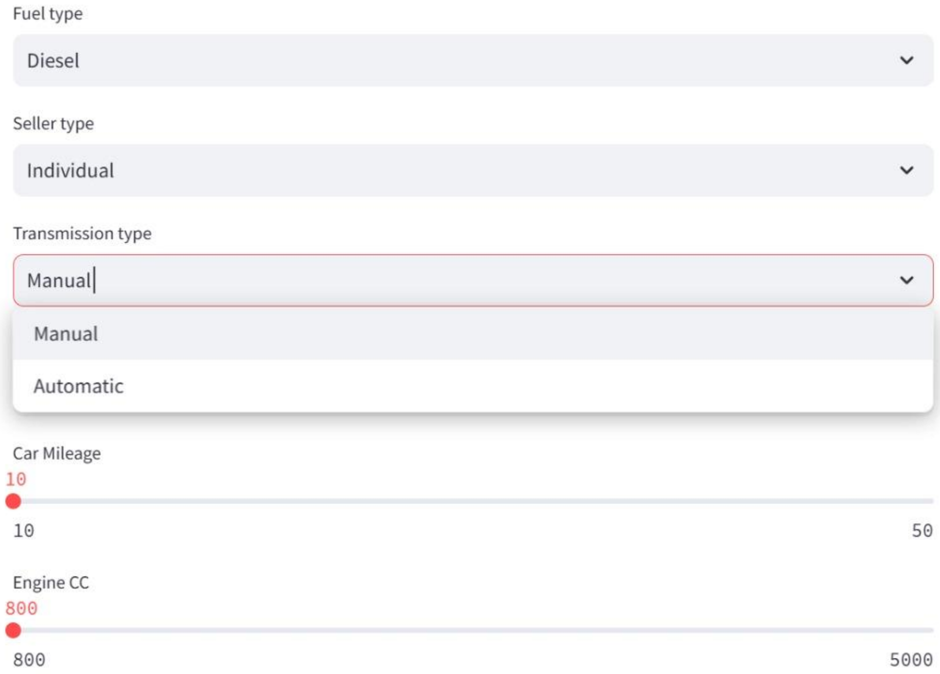
No of kms Driven: 58815

Fuel type: Diesel

Seller type: Individual

First Owner

Car Mileage: 10



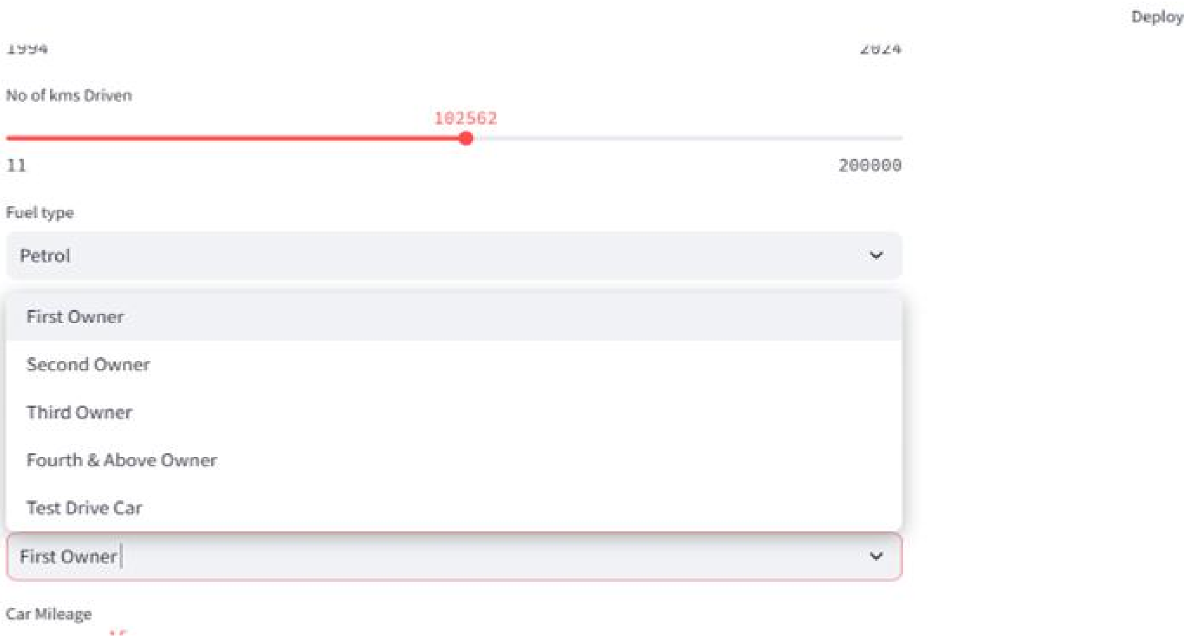
Fuel type: Diesel

Seller type: Individual

Transmission type: Manual

Car Mileage: 10

Engine CC: 800

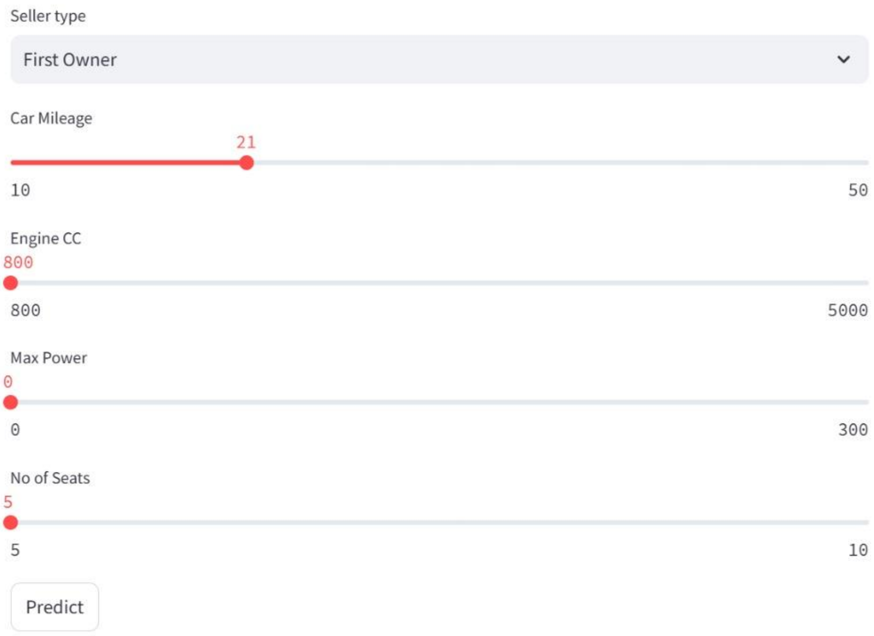


No of kms Driven
102562
11
200000

Fuel type
Petrol

First Owner

Car Mileage



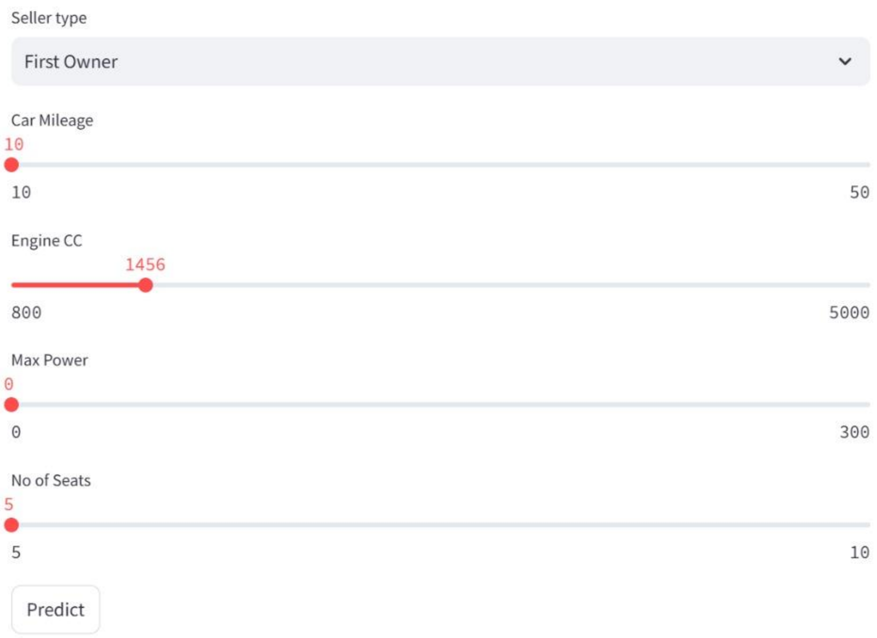
Seller type
First Owner

Car Mileage
21
10
50

Engine CC
800
800
5000

Max Power
0
0
300

No of Seats
5
5
10



Seller type

First Owner

Car Mileage

10

10 50

Engine CC

1456

800 5000

Max Power

0

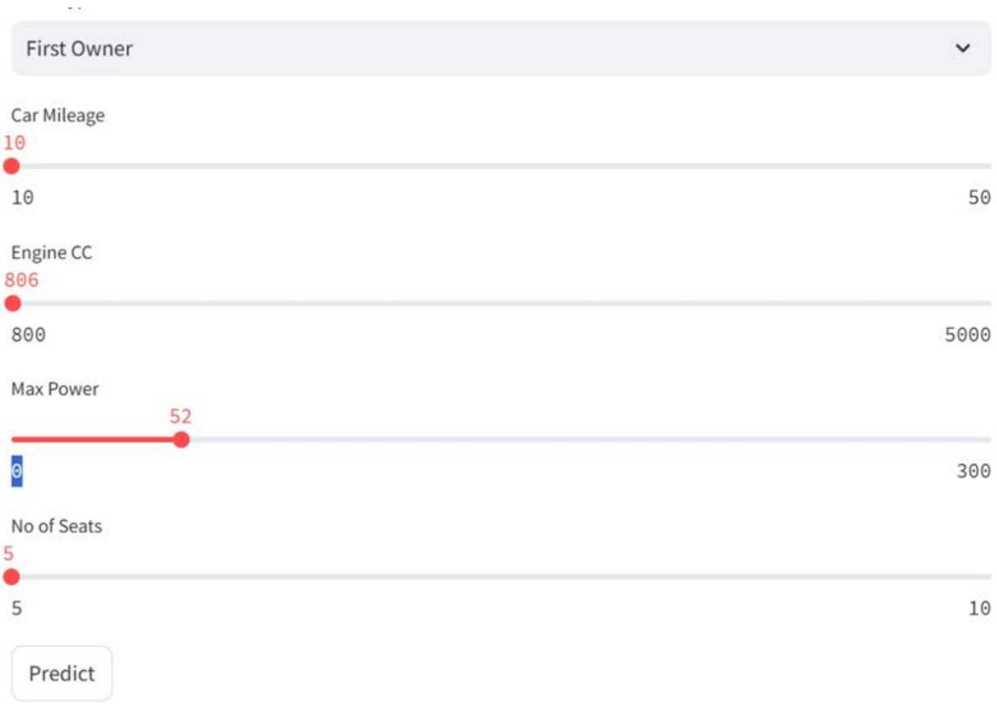
0 300

No of Seats

5

5 10

Predict



First Owner

Car Mileage

10

10 50

Engine CC

806

800 5000

Max Power

52

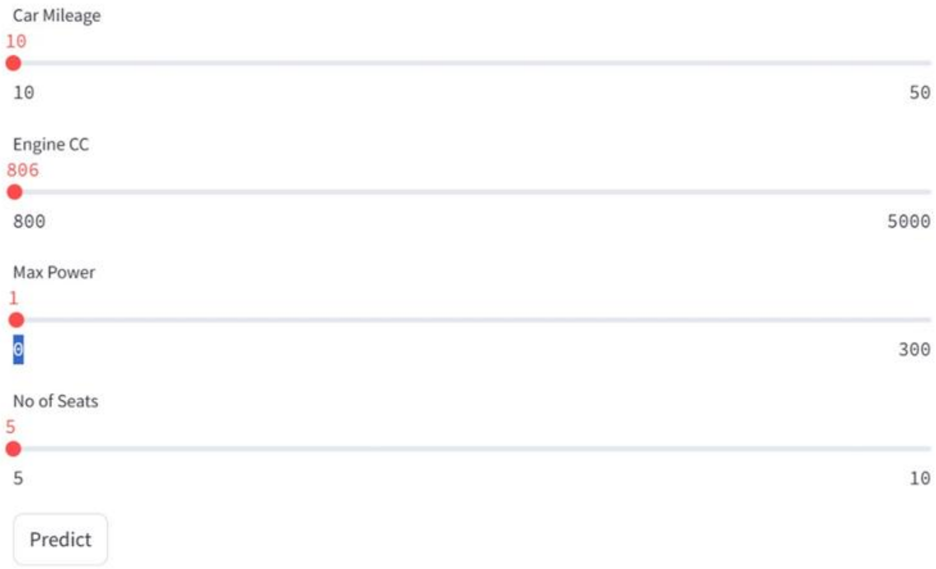
0 300

No of Seats

5

5 10

Predict



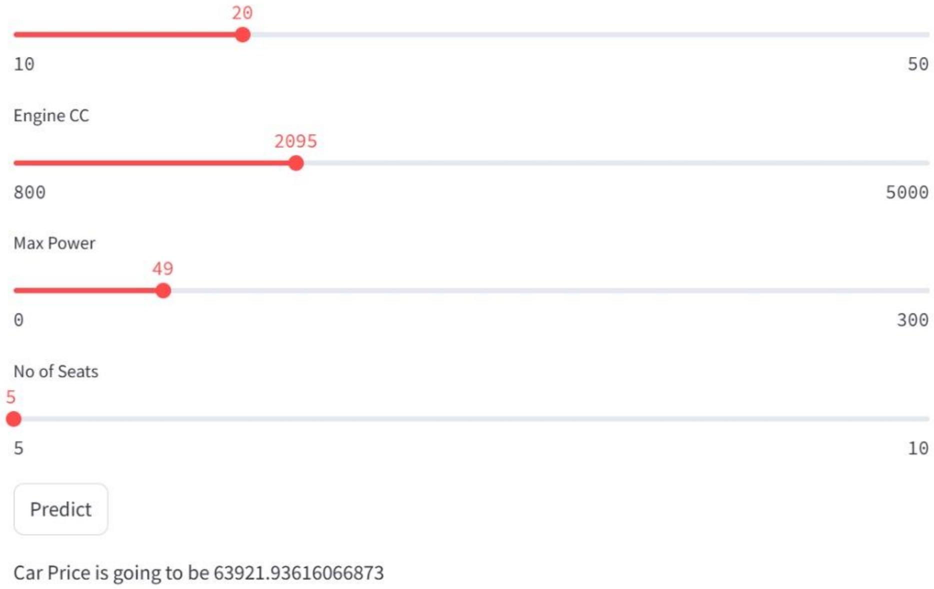
Car Mileage
10
10 50

Engine CC
806
800 5000

Max Power
1
0 300

No of Seats
5
5 10

Price predicted for pre-owned car:



20
10 50

Engine CC
2095
800 5000

Max Power
49
0 300

No of Seats
5
5 10

Car Price is going to be 63921.93616066873

CHAPTER-7

TESTING

Testing is a process of identifying the correctness of software by considering its all attributes like Readability, scalability, Portability, Re-usability, Usability. It evaluating the execution of software components to find the software bugs or errors or defects. Testing provides an independent view and objective of the software and gives surety of fitness of the software. It involves testing of all components under the required services to confirm that required services to confirm that whether it is satisfying the specified requirements or not. The process is also providing the client with information about the quality of the software Testing is mandatory because it will be a dangerous situation if the software fails any of time due to lack of testing. So, without testing software cannot be deployed to the end user. Testing is a group of techniques to determine the correctness of the application under the predefined script but, testing cannot find all the defect of application.

Types of Testing

1. System Testing
2. White Box Testing
3. Black Box Testing
4. Unit Testing

System Testing

System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution. It tests it the system meets the specified requirements and if it is suitable for delivery to the end-users. This type of testing is performed after the integration testing and before the acceptance testing.

White Box Testing

White box testing is a software testing technique that examines the internal structure of a program to ensure it works as expected. It is also known as clear box or glass box testing. It examines the code, logic, and structure of a program to identify issues.

Black Box Testing

Black box Testing is a software testing method that evaluates a systems functionality without knowing how its coded or designed. In this technique, the tester provides input and observes the systems output, simultaneously user activity to identify how the system responds. Black box testing can help identify issues with a systems usability, reliability, response time.

Unit Testing

Unit testing is the process where you test the smallest functional unit of code. Software testing helps ensure code quality, and its an integral part of software development. It’s a software development best practice to write software as small, functional units.

CHAPTER-8

CONCLUSION

The model we were making is to predict the value of a second- hand car using machine learning techniques. We have collected the data of cars from Kaggle having attributes like different cars and their year, km driven, fuel type, model name, company etc. The data is then processed using different algorithms where we chose linear regression and random forest algorithms and compared them. It would be available in GUI as a Web application developed using Python flask making it user-friendly so that users could give input and get the price of a car according to it. Using data machine and machine learning approaches, our project proposed a scalable framework for pre-owned car price prediction.

An efficient machine learning model is built by training, testing, and evaluating two machine learning regressors named random forest regression, linear regression. The increased prices of new cars and the financial incapability of the customers to buy them, used car sales are on a global increase. There is an urgent need for a Pre - Owned car price prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of pre - owned car price prediction.

Prediction of pre – owned car prices using machine learning is a complex task that require careful consideration of various factors, including data quality, future engineering, model selection and hyperparameter tuning. By leveraging machine learning algorithms and techniques, we can develop accurate and reliable predictive models that help buyers, sellers, and dealers make informed decisions.

CHAPTER-9

FUTURE SCOPE

The future scope of predicting pre - owned car prices using machine learning holds immense potential for growth and innovation. As the automotive industry continues to evolve, machine learning models will play a crucial role in providing accurate and personalized price predictions. Future developments will focus on integrating advanced technologies such as computer vision, natural language processing, and IOT devises to enhance data accuracy and model efficiency. Advances in explainable AI will provide stakeholders with actionable insights into pricing decisions, while continuous learning and updating of models will maintain predictive accuracy. Collaboration between industry experts, researchers, and policymakers will drive regulatory compliance and standardization. As the pre-owned car market continues to grow, machine learning-based price predictions will become increasingly essential for buyers, sellers, and dealerships. With ongoing innovation and refinement, these models will revolutionize the automotive industry, driving efficiency, transparency, and customer satisfaction. The expansion of predictive models to other industries, such as real estate and finance, will unlock new revenue streams and opportunities. The development of user-friendly interfaces and mobile applications will make price predictions accessible to a broader audience.

REFERENCES

1. Hsieh, M. H., Huang, C. Y., Lin, C. J., & Liou, J. J. (2017). Prediction of used car prices using artificial neural networks. Journal of Intelligent Manufacturing, 28(5), 1035-1044.

2. Jain, A., Kaur, P., & Jain, A. (2019). Prediction of Used Car Prices Using Machine Learning Techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1257-1260). IEEE.

3. Wang, X., Wu, J., & Xie, G. (2021). An empirical study of used car price prediction based on machine learning algorithms. Journal of Cleaner Production, 317, 128215.

4. Khoshgoftaar, T. M., & Van Hulse, J. (2010). An empirical study of learning from imbalanced data using random forest. In Proceedings of the 19th International Conference on Pattern Recognition (ICPR) (pp. 553-556). IEEE.

5. "Prediction of Used Car Prices Using Machine Learning Techniques" by M. I. A. Miah, M. S. Are fin, and M. S. Islam. (2022)

6. Predicting Used Car Prices Using Neural Networks (2022)" by A. F. Rahman, M. A. Uddin, and M. M. Islam.

7. "A Hybrid Machine Learning Approach for Predicting Used Car Prices" by S. Yousuf and M. Uddin (2022).