RESALE RECKONER

A PROJECT REPORT
Submitted by

MUHAMMED AJFAN T P
(SNG22MCA-2056)

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master Of Computer Applications

Department of Computer Applications
Sree Narayana Gurukulam College of Engineering,
Kadayiruppu, 682311
April 2024

RESALE RECKONER

PROJECT REPORT
Submitted by

MUHAMMED AJFAN T P
(SNG22MCA-2056)

Under the Guidance of
Asst. Prof. Meera K Chandran

to

the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree
of
Master Of Computer Applications

Department of Computer Applications
Sree Narayana Gurukulam College of Engineering,
Kadayiruppu, 682311
April -2024

DECLARATION

I undersigned hereby declare that the project report "**RESALE RECKONER**", submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under the supervision of **Assistant Professor. Meera K Chandran**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu **MUHAMMED AJFAN T P**
Date: **Reg No: SNG22MCA-2056**

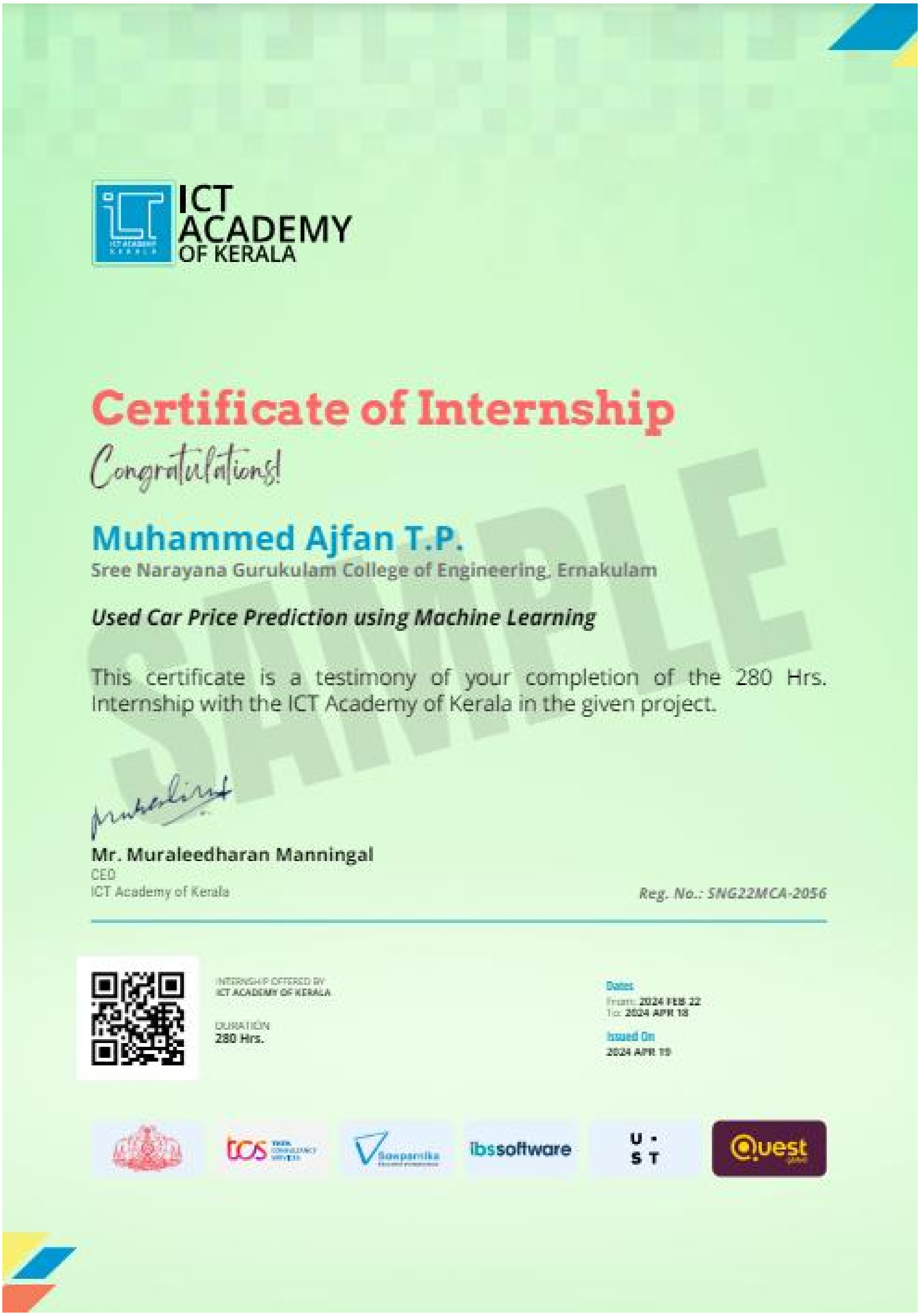
SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING, KADAYIRUPPU, KOLENCHERY- 682311

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report entitled "**RESALE RECKONER**" submitted by **MUHAMMED AJFAN T P** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications is a bona fide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose**.**

Asst. Prof. Meera K Chandran	**Prof. Dr Sandhya R**
Project Guide	Head of the Department
Department of Computer Applications	Department of Computer Applications



ICT ACADEMY OF KERALA

Certificate of Internship

Congratulations!

Muhammed Ajfan T.P.

Sree Narayana Gurukulam College of Engineering, Ernakulam

Used Car Price Prediction using Machine Learning

This certificate is a testimony of your completion of the 280 Hrs. Internship with the ICT Academy of Kerala in the given project.

Mr. Muraleedharan Manningal
CEO
ICT Academy of Kerala

Reg. No.: SNG22MCA-2056

INTERNSHIP OFFERED BY
ICT ACADEMY OF KERALA

DURATION
280 Hrs.

Dates
From: 2024 FEB 22
To: 2024 APR 18

Issued On
2024 APR 19

ACKNOWLEDGEMENT

I thank GOD almighty for guiding me throughout the project. I would like to thank all those who have supported to the completion of the project and helped me with valuable suggestions for improvement.

I owe my deep sense of gratitude to the Management of Sree Narayana Gurukulam College of Engineering, providing the best suited academic environments for the fulfilment of my project. I would like to place on record my sincere thanks to **Prof. Dr. S. Jose** Principal of our institution, for his valuable comments and suggestions to this project.

I wish to express my sincere thanks to our Head of the Department **Prof. Dr. Sandhya R**, for providing me the opportunity to undertake this project. I'm deeply indebted to my project guide **Asst.Prof. Meera K Chandran**, and project coordinator **Asst.Prof Dhanya Sukumaran** in the Department of Computer Applications for providing me with valuable advice and guidance during the course of the project. I thank all faculty members of our department and friends for extending their cooperation during our project.

Finally, I convey my thanks to each and every one, who helped me directly and indirectly to carry out this project successfully.

MUHAMMED AJFAN T P

ABSTRACT

"ResaleReckoner: A Machine Learning-Powered Web Application for Used Car Price Prediction"

ResaleReckoner is a web application leveraging advanced machine learning techniques to estimate the resale value of cars based on input parameters like model, manufacturing year, fuel type, and kilometers driven. Beginning with data preprocessing and feature engineering on a diverse dataset of vehicle information, the project trains a machine learning model for accurate price prediction. Integrated into a user-friendly interface, ResaleReckoner allows users to input their car details and receive instant estimates of resale value, with filtering options for specific criteria. Empowering users to make informed decisions in the automotive market, ResaleReckoner exemplifies the practical application of machine learning in addressing real-world challenges

CONTENTS

RESALE RECKONER

A Machine Learning-Powered Web Application for Used Car Price Prediction

1. INTRODUCTION

In the ever-evolving automotive market, accurately gauging the resale worth of a used car stands as a pivotal concern for both prospective buyers and sellers alike. Numerous variables such as the car's model, manufacturing year, fuel type, and mileage intricately interplay to influence its market value, rendering manual estimation a daunting and often imprecise endeavor.

Addressing this challenge head-on, ResaleReckoner emerges as a beacon—a machine learning-driven web application meticulously crafted to furnish users with dependable estimates of used car prices. By harnessing the prowess of sophisticated algorithms and an extensive repository of vehicle data, ResaleReckoner proffers a seamless and expedient avenue for users to ascertain the potential value of their vehicles.

This introduction endeavors to furnish an expansive panorama of ResaleReckoner's objectives, elucidate its methodological underpinnings, and underscore its consequential implications in arming users with the requisite insights to navigate the automotive realm with acumen. Through ResaleReckoner, users are not merely equipped with a tool; rather, they are endowed with a potent instrument that augments transparency and efficacy in the vehicular exchange, ultimately fostering a landscape characterized by informed decision-making and equitable transactions.

2. SYSTEM STUDY AND ANALYSIS

2.1 Existing System

Traditionally, estimating the resale value of used cars has relied heavily on manual processes and subjective assessments. Sellers often base their pricing decisions on personal experience, market trends, or advice from automotive experts. However, this approach is prone to inaccuracies and biases, leading to disparities between perceived and actual values.

In the absence of standardized tools or platforms, users typically resort to online classifieds, forums, or dealership valuations to gather information about similar cars and their selling prices. While these sources offer some insights, they lack the sophistication and predictive power afforded by advanced data analytics and machine learning algorithms.

Moreover, the lack of transparency in the valuation process can result in frustration and dissatisfaction among users, particularly when they encounter discrepancies between their expectations and the offers they receive. As a result, there is a clear need for a more efficient, accurate, and user-friendly solution to facilitate the estimation of used car prices, thereby enhancing confidence and transparency in the automotive resale market.

2.2 Proposed System

The proposed system, ResaleReckoner, aims to revolutionize the process of estimating the resale value of used cars by leveraging cutting-edge machine learning techniques and a comprehensive dataset of vehicle information. Unlike traditional methods reliant on subjective assessments, ResaleReckoner employs objective data-driven models to provide users with accurate and reliable price predictions.

By analyzing key factors such as the car's model, manufacturing year, fuel type, mileage, and ownership history, ResaleReckoner generates personalized estimates tailored to each user's specific vehicle characteristics. Through an intuitive web interface, users can input their car details and receive instant insights into its market value, empowering them to make informed decisions when selling or purchasing used cars.

Furthermore, ResaleReckoner offers transparency and consistency in the valuation process, mitigating the potential for discrepancies and misunderstandings between buyers and sellers. By democratizing access to advanced valuation tools, the proposed system fosters a more equitable and efficient automotive resale market, ultimately benefiting both consumers and industry stakeholders alike.

2.3 Feasibility Study

A feasibility study is carried out to select the test system that meets the performance requirements. The main aim of the feasibility study is to determine whether it would be technically feasible to develop the website. The feasibility study involves the activity of user and admin. It is a test of a system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer for see the future of the project and the usefulness. A feasibility study of system proposal is according to its work ability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic, Operational and social feasibility's.

2.3.1 Technical Feasibility

- Data Availability: ResaleReckoner relies on a rich dataset of vehicle information, which may be readily accessible through various sources such as online databases, APIs, or data scraping techniques.
- Machine Learning Expertise: Development of machine learning models for price prediction requires expertise in data preprocessing, feature engineering, model selection, and evaluation, which may necessitate the involvement of skilled data scientists or machine learning engineers.
- Web Development: Building a user-friendly web interface for ResaleReckoner involves frontend and backend development, database management, and integration of machine learning models, requiring proficiency in web technologies such as HTML, CSS, JavaScript, and frameworks like Django or Flask.

2.3.2 Economical Feasibility

- Cost of Development: The economic feasibility of ResaleReckoner depends on factors such as personnel costs, data acquisition expenses, software licenses, and infrastructure requirements. A detailed cost-benefit analysis should be conducted to assess the project's financial viability.

- Revenue Model: Potential revenue streams for ResaleReckoner may include subscription fees for premium features, advertising, partnerships with automotive industry stakeholders, or data monetization strategies.

2.3.3 Operational Feasibility

- User Adoption: The success of ResaleReckoner hinges on user adoption and engagement. Conducting market research and user testing can help identify user preferences, pain points, and feature requirements to ensure the application meets user needs effectively.

- Scalability: The system should be designed to accommodate potential growth in user base and data volume over time. Scalability considerations should be factored into the architectural design and infrastructure planning to ensure smooth operation as the platform expands.

2.3.4 Legal and Ethical Feasibility

- Data Privacy and Compliance: ResaleReckoner must comply with relevant data privacy regulations such as GDPR or CCPA to ensure the protection of user data. Implementing robust security measures and obtaining necessary permissions for data usage are essential to maintain legal and ethical integrity.

- Fair and Transparent Pricing: The machine learning models used in ResaleReckoner should be trained on unbiased and representative data to avoid perpetuating discrimination or unfair pricing practices. Transparency in the pricing methodology and accountability in model performance are critical for maintaining ethical standards.

Based on the findings of the feasibility study, ResaleReckoner appears technically achievable with the right expertise and resources. However, careful consideration of economic, operational, legal, and ethical factors is essential to ensure its successful implementation and long-term sustainability.

2.4 Project Planning

The project will go through the following stages of development in its Software Development Life Cycle.

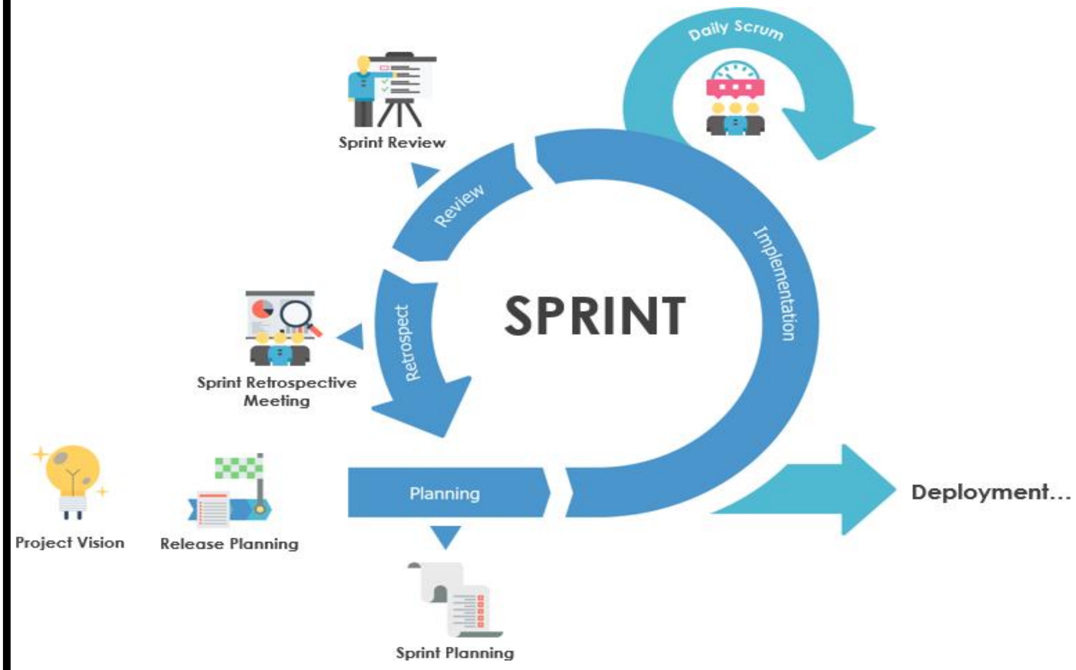


Figure 2.1: Project Planning

2.5 Development Environment

2.5.1 Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

2.5.2 Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google that allows users to write and execute Python code collaboratively. It provides a Jupyter Notebook-like environment where users can write and run Python code, along with Markdown cells for documentation and visualizations. One of the key features of Google Colab is its integration with Google Drive, allowing users to easily access and store their notebooks and datasets. Additionally, Google Colab provides free access to GPU and TPU resources, enabling users to train deep learning models efficiently. Overall, Google Colab offers a convenient and powerful environment for data analysis, machine learning, and AI research without the need for local hardware resources.

2.5.3 Jupyter Notebook

Jupyter Notebook provides an interactive computing environment ideal for data exploration, experimentation, and iterative development.

It allows for the integration of code, visualizations, and explanatory text in a single document, facilitating clear documentation and collaboration.

Jupyter Notebook can be used for tasks such as data preprocessing, exploratory data analysis, model training, and evaluation.

2.5.4 Visual Studio Code (VS Code):

Visual Studio Code is a lightweight yet powerful integrated development environment (IDE) with robust support for Python development.

VS Code offers features such as syntax highlighting, code completion, debugging, and version control integration, enhancing productivity and code quality.

Extensions for Jupyter Notebooks, Python linting, and virtual environments streamline development workflows and ensure code consistency.

VS Code's intuitive user interface and extensive customization options make it a popular choice among developers for various programming tasks.

2.6 REQUIREMENT SPECIFICATION

2.6.1 Software Requirement

The software requirements specification (SRS) is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties–the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development.

Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client’s problem and application area. SRS bridges this communication gap. Problem analysis is done to obtain a clear understanding of the needs of the clients and the users, and what exactly is desired from the software. Analysis leads to the actual specification. People performing the analysis called analysts, are also responsible for specifying the requirements. The software project is initiated by the client's needs .In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify their requirements by talking to these people and understanding their needs. These people and the existing documents about the current mode of operation are the basic source of information for the analyst.

2.6.2 Hardware Requirement

Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This step acquiring all the facts problem specification such as identifying the desired result determining what information is needed to produce these results and figuring out what process must be carried out to proceed to get the accurate result.

Software and Hardware Specification

Software Specification

OPERATING SYSTEM	Microsoft Windows 10 or 11
IDE	Visual Studio Code
Server Side	Python

Hardware Specification

Display	13.5 Inch or Above
Hard Disk	250GB
Processor	Intel i3 or Above
RAM	Minimum 8GB

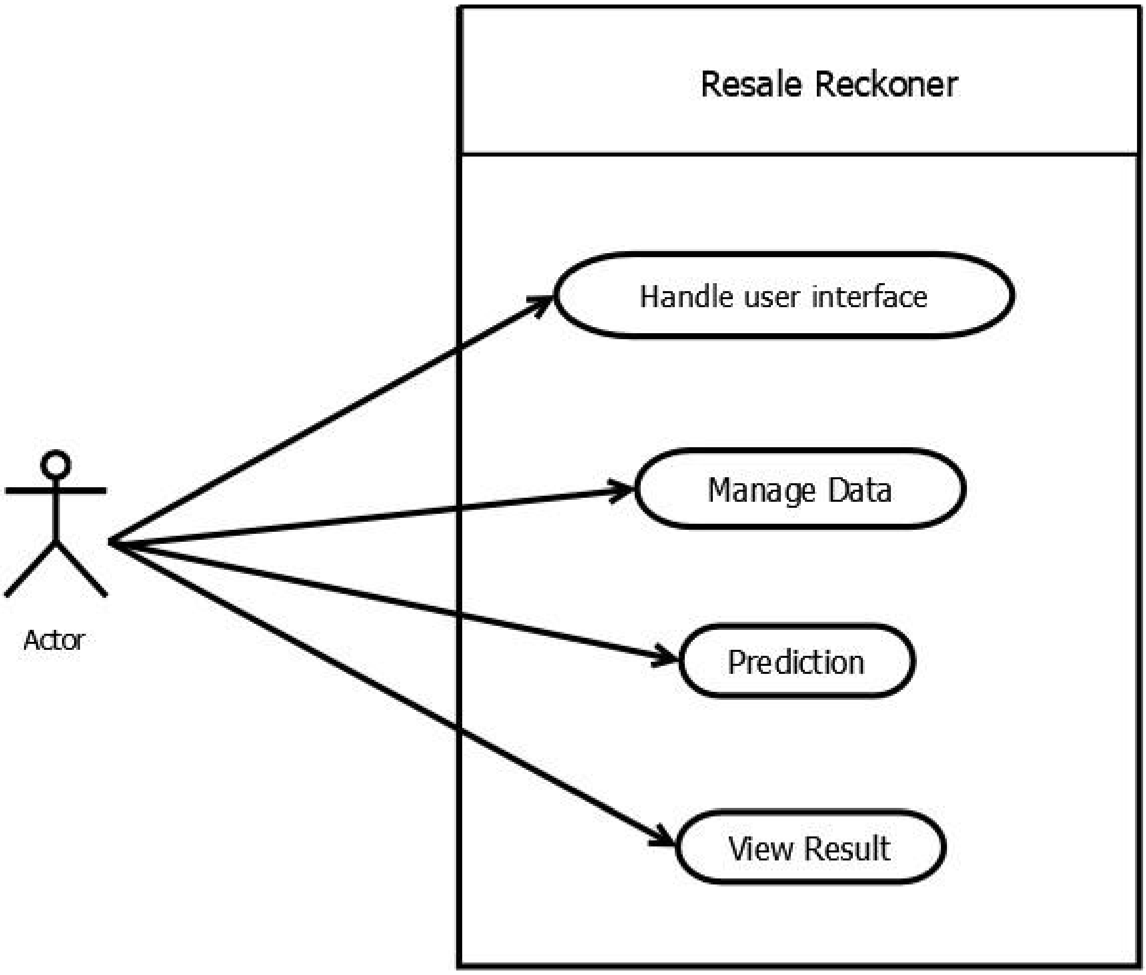
3 .SYSTEM MODELLING

3.1 Design Methodology

Design methodology refers to the development of a system or method for a unique situation. Today the term is most often applied to technological fields in reference to web design, software or information system design. The key design methodology is finding the best solution for each design situation, whether it be in industrial design, architecture or technology. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and wants of the end user is the most critical concern. Design methodology also employs basic research methods, such as analysis and testing.

3.2 Use case diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour.



V

Figure 3.1 Use Case Diagram

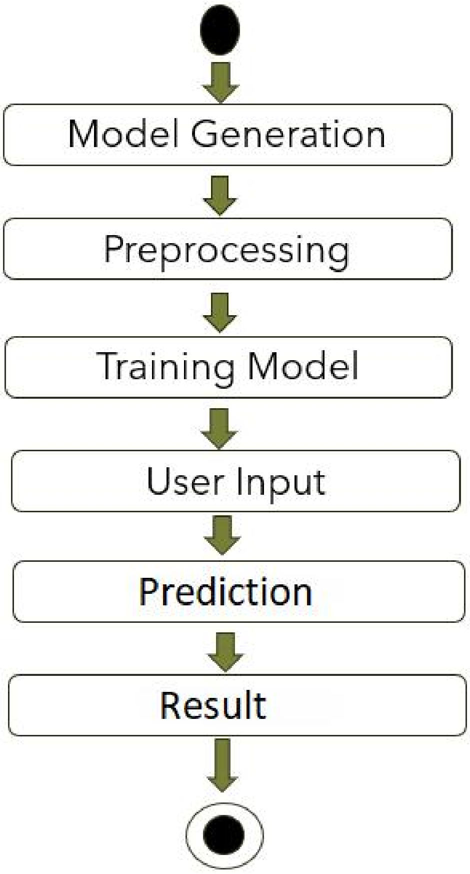


Figure 3.2 Activity Diagram

4. SYSTEM DESIGN

System design is the first step in the development phase for any engineering product or system. The term “Design” is defined as “The process of principles for the purpose of defining a processor a system insufficient detail to permit its realization”. And design is most creative and challenging phase of system development life cycle. It is an approach for the creation of the proposed system is designed, which will help in the system coding. System design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system. A number of sub- systems are to be identified which constitute the whole system.

System design high-end decisions are taken regarding the basic system architecture, platforms and tools to be used. The system design transforms a logical representation of what a given system is required to be in the physical specification. It is an approach for the creation of the proposed system is designed, which will help in the system coding. Design starts with system requirement specification and converts it to a physical reality during the development. Important design fact or such as reliability, response time, throughput of the system maintainability, expand-ability etc. should be taken into account.

System design is the process of developing specifications for the proposed system that meet the needs established in the structured analysis. A major step in the structured design is the preparation of input and output design which will be acceptable to the user. Structured design is the process of planning a new system to replace the old one. Characteristics of well-defined system is:

(1) Acceptability

(2) Decision making ability

(3) Economy

(4) Flexibility

(5) Reliability

(6) Simplicity

5. AGILE TECHNOLOGY OVERVIEW

5.1 Introduction to Scrum

Scrum is a subset of agile. Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum itself is a simple framework for effective team collaboration on complex products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum.

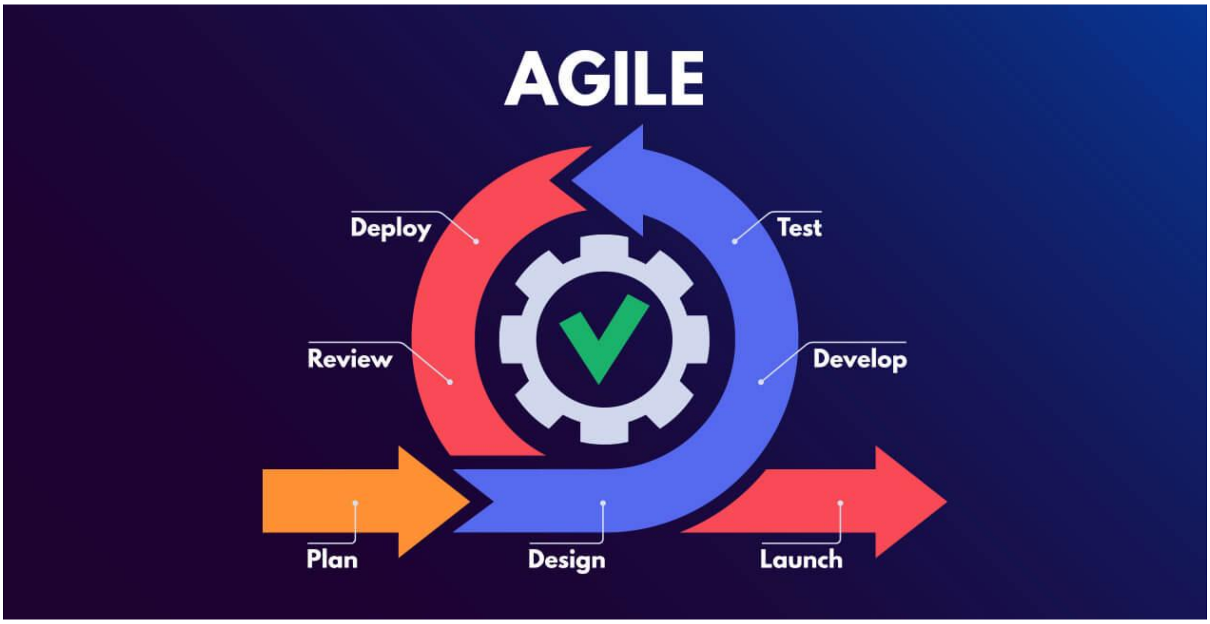


Fig 5.1 Agile Method

5.2 Principles or Methodology Used

The SCRUM methodology is defined by team rules, events(ceremonies),artifacts and roles.

5.2.1 Scrum Team

A scrum team is a collection of individuals (typically between five and nine members) working together to deliver the requested and committed product increments. To work effectively it is important for a scrum team that everyone within the team follow a common goal. The Scrum Team share different tasks and responsibilities related to the delivery of the product. Each role are closely inter-related. It is recommended for Scrum team members work together in the same location whenever possible. There are 3 roles in a scrum team:

1. The Product Owner: The product owner is a project’s key stakeholder-Usually an internal or external customer. or a scope person for the customer. There is only one product owner who conveys the overall mission and vision of the product which the team is building. The product owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

2. The Scrum Master: The scrum master is the servant leader to the product owner, development team and organization. With no hierarchical authority over the team but rather more of a facilitator, The scrum master ensures that the team adheres to scrum theory, practices and rules. The scrum master protect the team by doing anything possible to help the team perform of the highest level. This may include removing impediments, facilitating meetings and helping the product owner groom the backlog.

3. The Development Team: is a self organizing, cross functional group armed with all of the skills to deliver shippable increments at the completion of each sprint.

5.2.2 Scrum Events

Scrum events are time-boxed events that means in a project, every scrum events has a predefined maximum duration. These events enable transparency on the project progress to all who are involved in the projects. The vital events of scrum are -the Sprint.

1. The Sprint: A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprint are usually 2-4 weeks long but can be as short as 1 week .

2. Sprint Planning: Sprint Planning team meetings are time-boxed events that determine which product backlog items will be delivered and have the work will be achieved.

3. The daily Stand-up: The daily stand-up is a short communication meeting in which each team member quickly and transparently covers progress since the last stand-up ,planed work before the next meeting and any impediments that may be blocking his or her progress.

4. The Sprint Review: The sprint review is the ”show and tell” of demonstration events for the team to present the work completed during the sprint. The product owner checks the work against predefined acceptance criteria or either accept or reject the work. The stakeholder or client give feedback to ensure that the delivered incremental must the business model.

5. Retrospective: The retrospective or retro is the final team meeting in the sprint to determine what went well, what didn’t go well and how the team can improve the next sprint. Attended by the team and the scrum master, the retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its process.

5.2.3 Scrum Artifacts

Scrum artifacts are designed to increase transparency of information related to the delivery of the project, and provide opportunities for inspection and adaptation. They are management products useful for the creation of the specialist product of the project. There are 3 artifacts in scrum

1. Product Backlog An extended list of everything that might be needed in the final product.

2. Sprint Backlog Selected items for the product backlog to be delivered through a sprint, along with the task for delivering the item and realizing the sprint goal.

3. The Sprint Review Increment set of all the product backlog items so far in the project.

5.2.4 Scrum Rules

The rules of Agile Scrum Should be completely up to the team and governed for what works best for their processes. The best agile will tell teams to start with basic scrum events listed above and then inspect and adopt based on your teams unique needs so there is continuous improvement in the way teams work together.

5.3 Sprint

5.3.1 Product Backlog

In the simplest definition the Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user centric e.g. in the form of user stories. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the development Team collaborate on the details of Product Backlog items. A Scrum product backlog contains descriptions of the functionality desired in an end product. Agile backlog prioritization is the next step. The Product Backlog is the tool used by the product owner to keep track of all of the features that stakeholders would like to see implemented in the product whereas the Sprint Backlog is a subset of the Product Backlog representing the current active Sprint iteration. However, the product owner prioritizes it.

FEATURE_ID	USER_STORY	PRIORITY	ESTIMATED_TIME (HOURS)
1	As a user I want to access the landing page of the webpage	HIGH	50
2	As a user I want to enter the data required for prediction	HIGH	140
3	As a user I want to get the prediction	HIGH	90

Table 5.1 Product Backlog

5.3.2 Sprint Planner

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box. The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. As the Development Team works, it does so with the Sprint Goal always in mind

F_ ID	SPRINT	START DATE	END DATE	ESTIMATED HOURS	SPRINT GOAL
1.1to 2.2	1	05/02/24	18/12/24	30	Import necessary libraries and create a working environment. Collect suitable dataset. Create user interface

2.3 to 2.4	2	19/02/24	03/03/24	50	Preprocessing dataset to ensure quality. Find suitable ML algorithm.
2.5 to 2.6	3	04/03/24	17/03/24	90	Split the dataset into training and testing dataset. Also create a suitable training environment and train the model.
2.7 to 3.1	4	18/03/24	17/03/23	60	Evaluate the trained model. Display prediction values and check accuracy
3.2 to 3.3	5	01/04/24	14/04/24	50	Enabling users to view the prediction

Table 5.2 Sprint Planner

5.3.3 Ideal Burn Down Chart

A burndown chart is a graphic representation of how quickly the team is working through a customer’s user stories, an agile tool that is used to capture a description of a feature from an end-user perspective. The burndown chart shows the total effort against the amount of work for each iteration. The quantity of work remaining is shown on a vertical axis, while the time that has passed since beginning the project is placed horizontally on the chart, which shows the past and the future. The burndown chart is displayed so everyone on the team can see it and is updated regularly to keep it accurate.

There are two variants that exist for a burndown chart. A sprint burndown is for work remaining in the iteration. When illustrating the work remaining for the entire project, the chart is called a product burndown. The burndown chart has several points. There’s an x-axis, which is the project or iteration timeline. The y-axis is the work that needs to get done in the project. The story point estimates for the work that remains is represented by this axis. The project starting point is the farthest point to the left of the chart and occurs on day zero of the project or iteration. The project end point is farthest to the right and marks the final day of the project or iteration.

There is an ideal work remaining line, which is a straight line connecting the start point to the end point. It shows the sum of the estimates for all the tasks that need to be completed. At the end point, the ideal line crosses the x-axis and shows there is no work left to be done. This line is based on estimates and therefore not always accurate.

Then there is the actual work remaining line that shows the actual work that remains in the project or iteration. At the beginning the actual work remaining and the ideal work remaining are the same, but as the project or iteration progresses the actual work line will fluctuate above and below the ideal work line. Each day a new point is added to this line until the project or iteration is done to make sure it’s as accurate as possible. If the actual work line is above the ideal work line, it means there is more work left than originally thought. In other words, the project is behind schedule. However, if the actual work line is below the ideal work line, there is less work left than had been predicted and the project is ahead of schedule.

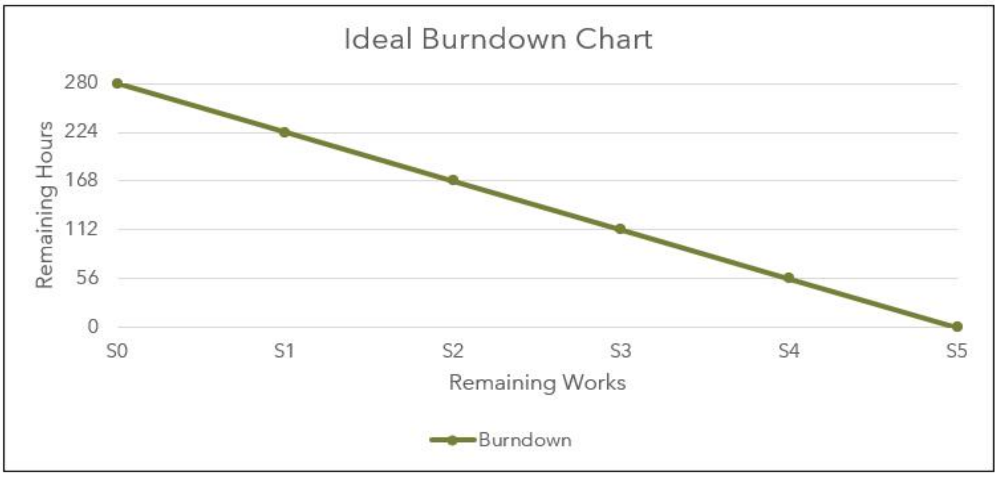


Fig 5.2 Ideal Burndown Chart

5.3.4 Git Hub Registration

GitHub is an online-browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more.

GitHub offers free accounts for users and organizations working on public and open source projects, as well as paid accounts that offer unlimited private repositories and optional user management and security features. Git hub account creation includes the following steps:

-Go to the GitHub sign up page, then Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters. Review carefully the GitHub Terms of Service and Privacy Policy before continuing and Choose a plan. Hereby anyone can finish the account creation procedure.

-You can store a variety of projects in GitHub repositories, including open source projects.

-In the upper-right corner of any page, click , and then click New repository.

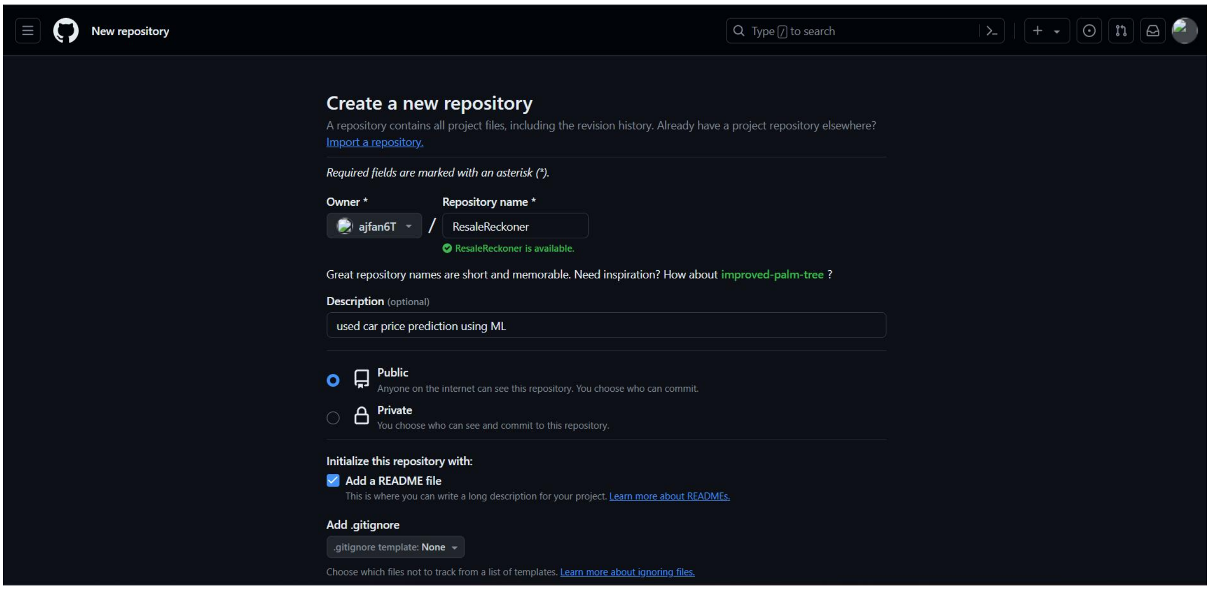
– Type a short, memorable name for your repository followed by Optionally, add a description of your repository, public or private repository.

– Select Initialize this repository with a README. finally Click Create repository

– After creation, need to collaborate members by the admin.

– In the left sidebar, click Collaborators and teams. Under ”Collaborators”, type the name of the person you’d like to give access to the repository, then click Add collaborator.

– Next to the new collaborators name, choose the appropriate permission level: Write, Read, or Admin. – The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * / Repository name *

ajfan6T / ResaleReckoner

ResaleReckoner is available.

Great repository names are short and memorable. Need inspiration? How about improved-palm-tree ?

Description (optional)

used car price prediction using ML

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

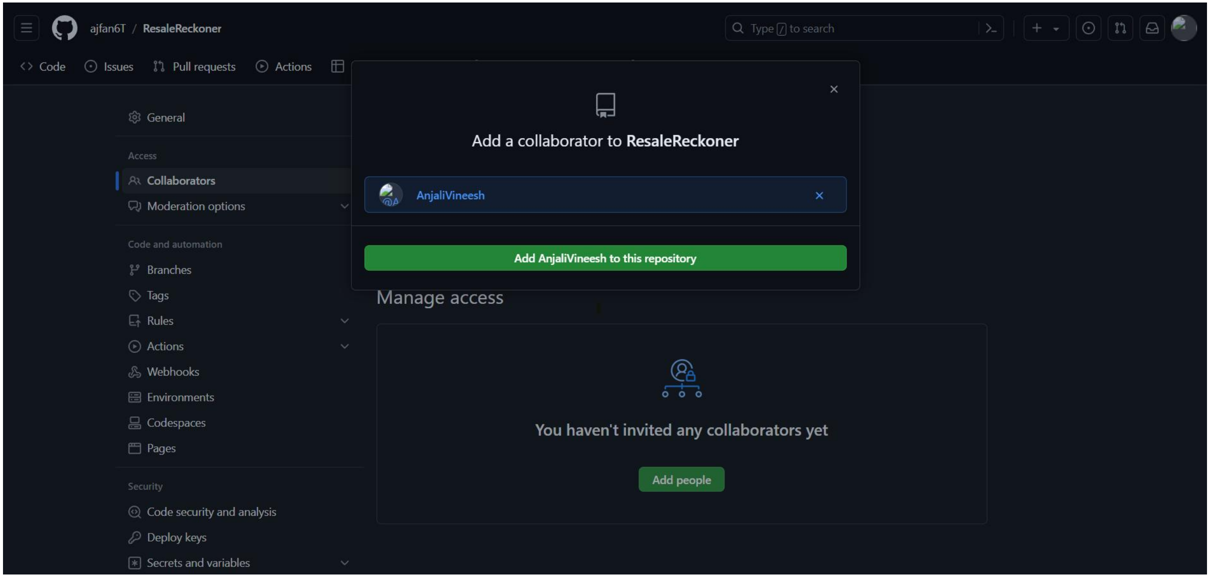
☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Figure 5.3 : Repository Creation



Add a collaborator to ResaleReckoner

AnjaliVineesh

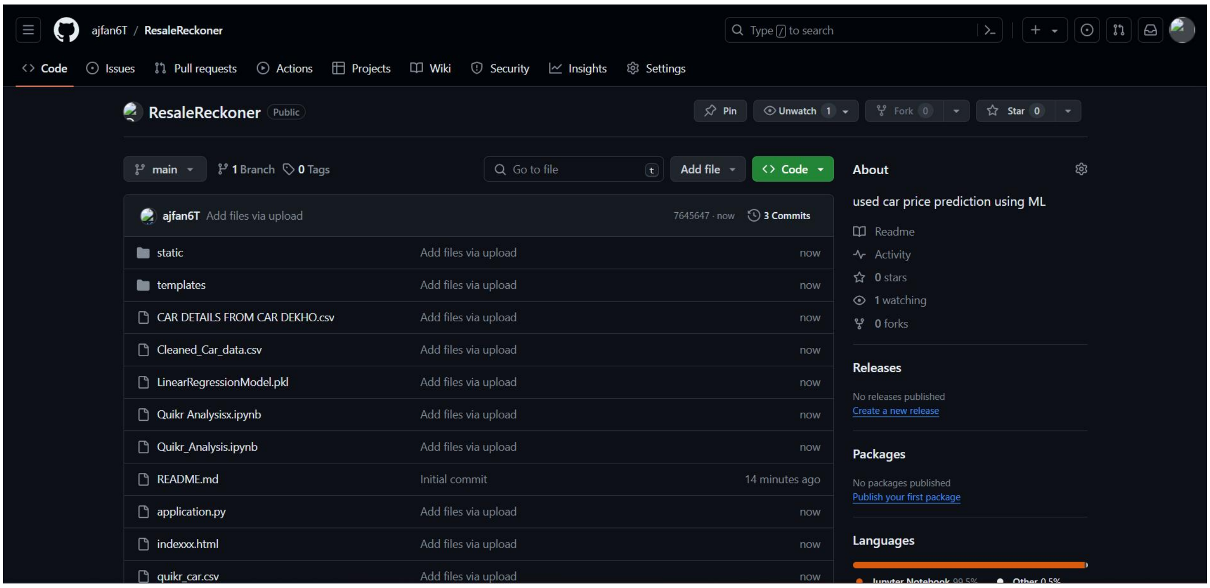
Add AnjaliVineesh to this repository

Manage access

You haven't invited any collaborators yet

Add people

Figure 5.4 : Adding Collaborator



ajfan6T Add files via upload 7645647 · now 3 Commits

static	Add files via upload	now
templates	Add files via upload	now
CAR DETAILS FROM CAR DEKHO.csv	Add files via upload	now
Cleaned_Car_data.csv	Add files via upload	now
LinearRegressionModel.pkl	Add files via upload	now
Quikr Analysisx.ipynb	Add files via upload	now
Quikr_Analysis.ipynb	Add files via upload	now
README.md	Initial commit	14 minutes ago
application.py	Add files via upload	now
indexxx.html	Add files via upload	now
quikr_car.csv	Add files via upload	now

About

used car price prediction using ML

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Jupyter Notebook 99.5% Other 0.5%

Figure 5.5: Project uploaded

6. CODING

The above system is translated into a machine-readable form which is termed as coding. It is basically translating the human readable format to a machine friendly one. The code generation step performs this task:

The following points are considered while converting the system design into coding:

- Are the initializations correct?
- Are the data types properly assigned?
- Is memory leak being dealt with?
- Does it comply with the coding standard?

6.1 Coding Standards Followed

Different modules specified in the design document are coded in the Coding phase according to the module specification. The main goal of the coding phase is to code from the design document prepared after the design phase through a high-level language and then to unit test this code.

Good software development organizations want their programmers to maintain to some well-defined and standard style of coding called coding standards. They usually make their own coding standards and guidelines depending on what suits their organization best and based on the types of software they develop. It is very important for the programmers to maintain the coding standards otherwise the code will be rejected during code review.

Coding standardization basically is the efficiency of our code which has been converted from the system design. The efficiency primarily depends upon:

- Readability: The code should be readable with proper indentation and spacing to make the contents clear of all the modules.
- Portability: The code is portable enough as it will work on various platform given all the necessary dependencies are installed?
- Debug Easily: The coding should be error-free as much as possible

7. CODE REVIEW AND TESTING

7.1 Code Review

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to Do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases.

Testing Objectives include:

– Testing is a process of executing a program with the intent offending an error.

– A good test case is one that has a probability of finding an as yet undiscovered error.

– A successful test is one that uncovers an undiscovered error.

Testing Principles are:

– All tests should be traceable to end user requirements

– Tests should be planned long before testing begin

– Testing should begin on a small scale and progress towards testing in large

– Exhaustive testing is not possible

– To be most effective testing should be conducted by an independent third party.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can cause chaos and confusion. The implementation stage involves the following tasks :

– Careful planning.

– Investigation of system and constrains. of methods to achieve the changeover.

– Training of staff in the changeover phase. – Evaluation of the changeover method.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

7.2 Testing Process

Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program. Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing present interesting anomaly for the software engineer.

Unit testing

This is the first of testing. In this different modules are tested against the specification produces during the design of the modules. It refers to the verification of single program module in an isolated environment. Unit testing focuses on the modules independently of one another to locate errors.

In our project we test each module and each forms individually. Each forms may tested using appropriate values. The input screens need to be designed very carefully and logically. While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered.

Validation Checks

As a web application developer, form validation is a crucial part of your work, and it should not be underrated as it could lead to security flaws in your application. You should consider it a must if you’re striving to provide a professional end user experience.

– Basic form validation

– Custom error messages

I always prefer to load common libraries and helpers in the constructor of the controller itself as it’s a nice habit to avoid code duplication elsewhere in controller methods. We load the form and url helpers so that we can use the utility methods provided by those helpers throughout the rest of the application. The form validation library in the constructor, you can access it using the form validation convention.

8. IMPLEMENTATION

The implementation of ResaleReckoner involves several key steps, including data preprocessing, model training, web development, and deployment. Here's a high-level overview of the implementation process:

1. **Data Preprocessing:**

- Import the dataset containing information about various vehicles, including their model, year, fuel type, kilometers driven, and selling price.
- Perform data cleaning to handle missing values, outliers, and inconsistencies in the dataset.
- Conduct feature engineering to extract relevant features and create new ones that may

enhance the predictive power of the model.

- Split the dataset into training and testing sets to evaluate model performance.

2. **Model Training:**

- Select appropriate machine learning algorithms for price prediction, such as regression algorithms (e.g., linear regression, random forest regression, gradient boosting regression).
- Train multiple models using the training dataset and evaluate their performance using appropriate evaluation metrics (e.g., mean absolute error, root mean squared error).
- Fine-tune hyperparameters of the selected models to optimize performance and generalization ability.
- Select the best-performing model based on evaluation metrics and save it for deployment.

3. **Web Development:**

- Choose a web development framework such as Flask or Django for building the web application.
- Design and develop the user interface (UI) of ResaleReckoner, including input forms for users to provide details about their cars and display areas for predicted prices.
- Integrate the trained machine learning model into the web application to enable real-time price prediction based on user inputs.
- Implement filtering options and user-friendly features to enhance the user experience and facilitate ease of use.

4. **Deployment:**

- Choose a deployment platform such as Heroku, AWS, or Microsoft Azure for hosting the web application.
- Set up the deployment environment and configure dependencies required for running the application.
- Deploy the web application to the chosen platform, ensuring that it is accessible to users via a web browser.
- Conduct thorough testing of the deployed application to verify its functionality, performance, and scalability in a production environment.

5. **Monitoring and Maintenance**:

- Implement monitoring tools to track application performance, user interactions, and any potential issues or errors.
- Regularly update the machine learning model with new data to improve its accuracy and

adapt to changing market trends.

- Address user feedback and feature requests to enhance the functionality and usability of the application over time.
- Conduct periodic maintenance and optimization tasks to ensure the smooth operation and reliability of ResaleReckoner.

By following these implementation steps, developers can create a robust and user-friendly web application that accurately predicts the resale value of used cars, empowering users to make informed decisions in the automotive market.

9. CONCLUSION

In summary, ResaleReckoner marks a significant step forward in the realm of used car price prediction, providing users with a dependable and user-friendly platform for estimating the resale value of their vehicles. Through the integration of cutting-edge machine learning algorithms, comprehensive data analysis, and intuitive web development, ResaleReckoner offers a seamless solution to the complexities of the automotive resale market. By empowering users with transparent pricing insights, real-time predictions, and tailored filtering options, ResaleReckoner facilitates informed decision-making, enhances market efficiency, and fosters trust among buyers and sellers alike.

Looking ahead, the ongoing evolution of ResaleReckoner will depend on continuous monitoring, refinement, and adaptation to meet the changing needs of users and the dynamic automotive landscape. By prioritizing user feedback, incorporating new data sources, and embracing emerging technologies, ResaleReckoner can continue to serve as a valuable resource for individuals navigating the complexities of buying and selling used cars, ultimately driving greater transparency, efficiency, and satisfaction in the automotive resale market.

10.FUTURE ENHANCEMENT

Mobile Application Development:

Develop a mobile application version of ResaleReckoner for both iOS and Android platforms. Optimize the user interface and experience for mobile devices, ensuring seamless navigation and usability on smaller screens.

Leverage platform-specific features and capabilities to enhance user engagement, such as push notifications, location services, and camera integration for VIN scanning or image recognition.

Additional Features for the Mobile App:

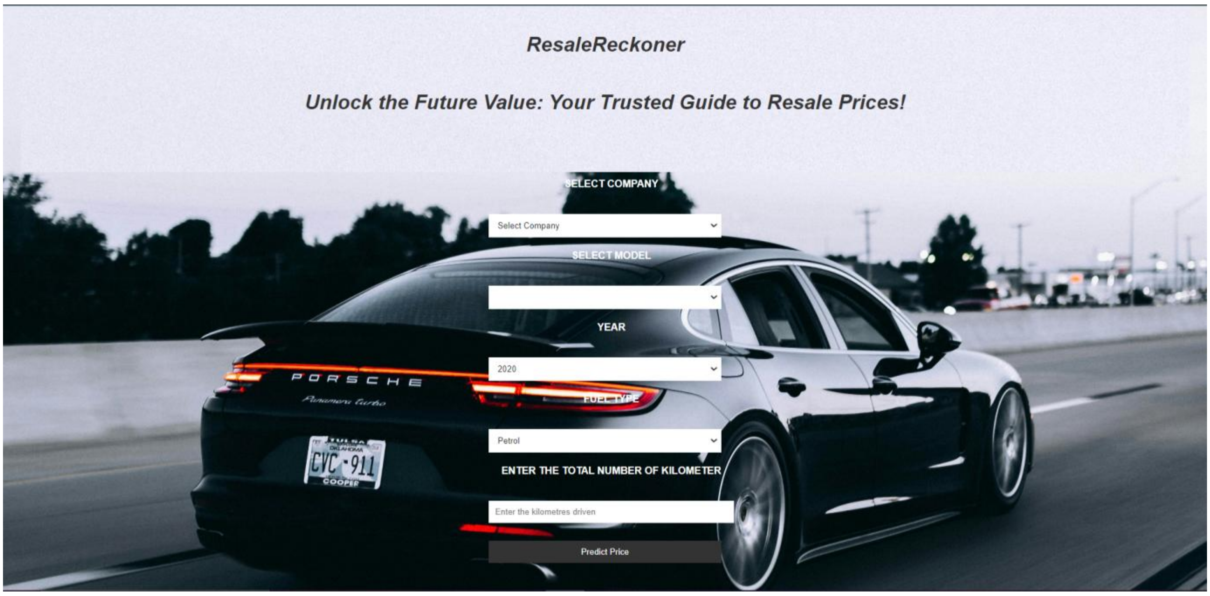
VIN Scanner: Implement a VIN (Vehicle Identification Number) scanner feature that allows users to scan the VIN barcode of their vehicle using their mobile device's camera. This feature automatically fetches relevant vehicle information, such as model, year, and trim, to streamline the input process for users.

Augmented Reality (AR) Visualization: Introduce an AR visualization feature that enables users to see estimated resale values overlaid onto their physical surroundings. This feature can provide users with an immersive and interactive way to visualize the potential value of their vehicle in different settings, such as their driveway or a dealership lot.

Price Alerts: Enable users to set price alerts for specific vehicles or market conditions of interest. Users can receive notifications on their mobile devices when the estimated resale value of their selected vehicles meets certain criteria, allowing them to stay informed about changes in the market and make timely decisions.

11. APPENDIX

11.1 SCREENSHOTS



ResaleReckoner

Unlock the Future Value: Your Trusted Guide to Resale Prices!

SELECT COMPANY: Select Company

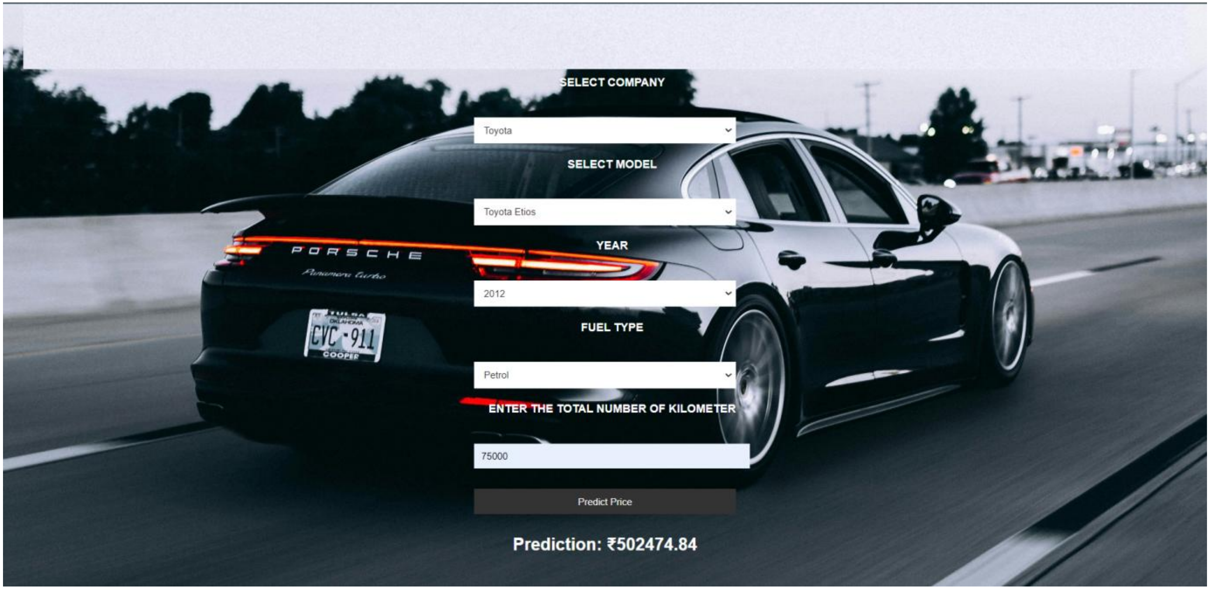
SELECT MODEL:

YEAR: 2020

FUEL TYPE: Petrol

ENTER THE TOTAL NUMBER OF KILOMETER: Enter the kilometres driven

Fig 10.1.1 Landing Page



SELECT COMPANY

Toyota

SELECT MODEL

Toyota Etios

YEAR

2012

FUEL TYPE

Petrol

ENTER THE TOTAL NUMBER OF KILOMETER

75000

Prediction: ₹502474.84

Fig 10.1.2 Prediction

11.2 SAMPLE CODE

11.2.1 application.py

```
from flask import Flask,render_template,request,redirect
from flask_cors import CORS,cross_origin
import pickle
import pandas as pd
import numpy as np
app=Flask(__name__)
cors=CORS(app)
model=pickle.load(open('LinearRegressionModel.pkl','rb'))
car=pd.read_csv('Cleaned_Car_data.csv')
@app.route('/',methods=['GET','POST'])
def index():
    companies=sorted(car['company'].unique())
    car_models=sorted(car['name'].unique())
    year=sorted(car['year'].unique(),reverse=True)
    fuel_type=car['fuel_type'].unique()
    companies.insert(0,'Select Company')
    return render_template('index.html',companies=companies, car_models=car_models,
years=year,fuel_types=fuel_type)
@app.route('/predict',methods=['POST'])
@cross_origin()
def predict():
   company=request.form.get('company')
    car_model=request.form.get('car_models')
    year=request.form.get('year')
    fuel_type=request.form.get('fuel_type')
    driven=request.form.get('kilo_driven')
    prediction=model.predict(pd.DataFrame(columns=['name', 'company', 'year', 'kms_driven',
'fuel_type'],
   data=np.array([car_model,company,year,driven,fuel_type]).reshape(1, 5)))
    print(prediction)
    return str(np.round(prediction[0],2))
if __name__=='__main__':
    app.run()
```

11.2.2 car.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
mpl.style.use('ggplot')
carq=pd.read_csv('D:/neww/CAR DETAILS FROM CAR DEKHO.csv')
carq
carq['company'] = carq['name'].str.split().str[0]
company_column = carq.pop('company')
# Insert the 'company' column after the 'name' column
carq.insert(carq.columns.get_loc("name") + 1, "company", company_column)
carq.drop(['seller_type', 'owner'], axis=1, inplace=True)
carq
carq['namen'] = carq['name'].apply(lambda x: ' '.join(x.split()[:2]))
carq = carq.drop(columns=['name'])
carq = carq.rename(columns={'namen': 'name'})
carq = carq[['name'] + [col for col in carq if col != 'name']]
carq = carq.rename(columns={'selling_price': 'Price'})
carq = carq.rename(columns={'km_driven': 'kms_driven'})
carq = carq.rename(columns={'fuel': 'fuel_type'})
carq
carq.shape
carq.info()
##### Creating backup copy
backup=carq.copy()
## Cleaning Data
carq.shape
#### Company does not need any cleaning now. Changing car names. Keeping only the first three words
carq['name']=carq['name'].str.split().str.slice(start=0,stop=3).str.join(' ')
#### Resetting the index of the final cleaned data
carq=carq.reset_index(drop=True)
## Cleaned Data
carq
carq.to_csv('Cleaned_Car_data.csv')
carq.info()
carq.describe(include='all')
#car=car[car['Price']<6000000]
### Checking relationship of Company with Price
carq['company'].unique()
import seaborn as sns
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='company',y='Price',data=carq)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
### Checking relationship of Year with Price
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=carq)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

```
### Checking relationship of kms_driven with Price
sns.relplot(x='kms_driven',y='Price',data=carq,height=7,aspect=1.5)
### Checking relationship of Fuel Type with Price
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=carq)
### Relationship of Price with FuelType, Year and Company mixed
ax=sns.relplot(x='company',y='Price',data=carq,hue='fuel_type',size='year',height=7,aspect=2)
ax.set_xticklabels(rotation=40,ha='right')
### Extracting Training Data
X=carq[['name','company','year','kms_driven','fuel_type']]
y=carq['Price']
X
y
y.shape
### Applying Train Test Split
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
#### Creating an OneHotEncoder object to contain all the possible categories
ohe=OneHotEncoder()
ohe.fit(X[['name','company','fuel_type']])
#### Creating a column transformer to transform categorical columns
column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['na
me','company','fuel_type']),
 remainder='passthrough')
#### Linear Regression Model
lr=LinearRegression()
#### Making a pipeline
pipe=make_pipeline(column_trans,lr)
#### Fitting the  model
```

```
pipe.fit(X_train,y_train)
y_pred=pipe.predict(X_test)
#### Checking R2 Score
r2_score(y_test,y_pred)
#### Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
np.argmax(scores)
scores[np.argmax(scores)]
carq
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Wagon','Maruti',2007,50000,'Petrol']).reshape(1,5)))
#### The best model is found at a certain random state
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.argmax(scores))
lr=LinearRegression()
pipe=make_pipeline(column_trans,lr)
pipe.fit(X_train,y_train)
y_pred=pipe.predict(X_test)
r2_score(y_test,y_pred)
import pickle
pickle.dump(pipe,open('LinearRegressionModel.pkl','wb'))
pipe.predict(pd.DataFrame(columns=['name','company','year','kms_driven','fuel_type'],data=np.array(['Maruti Wagon','Maruti',2007,50000,'Petrol']).reshape(1,5)))
pipe.steps[0][1].transformers[0][1].categories[0]
```

12. REFERENCES

- Raschka, S., & Mirjalili, V. (2019). "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2." Packt Publishing.
- Brownlee, J. (2020). "Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End." Machine Learning Mastery.
- Kaggle (https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho): Kaggle is a leading platform for data science and machine learning, offering datasets, competitions, and discussions. It's a valuable resource for accessing datasets related to used car sales and market insights.
- Towards Data Science (https://towardsdatascience.com/): Towards Data Science is an online publication featuring articles and tutorials on data science and machine learning. It provides insights into data preprocessing, model selection, and deployment best practices.