LOVELY PROFESSIONAL UNIVERSITY

Price Prediction of Used Cars

SRS

(Software Requirements Specification)

Developed By:

Vivek Kumar Gupta (12215379)

Sachin Thakur (12215076)

Intiyaz Ali (12215366)

Submitted To:

Dr. Tarandeep Walia

Table of Contents

1. Executive Summary

1.1 PROJECT OVERVIEW

The "Used Car Price Prediction" project is a data-driven initiative designed to provide an accurate and reliable estimation of the market value of pre-owned automobiles. Leveraging the power of machine learning and a rich dataset of historical used car sales, this project aims to assist both car sellers and buyers in making informed decisions when it comes to pricing and transacting used vehicles.

Approximately 40 million used vehicles are sold each year. Effective pricing strategies can help any company to efficiently sell its products in a competitive market and making profit. In the automotive sector, pricing analytics play an essential role for both companies and individuals to assess the market price of a vehicle before putting it on sale or buying it. And, the rise of used cars sales is exponentially increasing. Car sellers sometimes take advantage of this scenario by listing unrealistic prices owing to the demand.

Therefore, arises a need for a model that can assign a price for a vehicle by evaluating its features taking the prices of other cars into consideration. In this Notebook, we use supervised learning methods to predict the prices of used cars. The model has been chosen after careful exploratory data analysis to determine the impact of each feature on price.

Objectives:

- **Accurate Valuation:** Develop a robust machine learning model that can accurately predict the market value of used cars based on a range of factors, including make, model, year, mileage, condition, and more.
- **User-Friendly Interface:** Create an intuitive and user-friendly interface that enables users to effortlessly input the details of a vehicle and receive instant and precise price predictions.
- **Data Integrity:** Establish a comprehensive and up-to-date database of used car listings, ensuring the predictions are based on real-market data.
- **Transparency:** Provide users with a clear and transparent breakdown of the factors that influence the price prediction. This transparency helps users understand why a specific valuation was generated.
- **Comparison Tools:** Allow users to compare the estimated price with similar listings, enabling them to assess the competitiveness of their listing or the value of a potential purchase.
- **Historical Data:** Offer access to historical price trends, helping users understand how a car's value may change over time.
- **Customization:** Enable users to fine-tune predictions by providing additional details, such as optional features, vehicle history, or service records for more precise valuations.

1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION

Purpose :

The purpose of used car price prediction is to estimate the fair market value of a used car based on a set of features, such as the make, model, year, mileage, condition, and location. This information can be used by buyers and sellers to make informed decisions about the price of a used car.

In Python, there are a number of libraries and tools that can be used to build and train machine learning models for used car price prediction. One popular library is scikit-learn, which provides a variety of regression algorithms that can be used to predict continuous values, such as the price of a car.

To build a used car price prediction model in Python, the first step is to collect a dataset of used car sales.

Scope:

The scope of this specification defines the boundaries and parameters within which the project will operate. It outlines the specific requirements and constraints that the development team must consider during the project's lifecycle. Here are key elements of the scope:

Functional Requirements: This specification will detail the specific functions and features that the Expense Tracker System must have. These include expense recording, categorization, analysis, reporting, and any other relevant functionalities.

User Requirements: The system should be designed with the end-users in mind. User requirements will be addressed to ensure that the system is user-friendly, accessible, and efficient.

Technical Requirements: The specification will highlight the technical aspects that must be considered, such as compatibility with various devices, platforms, and databases, as well as security protocols.

Performance Requirements: This section will outline the expected performance benchmarks, such as response times, system reliability, and scalability.

Constraints: Constraints may include budget limitations, time constraints, and resource limitations that the project team needs to work within.

Compliance and Standards: If applicable, the specification may reference industry standards, regulations, or compliance requirements that the system must adhere to.

Integration Requirements: If the system needs to interact with other software or systems, these integration requirements will be detailed.

Testing and Quality Assurance: The specification will also describe the testing procedures and quality assurance measures that will be in place to ensure the system meets its intended functionality and performance levels.

2. PRODUCT / SERVICE DESCRIPTION

Used car price prediction is a service that uses machine learning to estimate the fair market value of a used car. It is a valuable tool for both buyers and sellers, as it can help buyers to avoid overpaying and sellers to get the best possible price for their car.
Used car price prediction services work by taking into account a variety of factors, including the make, model, year, mileage, condition, location, features, and options of the vehicle. They also consider historical data on used car sales to train their models.

Benefits to using a used car price prediction service:

- Accuracy: Used car price prediction services are trained on large datasets of historical used car sales data, so they can provide a very accurate estimate of the fair market value of a vehicle.
- Convenience: Used car price prediction services are typically very easy to use. Simply input the relevant information about the vehicle and the service will provide you with an estimated price.
- Transparency: Used car price prediction services can help to create a more transparent market for used cars. By providing buyers and sellers with an accurate estimate of the fair market value of a vehicle, these services can help to reduce the amount of haggling and negotiation that is involved in buying and selling used cars.

Used car price prediction services can benefit a wide range of people, including:

- **Car buyers:** Used car buyers can use these services to avoid overpaying for a vehicle. By having an accurate estimate of the fair market value of a car, buyers can be better informed when negotiating a price with the seller.
- **Car sellers**: Used car sellers can use these services to get the best possible price for their car. By knowing the fair market value of their car, sellers can set a realistic price and avoid underselling themselves.
- **Car dealerships:** Car dealerships can use used car price prediction services to value their inventory and set prices for used cars that are fair and competitive.
- **Lenders:** Lenders can use used car price prediction services to assess the value of collateral for loans.

2.1 USER CHARACTERISTICS

User characteristics are the attributes that describe a user of a product or service. They can include demographic factors such as age, gender, location, and income, as well as behavioural factors such as usage patterns, preferences, and motivations.

User characteristics are important to understand because they can be used to design products and services that are more user-friendly and meet the needs of the target audience. For example, if a company knows that its users are mostly young adults, it may design its website and marketing materials with a more youthful tone and style.

User characteristics can also be used to segment users into different groups, which can be useful for targeted marketing and personalization. For example, a company may segment its users by age group and send different marketing messages to each group.

User characteristics that may be relevant to a used car price prediction project:

- **Demographics:**
  - Age
  - Gender
  - Location (country, state, city)
  - Income
  - Education level
- **Car buying history:**
  - Number of used cars purchased in the past
  - Make and model of previous used cars
  - Average price paid for previous used cars
- **Car preferences:**
  - Make and model of preferred used cars
  - Year and mileage preferences
  - Condition preferences (new, used, like-new, certified pre-owned)
  - Features and options preferences
- **Budget:**
  - Maximum and minimum price willing to pay for a used car
- **Motivation:**
  - Reason for buying a used car (e.g., to save money, to get a reliable car, to get a car with certain features or options)

Understanding the user characteristics that are relevant to your used car price prediction project can help you to develop a more accurate and useful model. For example, if you know that your target users are young adults with a limited budget, you can focus on including features and options that are important to this demographic group.

2.2 ASSUMPTIONS

Assumptions are statements that are believed to be true, but have not been proven. They are important to identify and document because they can have a significant impact on the success of a project.

Technical Assumptions:

1. **Python and Libraries:** It is assumed that the project will be developed using the Python programming language, and necessary libraries such as NumPy, Pandas, Scikit-Learn, and Matplotlib will be available for data analysis and machine learning.
2. **Data Source:** The project assumes access to a reliable data source, which may include a dataset of historical used car listings with attributes like make, model, year, mileage, condition, and price.
3. **Data Pre-processing:** It is assumed that data pre-processing tasks, such as cleaning, encoding categorical features, and handling missing values, will be performed before model training.

2.3 CONSTRAINTS

Constraints are limitations or restrictions that must be considered during a project. They can be external factors, such as budget or deadlines, or internal factors, such as the availability of resources or the capabilities of the team.

- **Time:** Developing and deploying a used car price prediction model can be a time-consuming process. If the project needs a model quickly, it may need to make sacrifices in terms of accuracy or complexity.

- **Budget:** The project will have a budget that will constrain the types of data that can be purchased, the computational resources that can be used, and the level of expertise that can be hired.

- **User-Related Constraints:**

  User Behaviour: User behaviour, such as inaccurate data input or low user adoption, can be a constraint that affects the system's performance and predictions.

2.4 DEPENDENCIES

Dependencies are relationships between tasks or activities. They occur when one task cannot be started or completed until another task is completed.

- **Hardware:**
  - A computer with sufficient computational resources to train and evaluate the model.
- **Software:**
  - Python programming language
  - NumPy library
  - Pandas library
  - Scikit-learn library
  - Matplotlib
- **Data:**
  - A large and representative dataset of used car sales data.

Additional dependency:

- A web development framework: If you are developing a web-based used car price prediction application, you will need to use a web development framework such as Django or Flask.

3. REQUIREMENT

3.1 FUNCTIONAL REQUIREMENT

Data Collection and Input:

The system must allow users to input details of a used car, including make, model, year, mileage, condition, and additional features.

Users must have the option to provide historical maintenance and service records for more accurate price predictions.

The system must validate and sanitize user inputs to ensure data accuracy.

Data Pre-processing:

The system must preprocess user inputs, including encoding categorical features and handling missing values.

Data cleaning procedures should be implemented to remove outliers and ensure data quality.

Machine Learning Model:

The system must employ a machine learning model to predict used car prices.

The model must be capable of handling regression tasks and be trained using historical used car data.

The system must periodically retrain the model to adapt to changing market conditions.

The system must use appropriate machine learning algorithms for regression tasks, such as linear regression, decision trees, or ensemble methods.

Price Prediction:

The system should provide users with an estimated fair market price for the entered used car details.

The price prediction must be based on the machine learning model's output, factoring in all provided details.

User Interface:

The system should have a user-friendly web-based interface for users to input car details and view price predictions.

Users should receive clear and transparent explanations of how the price prediction was generated.

Non-Functional Requirements:

Non-functional requirements are specifications that describe the qualities or characteristics of the system that are not directly related to its primary functionality but are crucial for its overall performance, usability, and reliability.

Hardware Requirements:

- Processor: Intel Dual-Core or above
- RAM: 4 GB or above

1. **Processor: Intel Dual-Core or above:** This requirement specifies the minimum hardware processing power necessary to run the Expense Tracker System efficiently. It suggests that the system should be compatible with processors that are at least dual core from the Intel brand or any equivalent processor from other manufacturers.

2. **RAM: 4 GB or above:** This requirement sets a minimum RAM (Random Access Memory) capacity that the user's device should have to operate the Expense Tracker System effectively. It specifies that the system should be able to run on devices with at least 4 gigabytes of RAM.

Software Requirements:

1. **Operating System: Windows, macOS, or Linux:** This requirement specifies the compatible operating systems on which the product/service can be installed and run. It includes three widely used operating systems: Windows, macOS (Apple's operating system), and Linux (a popular open-source OS).

2. **Web Browsers: Chrome, Firefox, Safari:** This requirement outlines the web browsers that the system is optimized for. It includes Google Chrome, Mozilla Firefox, and Apple Safari, which are well-known and widely used web browsers.

3.2 USER INTERFACE REQUIREMENT

- The UI should be easy to use and navigate. Users should be able to easily input the relevant information about their used car and receive a prediction of the price.
- The UI should be informative. Users should be able to see the different features that were used to make the prediction and the weight that each feature was given.
- The UI should be customizable. Users should be able to choose which features they want to include in the prediction and the level of detail that they want to see in the results.
- We can use web frameworks like Flask or Django for web applications or GUI libraries like Tkinter for desktop applications.

3.3 USABILITY

Gather data: Need a dataset of car prices and features. This data can be collected from a variety of sources, such as online car listings or government databases.

Prepare the data: Once you have gathered your data, you will need to clean it and prepare it for modeling. This may involve removing outliers, converting categorical variables to numerical variables, and scaling the data.

Use the model to make predictions: Once you are satisfied with the model's performance, you can use it to make predictions for new cars.

3.4 PERFOMANCE

3.4.1 Scalability

If application becomes popular and attracts a large number of users, you may need to consider scalability. Cloud-based solutions or distributed computing may be necessary to handle increased demand.

3.4.2 Availability

Data: Need a dataset containing information about used cars, such as make, model, year, mileage, and price. The availability of this data depends on your sources, such as scraping websites, purchasing data, or obtaining it from open datasets.

Python: Ensure to have Python installed on your system. Python is widely available and can be downloaded from the official website (python.org) or as part of data science distributions like Anaconda.

Python Libraries:
Pandas: Use Pandas for data manipulation and pre-processing.
Matplotlib and Seaborn: These libraries can help with data visualization.
NumPy: Useful for numerical operations and working with arrays.
Jupyter Notebook: An interactive environment for running code and documenting your work.

3.4.3 Latency

Objective :

Latency requirements address the system's responsiveness, with a focus on timely data retrieval and seamless user interactions. Requirement
1. The system shall deliver responsive user interactions, targeting low latency for generating visualizations, processing user queries, and data retrieval tasks.

2. Data retrieval from external sources, including Kaggle, must be optimized to minimize data loading time. Specifically, the time required to fetch and import new data should not exceed 10 seconds for datasets up to 5MB in size.

3. User interactions with the user interface, encompassing actions such as button clicks and form submissions, must maintain a latency of no more than 1 second for most operations.

4. The system shall incorporate robust latency monitoring tools to continuously track response times. Automated alerts must be triggered if the system's response time surpasses acceptable thresholds, promptly notifying administrators for immediate investigation and resolution.

3.5 MANAGEABILITY / MAINTAINABILITY

3.5.1 Monitoring

Objective :

Monitoring requirements focus on the system's ability to track its own performance, detect issues, and provide data for analysis

Train a car price prediction model: You can use the steps described in my previous response to train a car price prediction model.

Deploy the model: Once you have trained a model, you need to deploy it so that you can use it to make predictions on new data. There are a number of ways to deploy a Python model, such as using a cloud computing platform or a local web server.

Collect new data: You need to collect new data about the used car that you want to monitor the price of. This data should include the same features that you used to train your model.

Make a prediction: Use the deployed model to make a prediction of the used car's price.

3.5.2 Maintainance

- **Document the code:**

Write clear and concise documentation for all of the code in the system. This will help you and other developers to understand how the system works.

- **Use version control:**

  Use a version control system, such as Git, to track changes to the code. This will make it easy to revert to a previous version of the code if necessary.

- **Test the code:**

  Write comprehensive unit tests and integration tests for all of the code in the system. This will help you to identify and fix bugs early on.

3.5.3 Operation

Operations requirements define the processes and procedures for the day-to-day management and execution of the system.

Requirement

1. A detailed operations manual shall be created, providing clear instructions for system administrators and operators regarding system operation, routine tasks, and troubleshooting procedures.

2. An incident response plan shall be established, outlining steps to be taken in case of system outages, data breaches, or other critical incidents. This plan should detail responsibilities, escalation procedures, and communication protocols.

3. Backup and disaster recovery plans shall be implemented to ensure data integrity and system availability in case of data loss or unforeseen disasters.

4. Routine system health checks shall be conducted to verify system integrity and performance, with identified issues addressed promptly.

3.6 SYSTEM INTERFACE INTEGRATION

1 External System Interfaces

Objective
This section describes how the system interfaces with external systems, tools, or services.

Requirement

1. The system shall integrate with external data sources such as Kaggle for fetching used car price data. The integration should adhere to Kaggle's API standards and guidelines.

2. The system shall provide an interface for users to interact with external visualization tools, such as Plotly and Matplotlib, to generate charts, graphs, and reports based on the analyzed data.

3. Authentication and authorization mechanisms shall be in place to ensure secure access to external systems and data sources

2. Internal System Interfaces

Objective

This section addresses how different system components and modules interact internally.

Requirement

1. The system shall employ a modular architecture, with well-defined interfaces between system components for ease of maintenance and future enhancements.

2. Data communication between system modules shall follow standard data formats and protocols to ensure compatibility and consistency.

Integration with User Interfaces

Objective

This section specifies how the system interfaces with the user interfaces.

Requirement

1. The system shall provide a user-friendly web-based interface for users to access and interact with the application. The interface should be accessible via standard web browsers.

2. The user interface shall be designed to accommodate various user devices, including desktop computers, tablets, and mobile phones, ensuring a responsive and consistent user experience.

3. The system shall offer options for customizing user interface elements and reports to cater to user preferences and needs.

Integration with Data Sources

Objective

This section outlines how the system interacts with various data sources.

Requirement

1. The system shall support data extraction from multiple sources, including structured and unstructured data, databases, and CSV files. It shall provide mechanisms for data transformation and cleaning.

2. Data integration shall include error handling and validation to ensure data consistency and accuracy.

- Input interface: The system must be able to accept input from users, such as the make, model, year, mileage, condition, location, features, and options of the vehicle. This input can be accepted through a variety of interfaces, such as a web form, a mobile app, or a command-line interface.
- Output interface: The system must be able to provide output to users, such as a prediction of the price of the vehicle and a breakdown of the factors that influenced the prediction. This output can be provided through a variety of interfaces, such as a web page, a mobile app, or a text file.

3.7 Security

3.7.1 Protection

Objective

This section outlines the security measures in place to protect the system and its data from unauthorized access, breaches, and data integrity threats.

- **Use a secure development environment:** When developing and testing your model, use a secure development environment. This includes using strong passwords, keeping your software up to date, and using a firewall.

- Do not share your model with unauthorized users.
- Use a web application firewall (WAF).
- Keep your software up to date.

Requirement

1. The system shall employ encryption mechanisms (e.g., SSL/TLS) for securing data transmission between users' devices and the server.

2. Data at rest shall be encrypted using industry-standard encryption algorithms to protect sensitive information in the database.

3. Access control lists (ACLs) shall be implemented to restrict access to sensitive system components and data to authorized users and administrators.

4. The system shall regularly undergo vulnerability assessments and penetration testing to identify and address potential security weaknesses.

3.7.2 Athorization & Authentication

This section defines how the system authenticates and authorizes users to ensure secure access and data protection.

Requirement

1. User authentication shall be performed through a secure login process, requiring unique usernames and strong passwords. Passwords should adhere to defined complexity standards.

2. Multi-factor authentication (MFA) shall be available for users, adding an extra layer of security for sensitive operations and data access.

3. Role-based access control (RBAC) shall be implemented, allowing administrators to define and manage user roles and permissions for different system components and features.

4. User sessions shall have an automatic timeout feature to log users out after a specified period of inactivity.

5. All authorization and authentication processes must comply with industry standards and best practices, including OAuth, OpenID, and SAML for third-party authentication services.

3.8 Data Management

- Data collection: The project must collect a large and representative dataset of used car sales data. This data can be collected from a variety of sources, such as online car marketplaces, government agencies, and data brokers.
- Data cleaning: The collected data needs to be cleaned to remove any errors or inconsistencies. This may involve tasks such as correcting spelling errors, removing duplicate rows, and filling in missing values.
- Data preparation: The cleaned data needs to be prepared for training and evaluating the machine learning model. This may involve tasks such as converting categorical variables to numerical variables and scaling the data.
- Data storage: The data needs to be stored in a secure and accessible location. This may involve storing the data in a database, a cloud storage service, or a local file system.

- Data security: The data needs to be protected from unauthorized access, modification, or destruction. This may involve implementing security measures such as encryption and access control.

4. USER SCENERIOS / USE CASES

1. User Requests a Price Prediction:

- *Actor:* User
- *Description:* The user accesses the system's user interface, inputs details of a used car (make, model, year, mileage, condition, etc.), and submits the request for a price prediction.
- *System Response:* The system processes the input data using its machine learning model and provides the user with a predicted fair market price for the specified vehicle.

3. User Customizes Predictions:

- *Actor:* User
- *Description:* The user has the option to customize predictions by providing additional details, such as optional features, vehicle history, or recent maintenance records.
- *System Response:* The system recalculates the price prediction based on the updated information and presents the user with the revised estimate.

4. Price Comparison for a User's Car:

- *Actor:* User
- *Description:* The user, after receiving a price prediction for their vehicle, has the option to compare their vehicle's price with similar listings in the system's database.
- *System Response:* The system presents a list of comparable used car listings, allowing the user to evaluate the competitiveness of their vehicle's price.

5. User Accesses Historical Price Trends:

- *Actor:* User
- *Description:* The user is interested in understanding how the price of a specific make and model has changed over time.
- *System Response:* The system displays historical price trends using graphical representations, enabling the user to see how prices have evolved.