Car Evaluation Report

**Introduction:**

Our data science product aims to predict the car's class based on the attributes provided by the user. The product's design is based on the car evaluation dataset, which provides six attributes and one class attribute, making it a suitable choice for accurate predictions. The end-users' requirements, including accuracy, ease of use, and simplicity, were taken into account during the product's design process. To achieve these requirements, functional and non-functional requirements were specified, and the software architecture comprises a machine learning model and a user interface. Additionally, we have adopted Waterfall methodology, quality control measures, system testing, and user evaluation to ensure the product's quality. With the use of appropriate software tools and platforms and a well-defined project management plan, we aim to successfully develop a data science product that satisfies end-users' requirements while maintaining data security and privacy.

**Product Design Section:**

Product design is a critical aspect of any software development project, and our data science product is no exception. The design of our product is based on the car evaluation dataset, which we have obtained from the UCI Machine Learning Repository. The dataset consists of six attributes, including buying, maintenance, doors, persons, lug\_boot, and safety, and one class attribute, which is the car's class. The goal of our product is to predict the car's class based on the given attributes.

In selecting the data source and theme, we have considered the end-users' requirements, which include accuracy, ease of use, and simplicity. The car evaluation dataset is a good fit for our product as it is widely used in the automotive industry and provides a rich set of attributes to make accurate predictions.

To meet the end-users' requirements, we have specified the product's functional and non-functional requirements. The functional requirements include the ability to take input from the user and make a prediction based on the input. The non-functional requirements include the product's usability, reliability, and maintainability.

To achieve these requirements, we have designed the product's software architecture. The architecture consists of two main components, a machine learning model and a user interface. The machine learning model is responsible for predicting the car's class based on the input, while the user interface is responsible for taking input from the user and displaying the prediction.

The machine learning model is trained on the car evaluation dataset using supervised learning techniques. The model uses the six attributes of buying, maintenance, doors, persons, lug\_boot, and safety to predict the car's class. We have chosen a decision tree algorithm for our machine learning model as it is simple to understand and interpret, and provides good accuracy on this type of dataset.

The user interface is designed to be simple and intuitive to use. The user is presented with a form where they can input the car's attributes. The form includes dropdown menus and radio buttons to make it easy for the user to select the attribute values. Once the user inputs the values, they can click a button to submit the form, and the product then uses the machine learning model to predict the car's class. The prediction is then displayed to the user on the screen.

The product's use case specifications include the steps involved in using the product. The user first inputs the car's attributes, and the product then uses the machine learning model to predict the car's class. The prediction is then displayed to the user. The user can then choose to input another set of attributes or exit the product.

**Product Development Section:**

Product development is a critical phase in any software development project. The development of our data science product involves selecting appropriate software tools/platforms and hardware methodologies, product development software engineering methodology, system testing method, and user evaluation plan and methods.

For software tools/platforms and hardware methodologies, we have selected Python as the programming language and Google Colab as the development platform. Python is a popular language for data science, and it has a wide range of libraries and frameworks that can be used for machine learning. Google Colab provides a free cloud-based environment for developing and testing our product. The platform offers easy access to computing resources and makes it easy to share and collaborate with team members.

For the product development software engineering methodology, we have chosen the Waterfall methodology. This methodology is well-suited for data science projects because it allows for quick iterations and feedback. We will work in sprints to deliver small, working increments of the product that can be tested and evaluated by the team and end-users.

For system testing, we have used unit tests to ensure that the machine learning model is functioning correctly. We have also used integration testing to ensure that the user interface and machine learning model are working together correctly. We will use continuous integration and deployment practices to ensure that the product is always in a releasable state.

For user evaluation, we have designed a plan to collect feedback from the end-users. We will ask the users to use the product and provide feedback on its usability, reliability, and accuracy. We will also gather feedback on the user interface and any additional features that the users would like to see in the product. We will use this feedback to improve the product and ensure that it meets the end-users' requirements.

**Project Management Section:**

Product management involves various tasks, including time management, risk assessment, quality control, customer/user relationship management, and product marketing strategy. These tasks are essential for the success of any project, and failure to implement them effectively can result in poor product quality, missed deadlines, and a negative customer experience.

**Time Management**

Time management is crucial in ensuring that a project is completed within the given timeframe. To achieve this, a Gantt chart can be created to outline the various tasks and timelines. The Gantt chart helps track the project's progress and identify any delays or issues that need to be addressed promptly.

**Risk Assessment**

Risk assessment is also an essential aspect of product management. It involves identifying potential risks and developing strategies to mitigate them. For example, when handling personal information and data, risks such as data privacy breaches, data security breaches, misuse of data, technical failures, and delays in project timelines need to be mitigated. Mitigation strategies include encryption of sensitive data, implementation of access controls, regular backups of data, regular testing of system security, contingency plans in case of technical failures, and regular communication with stakeholders to address any delays or issues.

Quality control measures are necessary to ensure that the software meets the required standards. Regular code reviews, testing, and debugging help identify errors and issues. Version control is also implemented to track changes made to the codebase and ensure that the latest version is always in use.

**Customer/User Relationship Management**

Customer/user relationship management is another crucial aspect of product management. It involves gathering feedback from users to identify areas for improvement and providing regular updates and support to ensure that their needs are met. User surveys and interviews are conducted to gather feedback, which is then used to improve the product and enhance the user experience.

Product marketing strategy is also an essential aspect of product management. Effective marketing strategies help reach a wider audience and increase sales. Creating a website to promote the product, creating social media accounts, and running targeted ads are some of the strategies used to market the product.

In conclusion, effective product management requires various tasks to be completed, including time management, risk assessment, quality control, customer/user relationship management, and product marketing strategy. Failure to implement these tasks effectively can lead to poor product quality, missed deadlines, and a negative customer experience.

**Conclusion:**

The main goal of the data science product described in the introduction is to predict the car's class based on the attributes provided by the user. The six attributes used to predict the car's class are buying, maintenance, doors, persons, lug\_boot, and safety. The decision tree algorithm is used for the machine learning model. The product's software architecture comprises a machine learning model and a user interface. The machine learning model is responsible for predicting the car's class based on the input, while the user interface is responsible for taking input from the user and displaying the prediction. The software tools and platforms used for product development are Python as the programming language and Google Colab as the development platform. The Waterfall methodology is used as the software engineering methodology for product development. Unit tests and integration testing are used as system testing methods. The user evaluation plan involves collecting feedback from end-users on the product's usability, reliability, and accuracy. Project management tasks for this product include time management, risk assessment, quality control, customer/user relationship management, and product marketing strategy.

**References:**

• Software Engineering Methodologies:

https://en.wikipedia.org/wiki/Software\_development\_process <https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-itsadvantages-disadvantages>

• Car Evaluation Dataset Link:

[UCI Machine Learning Repository: Car Evaluation Data Set](https://archive.ics.uci.edu/ml/datasets/Car+Evaluation)