PROJECT DESIGN

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	For a given gender, annual salary and age the model must predict if the person can buy a car based on the data given.
2.	Idea / Solution description	In our solution we have analyzed the data ,preprocessed it and checked for the model with the most accuracy.
3.	Novelty / Uniqueness	In our model we used GridsearchCV with highest accuracy .
4.	Social Impact / Customer Satisfaction	Our customers will be immensely satisfied with our model the accuracy is high.

5.	Business Model (Revenue Model)	Customer Segments: Individuals interested in buying a car. Auto dealerships looking for potential customers. Financial institutions interested in targeting potential car loan customers. Value Propositions: Accurate and personalized car-buying predictions. Time-saving for customers and dealerships in identifying potential leads. Improved targeting for financial institutions offering car loans. Channels: Web platform: The primary channel for users to access and use the prediction tool. Marketing: Utilize online marketing channels (social media, search engine optimization) to drive traffic. Partnerships: Collaborate with auto dealerships and financial institutions for mutual benefits.
6.	Scalability of the Solution	The model can work in large scales and has the ability to provide optimum results.

Solution Architecture:

Web Application:

Frontend (Client-Side):

Developed a responsive and user-friendly interface using modern web technologies (HTML, CSS, JavaScript).

2. Prediction Engine:

Machine Learning Model

Develop and train a machine learning model for car-buying predictions. Common algorithms include regression models, decision trees, or more advanced techniques like neural networks.

Use popular ML frameworks like TensorFlow or PyTorch.

Model Deployment:

Deployed the trained model using Flask.
Implement versioning to manage updates and rollbacks of the prediction model.

3. APIs and Microservices:

User Management Microservice:

Manage user authentication, authorization, and account information.

Communicate with other microservices to retrieve and update user-related data.

Prediction Microservice:

Handle requests from the frontend and interact with the prediction engine to provide car-buying predictions.

Implement load balancing for handling varying prediction requests.

Data Processing Microservice:

Handle data processing tasks, such as cleaning and transforming incoming user data before it is fed into the prediction engine.

Utilize asynchronous processing for non-real-time tasks.

5. Scalability Measures:

Load Balancers:

Introduce load balancers to distribute incoming web traffic across multiple servers.

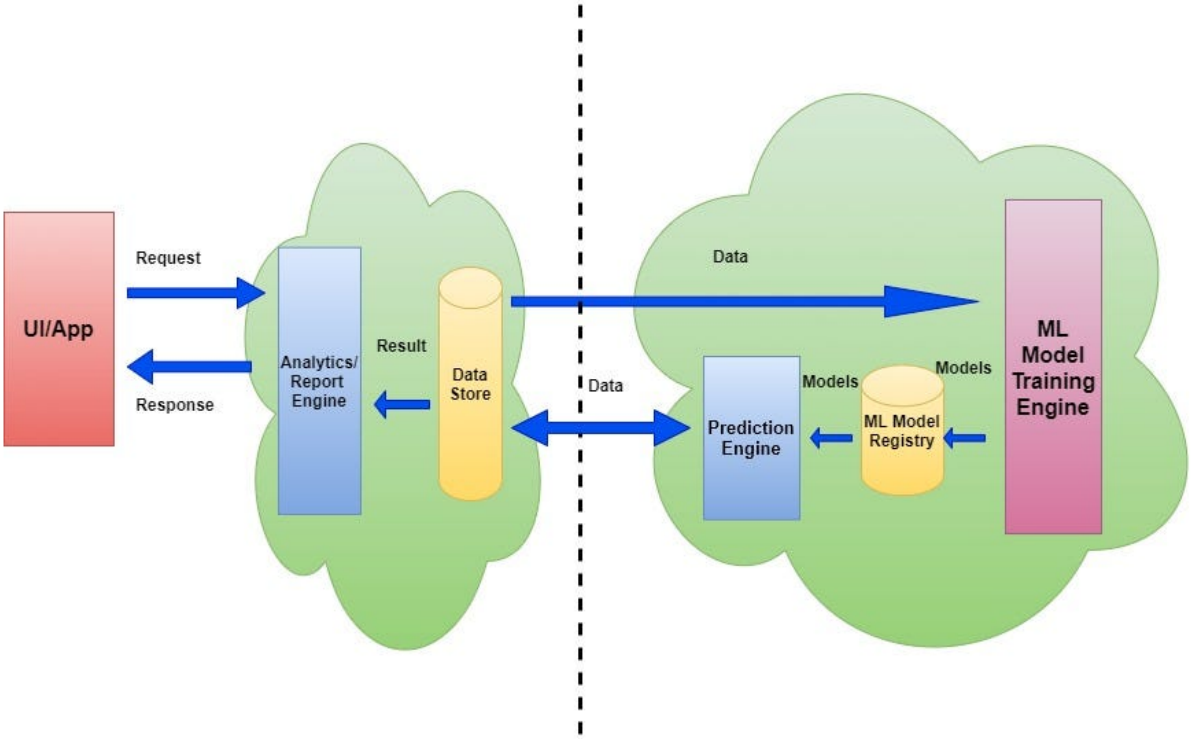
Consider using a service like AWS Elastic Load Balancing or Nginx.

Auto-Scaling:

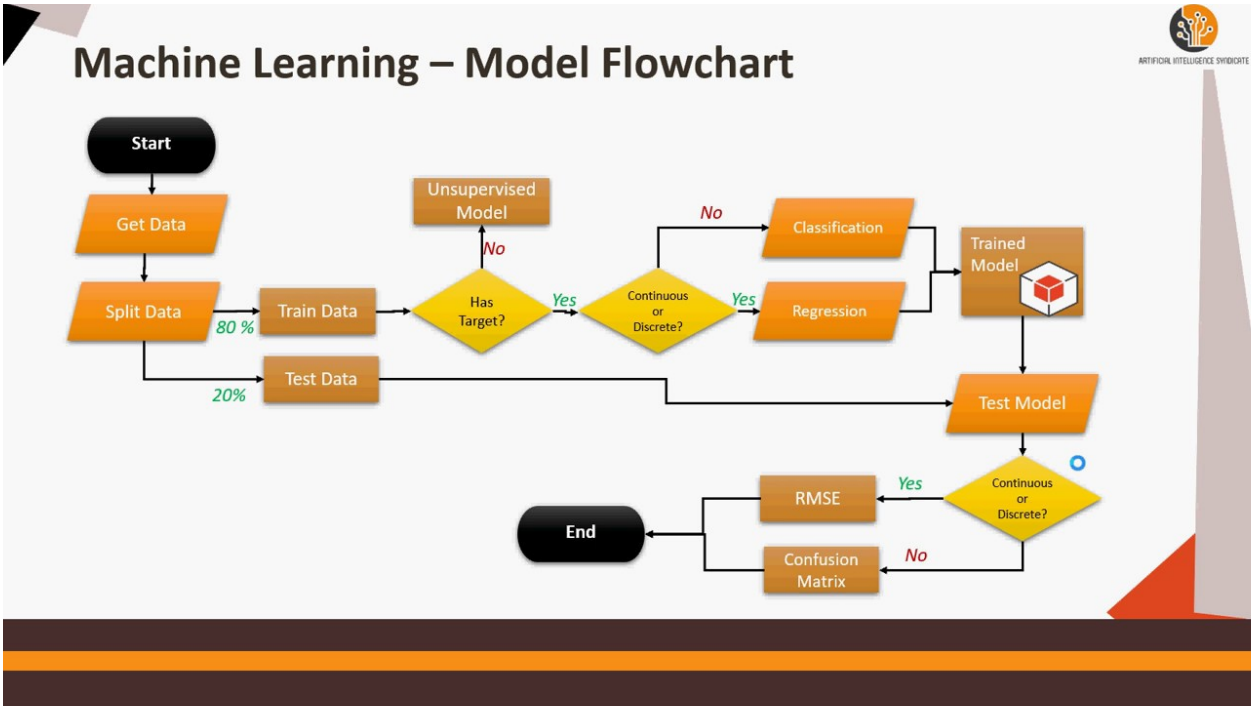
Implement auto-scaling for both the frontend and backend components to handle varying levels of user traffic.

Use cloud provider features or container orchestration tools like Kubernetes for this purpose.

Example - Solution Architecture Diagram:



Data Flow Diagrams:



Machine Learning – Model Flowchart

Start

Get Data

Split Data

80 %

Train Data

20%

Test Data

Has Target?

No

Unsupervised Model

Yes

Continuous or Discrete?

No

Classification

Yes

Regression

Trained Model

Test Model

Continuous or Discrete?

Yes

RMSE

No

Confusion Matrix

End