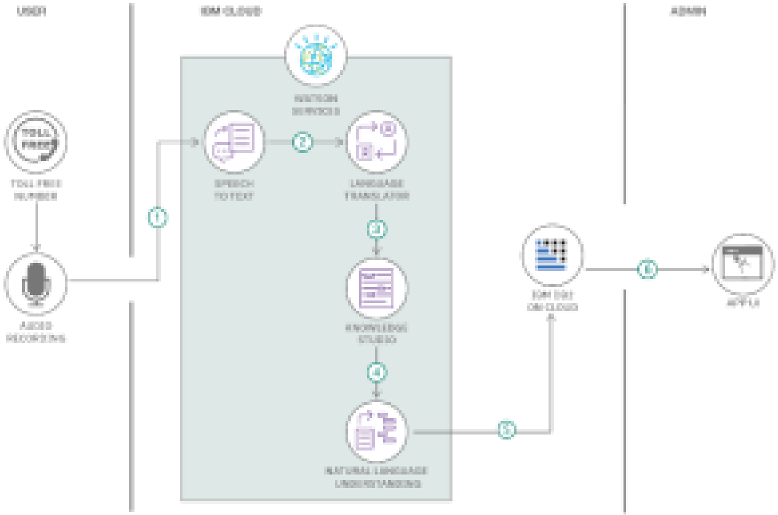
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27 October 2023
Team ID	Team-591995
Project Name	Project - car purchase prediction using ML
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2 **Project:Car Purchase Prediction Using ML**

Reference:



https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)

3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Developing a user interface to present predictions and relevant information to end-users.	● Web development frameworks (React, Angular, Vue.js) ● Visualization libraries (D3.js, Chart.js)
2.	Data Collection and Preprocessing	This component involves gathering data from various sources and preparing it for analysis. This can include cleaning data, handling missing values, and converting it into a suitable format.	● Data Collection: APIs, databases, data warehouses ● Data Preprocessing: Python libraries (Pandas, NumPy), SQL for data cleaning, and transformat.
3.	Feature Engineering	Feature engineering is the process of selecting and transforming raw data into features that can be used by machine learning models for better predictions.	● Python libraries (Pandas, NumPy) for data manipulation ● Scikit-learn for feature selection and extraction
4.	Machine Learning Model Selection:	Choosing the appropriate machine learning models based on the nature of the prediction task.	● Scikit-learn for traditional machine learning models ● TensorFlow or PyTorch for deep learning models

5.	Training and Testing:	Splitting the dataset into training and testing sets to train the model and evaluate its performance	Scikit-learn for model training and evaluation
6.	Model Deployment:	Deploying the trained machine learning model to a production environment.	● Cloud services (AWS, Azure, Google Cloud) ● Containerization tools like Docker ● Serverless computing (AWS Lambda, Azure Functions)
7.	Real-time Data Processing:	Setting up a pipeline to handle real-time data from car sensors or other sources.	● Apache Kafka or Apache Flink for stream processing ● Python (Flask, FastAPI) for building APIs to handle real-time data
8.	Scalability and Performance Optimization:	Ensuring the architecture can scale to handle a larger volume of data and optimizing performance.	● Kubernetes for container orchestration ● Load balancing technologies
9.	Monitoring and Logging:	Implementing tools to monitor the system's performance and log relevant information for troubleshooting.	● ELK Stack (Elasticsearch, Logstash, Kibana) for log management ● Prometheus and Grafana for monitoring
10.	Security and Privacy:	Implementing security measures to protect sensitive data and ensure user privacy.	● Encryption algorithms for data protection ● Identity and access management

			(IAM) solutions
11.	Documentation and Maintenance:	Documenting the architecture, code, and processes for future reference and establishing a maintenance plan.	● Documentation tools (Swagger, Sphinx) ● Version control systems (Git) for code management

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Data Collection and Preprocessing: ● Involves acquiring data from various sources. ● Cleans and preprocesses data to make it suitable for analysis.	Data collection and preprocessing ensure that the data used for training and testing machine learning models is clean, accurate, and in a format that can be effectively utilized.	● Data Collection: APIs (e.g., RESTful APIs), databases (e.g., MySQL, MongoDB), data warehouses (e.g., Amazon Redshift) ● Data Preprocessing: Python libraries (Pandas, NumPy), SQL for data cleaning and transformation
2.	Feature Engineering: ● Selects and transforms raw data into relevant features. ● Enhances the predictive power of machine learning models.	Feature engineering is crucial for creating input features that capture the essential patterns and relationships within the data, improving the model's ability to make accurate predictions.	● Python libraries (Pandas, NumPy) ● Scikit-learn for feature selection and extraction

3.	Machine Learning Model Selection: ● Involves choosing appropriate algorithms for the prediction task. ● Selection based on the nature of the data and the problem at hand.	Model selection is a critical decision that impacts the accuracy and efficiency of predictions. It involves choosing the best-suited machine learning algorithm or a combination of algorithms for the specific task.	● Scikit-learn for traditional machine learning models ● TensorFlow or PyTorch for deep learning models

S.No	Characteristics	Description	Technology
4.	Training and Testing: ● Splits the dataset into training and testing sets. ● Trains the machine learning model on the training set and evaluates its performance on the testing set.	Training and testing ensure that the machine learning model learns from the data and can generalize well to make accurate predictions on new, unseen data.	Scikit-learn for model training and evaluation
5.	Model Deployment: ● Involves deploying the trained model to a production environment. ● Enables the model to make predictions on new data.	Model deployment takes the trained model from the development environment and makes it available for use in real-world scenarios, allowing it to make predictions on live data.	● Cloud services (AWS, Azure, Google Cloud) ● Containerization tools like Docker ● Serverless computing (AWS Lambda, Azure Functions)

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d