

**Project Design Phase-I**  
**Solution Architecture**

Date	15 November 2023
Team ID	Team ID - 592603
Project Name	Smart Lender - Applicant Credibility Prediction For Loan Approval
Maximum Marks	4 Marks

**Solution Architecture:**

Solution architecture for a "Smart Lender - Applicant Credibility Prediction For Loan Approval " project involves several components. Here's a high-level overview of the architecture:

**1. Data Collection and Preprocessing:**

- Gather historical loan applicant data including personal information, financial history, credit scores, employment details, etc.
- Clean the data, handle missing values, encode categorical variables, and perform feature scaling if necessary.

**2. Feature Engineering:**

- Extract relevant features or create new features that might contribute to the prediction accuracy.
- Apply any necessary transformations or feature engineering techniques.
- Visualization of relationship between features (Heat map etc...)

**3. Machine Learning Model Development:**

- Choose appropriate machine learning algorithms for loan prediction.
- Split the dataset into training and testing sets for model training and evaluation.
- Train the machine learning model using the training data
- Train multiple classification algorithms (Decision Tree, Random Forest, KNN, XGBoost) using the preprocessed data.

#### 4. Model Evaluation:

- Evaluate the performance of the machine learning model using the testing dataset.
- Evaluate each model's performance using metrics like accuracy, precision, recall, and F1-score on a validation set.

#### 5. Flask Integration:

- API Development: Create Flask-based APIs to receive loan applicant data in a structured format and provide predictions based on the trained model.
- Endpoints: Design endpoints for receiving data (e.g., applicant details) and returning predictions (approval or rejection).

#### 6. IBM Cloud Deployment:

- IBM Cloud Services: Leverage IBM Cloud services like IBM Watson Studio for model development, IBM Cloud Functions or IBM Cloud Foundry for deploying the Flask application.
- Containerization (Optional): Utilize containerization platforms (like Docker) if needed for efficient deployment and scalability.
- Monitoring and Logging: Implement monitoring and logging mechanisms for tracking the application's performance and handling errors.

#### 7. Security Measures:

- Implement encryption techniques to secure sensitive applicant information during data transmission and storage.
- Define access controls and authentication mechanisms to ensure that only authorized personnel can access the application and its data.
- Ensure compliance with data protection regulations (like GDPR, HIPAA) and industry-specific security standards.

#### 8. Model Monitoring and Maintenance:

- Set up continuous monitoring to track the model's performance in real-time, identify degradation, and trigger alerts for necessary retraining.
- Implement drift detection mechanisms to identify changes in data distribution that might affect model performance.

- Schedule periodic retraining of the model using updated data to maintain its accuracy and relevance.

#### 9. User Interface (UI) Development:

- Develop a user-friendly dashboard to visualize key metrics, model performance, and approved/rejected loan statistics.
- Provide interfaces for bank personnel to interact with the model, input applicant details, and review predictions.

#### 10. Logging and Auditing:

- Implement comprehensive logging mechanisms to record activities, requests, and responses for auditing and debugging purposes.
- Conduct regular audits to ensure adherence to established protocols, compliance, and security standards.

#### 11. Error Handling and Recovery:

- Develop robust error-handling mechanisms to gracefully manage exceptions, invalid inputs, or system failures.
- Implement strategies for system recovery in case of failures, ensuring minimal downtime and data integrity.

#### 12. Feedback Loop and Model Improvement:

- Establish a system to gather feedback on loan decisions and use this data to continuously improve the model's accuracy.
- Implement a process to incorporate feedback into future iterations of the model to enhance its predictive capabilities.

#### 13. Scalability and Resource Management:

- Design the system to handle increased loads efficiently by employing scalable infrastructure and resources.
- Optimize resource utilization to ensure cost-effectiveness and efficient use of computational resources.

#### 14. Documentation:

- Create comprehensive documentation for the solution architecture, including data sources, model details, and system components.

#### 15. Deployment and Maintenance:

- Deploy the solution to a production environment.
- Establish a maintenance plan for regular updates, model retraining, and addressing potential issues.

#### Example - Solution Architecture Diagram:

