

Credit Card Default Prediction System Architecture Design

1. Data Ingestion:

- **Objective:** Collect and integrate data from various sources.
- **Components:**
 - **Data Sources:** Credit card transaction history, customer information, credit limits.
 - **Data Ingestion Service:** Responsible for collecting, cleaning, and preprocessing raw data.

2. Data Storage:

- **Objective:** Store and manage structured data efficiently.
- **Components:**
 - **Database:** SQL or NoSQL database to store the preprocessed and cleaned data.
 - **Data Warehouse (Optional):** For handling large volumes of historical data for analytics purposes.

3. Feature Engineering:

- **Objective:** Transform raw data into meaningful features for model training.
- **Components:**
 - **Feature Extraction Module:** Extract relevant information from transaction history, demographics, etc.
 - **Transformation Logic:** Apply transformations, scaling, and encoding for machine learning compatibility.

4. Model Development:

- **Objective:** Train machine learning models for credit card default prediction.
- **Components:**
 - **Model Training Service:** Train logistic regression, decision tree, and random forest models.
 - **Hyperparameter Tuning Module:** Optimize model parameters for better performance.

5. Model Evaluation:

- **Objective:** Assess the performance of trained models.
- **Components:**
 - **Metrics Calculation Module:** Evaluate accuracy, precision, recall, and F1-score.
 - **Visualization Module:** Display performance metrics through dashboards or reports.

6. Model Deployment:

- **Objective:** Make trained models available for predictions in a production environment.
- **Components:**
 - **Deployment Service:** Deploy models using containerization (e.g., Docker) or serverless architecture.
 - **API Gateway:** Manage API endpoints for model predictions.

7. Real-Time Prediction:

- **Objective:** Enable real-time credit card default predictions.
- **Components:**
 - **Real-Time Prediction Service:** Accept requests and invoke the deployed machine learning models.
 - **Scalability Measures:** Ensure the system can handle varying request loads.

8. User Interface (Optional):

- **Objective:** Provide a user-friendly interface for interacting with the system.
- **Components:**
 - **Web Application:** A dashboard for users to input data and receive predictions.
 - **User Authentication:** Secure access to the system.

9. Monitoring and Logging:

- **Objective:** Track system performance and log activities.
- **Components:**
 - **Monitoring Tools:** Monitor system health, response times, and resource usage.
 - **Logging Service:** Capture logs for debugging and auditing.

10. Security Measures:

- **Objective:** Ensure data privacy and protect against security threats.
- **Components:**
 - **Encryption Mechanisms:** Encrypt sensitive data during storage and transmission.
 - **Access Control:** Implement role-based access control for different system components.

11. Continuous Integration and Deployment (CI/CD):

- **Objective:** Automate the testing, integration, and deployment processes.
- **Components:**
 - **CI/CD Pipeline:** Automate the building, testing, and deployment of code changes.
 - **Version Control:** Use a version control system for code management.

12. Scalability:

- **Objective:** Design the system to scale horizontally or vertically based on demand.
- **Components:**
 - **Load Balancers:** Distribute incoming traffic to maintain system performance.
 - **Auto-Scaling:** Dynamically adjust resources based on demand.