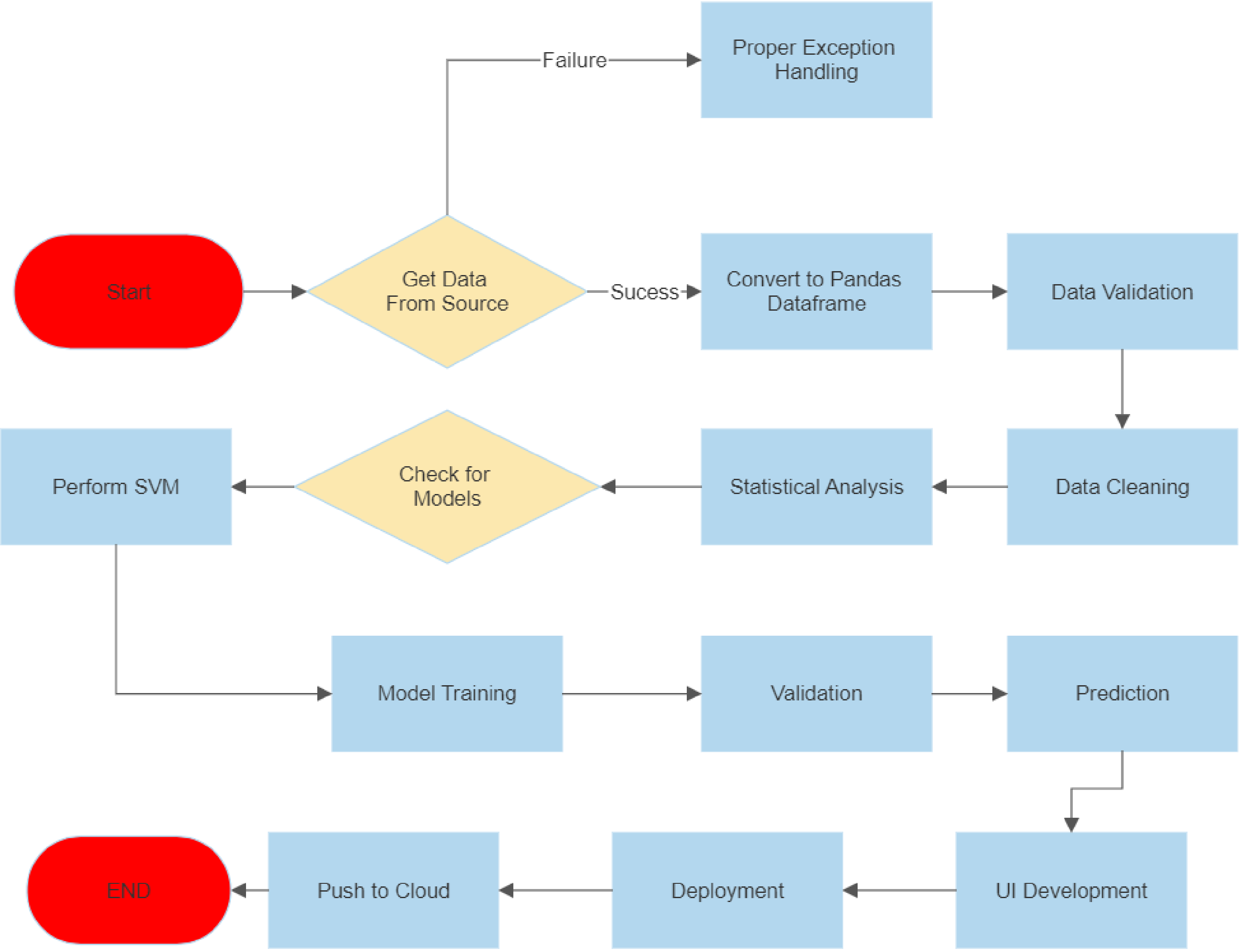
Architecture

Model Training/Validation Workflow



Data Collection: Historical credit card data is collected from various sources to create a dataset for model training.

Data Cleaning: The collected data is cleaned by removing duplicate entries, handling missing values, and dealing with anomalies to ensure data quality.

Data Preprocessing: The preprocessed data undergoes preprocessing steps such as feature scaling, normalization, and encoding categorical variables to prepare it for model training.

Feature Selection: The relevant features are selected from the preprocessed data, eliminating unnecessary variables that do not contribute significantly to the model's predictive power.

Model Selection: Different machine learning models, such as Random Forest, Support Vector Machines (SVM), and Logistic Regression, are evaluated and compared to choose the best-performing model.

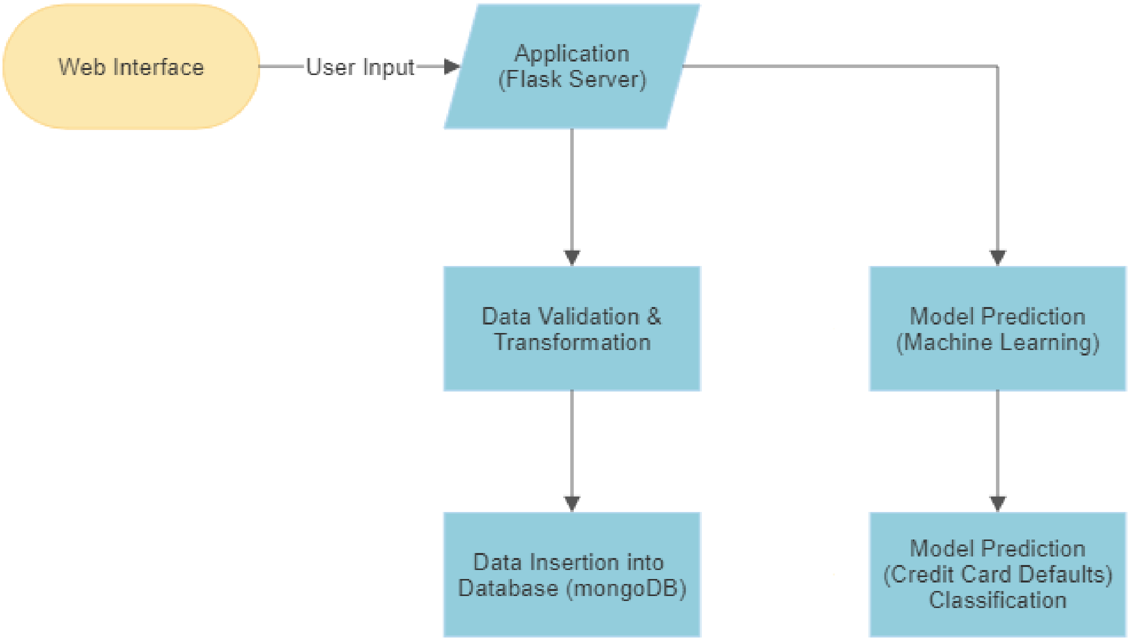
Model Training (Machine Learning): The selected model is trained using preprocessed data to learn from historical patterns and relationships.

Model Evaluation (Performance Metrics): The trained model's performance is evaluated using various performance metrics like accuracy, ROC curve, precision, recall, and F1-score.

Model Clustering (Optional Step): If applicable, the trained model can be clustered with other models based on certain criteria, creating model groups with similar characteristics.

Model Deployment (Web Application): The best-performing model is deployed in a web application (Flask server) to make real-time predictions based on user input.

User Input/Output Workflow


Web Interface
User Input
Application (Flask Server)
Data Validation & Transformation
Model Prediction (Machine Learning)
Data Insertion into Database (mongoDB)
Model Prediction (Credit Card Defaults) Classification

Web Interface: The workflow starts with the web interface, which serves as the front-end for user interaction. Users input their credit card information, including attributes like credit limit, gender, education, marital status, age, repayment status, bill amount, and previous payment amount.

Application (Flask Server): The web interface communicates with the Flask server, which acts as the back-end of the application. It receives user input and passes it to the subsequent components.

Data Validation & Transformation: The user-input data undergoes data validation to ensure that it meets the required format and constraints. Then, data transformation is applied to preprocess the data, ensuring it aligns with the format expected by the machine learning model.

Data Insertion into Database (MongoDB): After validation and transformation, the preprocessed data is inserted into the MongoDB database for storage and future reference. This allows for efficient data management and retrieval.

Model Prediction (Machine Learning): A pre-trained machine learning model (such as Random Forest or SVM) is used to predict the likelihood of credit card default based on the user's input data. The model leverages the historical credit card default data for prediction.

Model Prediction (Credit Card Default Classification): The model provides a prediction for the credit card default classification, indicating whether the user is likely to default on their credit card payments or not.

Result Display: The prediction result is communicated back to the Flask server, which sends the result to the web interface for display to the user. The result will be either "Default Payment" or "No Default Payment" based on the model's prediction.