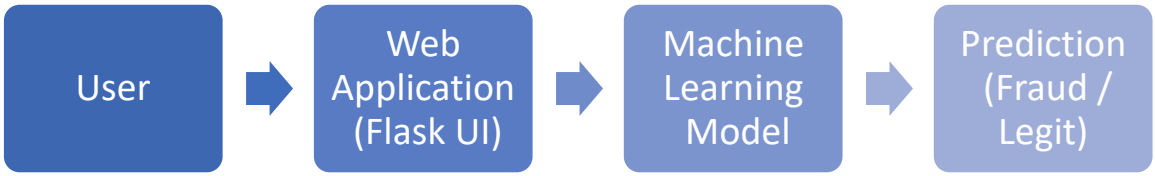
Requirement Analysis Phase – Online Payments Fraud Detection Project

1. Introduction

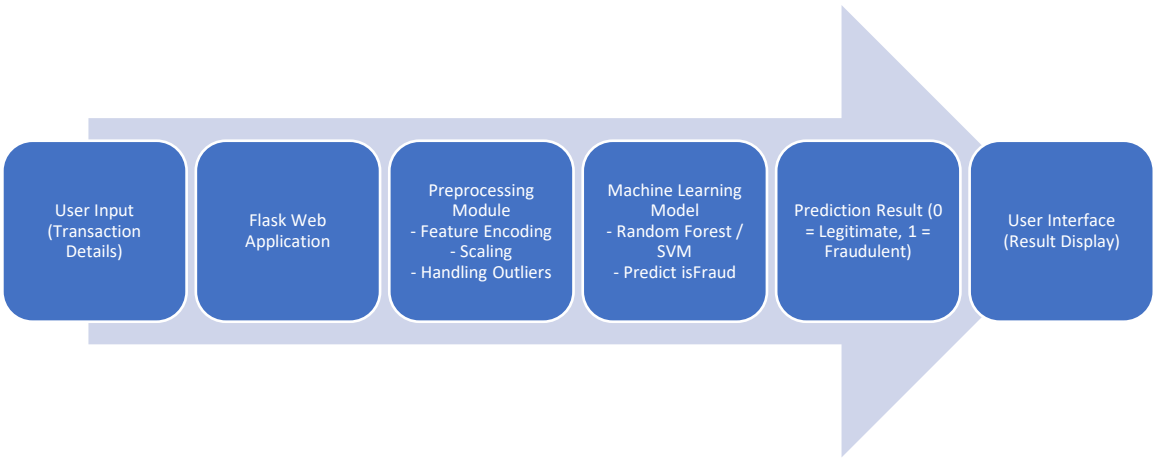
The Requirement Analysis Phase defines **what the system should do** (functional requirements), **how it should perform** (non-functional requirements), and the **resources needed**. This phase ensures a clear understanding of project scope, user needs, and technology requirements before development begins.

2. Data Flow Diagram (DFD)

Level 0 – Context Diagram



Level 1 – Detailed Data Flow:



3. Solution Requirements

3.1 Functional Requirements

1. **Data Handling**
    - Load transaction dataset (fraud_data.csv).
    - Handle **missing values, outliers, and categorical features**.
2. **Exploratory Data Analysis (EDA)**

- Generate **descriptive statistics**.
- Plot **countplots, distplots, and heatmaps**.

3. **Model Building**
    - Train **Random Forest, Decision Tree, Extra Trees, and SVM** models.
    - Evaluate models using **accuracy, confusion matrix, precision, and recall**.
    - Save the trained model with **Pickle**.
4. **Real-Time Prediction**
    - Accept **user input through web form**.
    - Return **fraud prediction (0 or 1)** immediately.
5. **Web Application**
    - Provide a **Flask-based interface**.
    - Validate user input and display results clearly.

3.2 Non-Functional Requirements

- **Performance:** Real-time prediction in <1 second.
- **Usability:** User-friendly web interface.
- **Reliability:** High model accuracy with minimal false negatives.
- **Scalability:** Ability to retrain model with new data.
- **Security:** Protect sensitive transaction data.

4. Technology Stack

Layer / Component	Technology / Tool	Purpose
Programming Language	Python 3.x	Backend development and ML implementation
Data Analysis & ML	Pandas, NumPy, Scikit-learn, Pickle	Data manipulation, model training, saving model
Visualization	Matplotlib, Seaborn	Visual analysis of features and fraud patterns
Web Application Framework	Flask	Real-time interaction with ML model
Frontend / UI	HTML, CSS	Display input forms and prediction results

Preprocessing	StandardScaler, LabelEncoder	Scale numeric features and encode categorical
IDE / Environment	PyCharm, Jupyter Notebook, Anaconda	Development and testing

5. Conclusion

The **Requirement Analysis Phase** ensures that the system has:

- A **clear data flow** from user input to prediction output.
- Well-defined **functional and non-functional requirements**.
- An appropriate **technology stack** to implement machine learning and web deployment.

This phase provides a **solid foundation for the design and development** of the Online Payments Fraud Detection system.