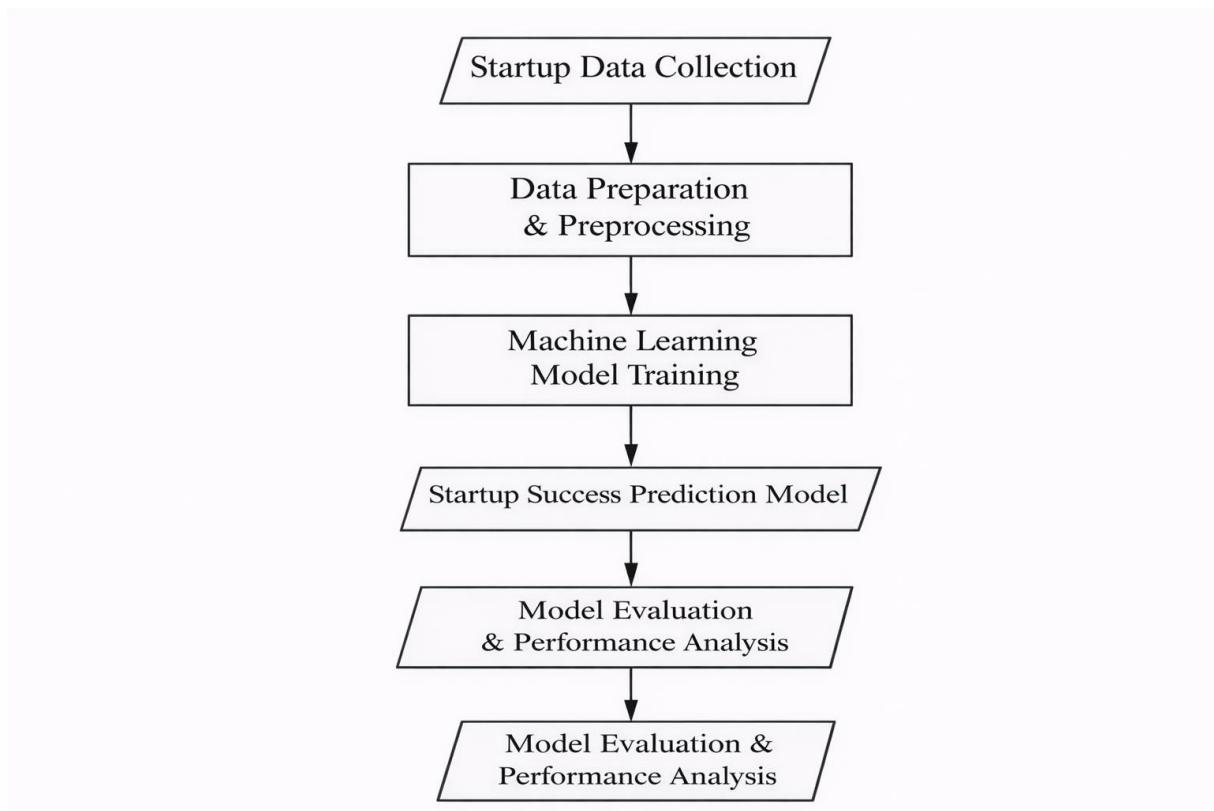


## Project Design Phase-II

### Technology Stack (Architecture & Stack)

Date	14 Feb 2026
Team ID	LTVIP2026TMIDS47450
Project Name	prosperity prognosticator: machine learning for startup success prediction
Maximum Marks	4 Marks

#### Technical Architecture:



Example: Rice Type Classification Using Advanced Machine Learning Techniques

**Table-1: Components & Technologies:**

S.No.	Component	Description	Technology
1.	User Interface	Simple web interface for entering startup details and viewing predictions	HTML, CSS, Flask / Streamlit
2.	Application Logic-1	Data preprocessing (handling missing values, encoding, feature scaling)	Python (Pandas, NumPy, Scikit-learn)

3.	Application Logic-2	Model training and prediction using ML algorithms	Python (Scikit-learn, Random Forest, Logistic Regression)
4.	Application Logic-3	Evaluation metrics visualization and reporting	Matplotlib, Seaborn
5.	Database	Startup dataset stored locally	CSV / Excel files, Pandas
6.	Cloud Database	(Not used)	---
7.	File Storage	Dataset and trained model stored locally	Local filesystem / Joblib (.pkl files)
8.	External API-1	(Not used)	---
9.	External API-2	(Not used)	---
10.	Machine Learning Model	Predict startup success (Success / Failure) using classification	Random Forest Classifier (ML)
11.	Infrastructure	Executed in local browser or cloud notebook environment	Jupyter Notebook, Google Colab, Flask

**Table-2: Application Characteristics:**

S.No.	Characteristic	Description	Technology
1	Open-Source Frameworks	Used for data preprocessing, model training, and deployment	Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Flask
2	Security Implementations	Data processed locally; no external data exposure	Local execution (offline or secured hosting)
3	Scalable Architecture	The system can be deployed as a web application and extended with REST APIs for integration with investment or analytics platforms.	Flask / Streamlit
4	Availability	Application can be hosted locally or on cloud platforms, making it accessible anytime through a browser.	Google Colab, Render, Heroku (optional)
5	Performance	Lightweight machine learning model optimized for fast real-time startup success predictions.	Scikit-learn (Random Forest), Python (CPU-based execution)