Solution Requirements

Prosperity Prognosticator: ML For Startup Success Prediction

1. Functional Requirements

FR ID	Requirement	Priority
FR-01	System shall accept startup metrics via a web form	High
FR-02	System shall preprocess and encode input data automatically	High
FR-03	System shall load the pre-trained ML model (.pkl file)	High
FR-04	System shall predict startup success or failure	High
FR-05	System shall display the prediction result to the user	High
FR-06	System shall support multiple ML algorithm comparison (notebook)	Medium
FR-07	System shall perform hyperparameter tuning to optimize accuracy	Medium
FR-08	System shall save the best-performing model to disk	High
FR-09	System shall display feature importance analysis	Medium
FR-10	System shall route users between Home and Result pages	High

2. Non-Functional Requirements

NFR ID	Requirement	Target
NFR-01	Performance – Prediction response time	< 3 seconds
NFR-02	Accuracy – Model prediction accuracy	> 80%
NFR-03	Usability – Simple, intuitive web interface	No technical knowledge required
NFR-04	Reliability – Consistent prediction results	Deterministic model outputs
NFR-05	Maintainability – Clean, documented codebase	Modular Python scripts
NFR-06	Portability – Runs on local machine	Python 3.x + Flask

3. Input Requirements

The system requires the following startup features as input:

- Age at first funding (years)
- Age at last funding (years)
- Age at first milestone (years)
- Age at last milestone (years)
- Number of relationships / connections
- Number of funding rounds
- Funding total (USD)
- Milestones achieved
- Number of Avg Participants
- Is_Top500

4. Output Requirements

- Binary prediction: Success (1) or Failure (0)
- Clear result message displayed on the results page
- Option to return to home and predict again

5. Constraints

- Must use Python and Scikit-learn for model building
- Deployment must use Flask web framework
- Dataset must be from Kaggle (startup_data.csv)
- Model must be saved as a .pkl file for integration