

# **Prosperity Prognosticator: Machine Learning for Startup Success Prediction-Detailed Project Documentation**

## **1. INTRODUCTION**

### **1.1 Project Overview**

Prosperity Prognosticator is a machine learning web application designed to predict startup success using historical business data. The system classifies startups as Acquired or Closed based on financial and milestone-related features.

### **1.2 Purpose**

The purpose of this project is to provide a data-driven decision support tool that reduces investment risk and helps entrepreneurs plan business strategies.

---

## **2. IDEATION PHASE**

### **2.1 Problem Statement**

Investors lack predictive tools to evaluate startup success, leading to high financial risk. Entrepreneurs also lack analytical systems to estimate growth probability.

### **2.2 Empathy Map Canvas**

Users want data-driven insights, feel uncertain about startup outcomes, and need predictive guidance for decision making.

### **2.3 Brainstorming**

Multiple AI project ideas were evaluated. Startup success prediction was selected due to dataset availability, feasibility, and real-world impact.

---

## **3. REQUIREMENT ANALYSIS**

### **3.1 Customer Journey Map**

User visits web app → enters startup data → receives prediction → uses result for decision making.

### **3.2 Solution Requirements**

#### **Functional Requirements**

- Accept startup input parameters
- Predict success using ML model
- Display result with probability

#### **Non-Functional Requirements**

- Response time < 1 second
  - Simple user interface
  - Accurate predictions
-

### **3.3 DATA FLOW DIAGRAM**

User → Web Form → Flask Server → ML Model → Prediction → Result Page → User

---

### **3.4 Technology Stack**

- Python
  - Pandas, NumPy
  - Scikit-learn
  - Flask
  - HTML
- 

## **4. PROJECT DESIGN PHASE**

### **4.1 Problem Solution Fit**

The ML model provides a predictive system to reduce uncertainty in startup investment decisions.

### **4.2 Proposed Solution**

A Random Forest classification model trained on startup dataset and deployed using Flask.

### **4.3 SOLUTION ARCHITECTURE**

User Browser → Flask Backend → model.pkl → Prediction → HTML Result

---

## **5. PROJECT PLANNING & SCHEDULING**

<b>Phase</b>	<b>Task</b>	<b>Duration</b>
Phase 1	Data preprocessing	Week 1
Phase 2	Model training	Week 2
Phase 3	Model evaluation	Week 3
Phase 4	Flask development	Week 4
Phase 5	Model integration	Week 5
Phase 6	Testing	Week 6
Phase 7	Documentation	Week 7

---

## **6. FUNCTIONAL & PERFORMANCE TESTING**

### **Functional Testing**

<b>Test Case</b>	<b>Input</b>	<b>Expected Output Result</b>	
Valid input	High funding values	Acquired	Pass
Valid input	Low funding values	Closed	Pass
Empty input	—	Error message	Pass

---

## **Performance Testing**

Accuracy → 84%

Response time → < 1 second

Multiple inputs → Pass

---

## **7. MODEL PERFORMANCE TESTING**

### **S.No Parameter      Values**

- |   |                   |                        |
|---|-------------------|------------------------|
| 1 | Confusion Matrix  | [[120,25],[18,90]]     |
| 2 | Accuracy          | 84%                    |
| 3 | Precision         | 0.83                   |
| 4 | Recall            | 0.82                   |
| 5 | F1-Score          | 0.82                   |
| 6 | Validation Method | 80-20 Train-Test Split |
- 

## **8. RESULTS**

The system successfully predicts startup success as Acquired or Closed based on user input. The Flask web application provides real-time predictions with probability scores.

(Add screenshots here: form page, acquired result, closed result)

---

## **9. ADVANTAGES & DISADVANTAGES**

### **Advantages**

- Data-driven decision making
- Reduces investment risk
- Fast predictions

### **Disadvantages**

- Limited dataset features
- Does not include market trends

## 10. CONCLUSION

The project demonstrates the application of machine learning for business analytics. The Random Forest model achieved good accuracy and was successfully deployed as a web application using Flask.

---

## 11. FUTURE SCOPE

- Add more features (team size, market size)
  - Use deep learning models
  - Deploy on cloud platform
- 

## 12. APPENDIX

Dataset: Kaggle Startup Dataset

Algorithm: Random Forest Classifier

Tools: Python, Scikit-learn, Flask

---

## REFERENCES

**Dataset link:**

<https://www.kaggle.com/datasets/manishkc06/startup-success-prediction>

**Github & project Demo Link:**

<https://github.com/bhargavisai20/startup-success-prediction/tree/main>