STARTUP SUCCESS PREDICTION USING MACHINE LEARNING
Detailed Project Report

---

1. INTRODUCTION

1.1 Project Overview

The Startup Success Prediction system is a machine learning-based web application developed to predict whether a startup is likely to succeed or fail based on historical data patterns. The project uses supervised learning classification algorithms trained on a Kaggle startup dataset. The system analyzes various startup features such as funding amount, operational factors, and business indicators to generate a prediction.

The complete system includes:

- Data preprocessing and feature engineering
- Model training and evaluation
- Random Forest classification model
- Model serialization using Pickle
- Web application development using Streamlit
- GitHub version control and deployment readiness

The model was trained and tested using Google Colab, and the final trained model was integrated into a frontend application for real-time predictions.

1.2 Purpose

The main purpose of this project is to help:

- Entrepreneurs evaluate startup viability
- Investors assess risk before funding
- Policymakers understand success-driving factors
- Business planners make data-driven decisions

The system reduces uncertainty by using predictive analytics instead of intuition-based decisions.

---

2. IDEATION PHASE

2.1 Problem Statement

Many startups fail due to poor financial planning, insufficient funding, lack of market understanding, and operational inefficiencies. Entrepreneurs and investors often rely on assumptions rather than analytical evidence.

Problem:
There is no simple AI-based tool that predicts startup success probability using historical data and machine learning techniques.

Objective:
To develop a predictive system that classifies startups as “Successful” or “Not Successful” using machine learning algorithms.

2.2 Empathy Map Canvas

Entrepreneurs
Think: “Will my startup survive in the market?”
Feel: Fear of financial loss and failure
Say: “I need data-driven insights.”
Do: Research, pitch to investors, analyze competitors

Investors
Think: “Is this investment safe?”
Feel: Risk-sensitive
Say: “Show me numbers and analysis.”
Do: Evaluate funding rounds and ROI potential

Policymakers
Think: “How can we support innovation?”
Feel: Responsible for economic growth
Say: “We need startup analytics.”
Do: Create policies and funding programs

2.3 Brainstorming

Different ideas considered:

- Startup risk scoring dashboard
- Investment recommendation engine
- Business performance analytics tool
- Startup success classification model

Final Idea Selected:
Startup Success Prediction using Machine Learning because it provides measurable output and clear classification results.

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Step 1: User opens web application
Step 2: Navigates to prediction page
Step 3: Enters startup-related inputs
Step 4: Clicks Predict button
Step 5: System processes data
Step 6: Results page displays prediction

3.2 Solution Requirement

Functional Requirements

- Accept startup input parameters
- Process inputs through trained ML model
- Display success prediction
- Handle input validation

Non-Functional Requirements

- Fast response time
- High prediction accuracy
- User-friendly interface
- Scalable design

3.3 Data Flow Diagram (Textual Representation)

User Input
↓
Frontend (Streamlit UI)
↓
Backend Processing
↓
Loaded Random Forest Model (.pkl)
↓
Prediction Output
↓
Result Display

3.4 Technology Stack

Programming Language: Python
Data Analysis: Pandas, NumPy
Visualization: Matplotlib, Seaborn
Machine Learning: Scikit-learn
Model Saving: Pickle
Development Platform: Google Colab
Frontend Framework: Streamlit
Version Control: GitHub

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

The project directly addresses the uncertainty faced by startups and investors by providing data-backed predictions. The dataset contains historical startup data that helps train a reliable classification model.

4.2 Proposed Solution

The system uses:

• Data preprocessing
• Feature selection
• Model training using Random Forest

• Model evaluation
• Integration into a web application

Random Forest was selected because it provided higher accuracy compared to Logistic Regression and Decision Tree during experimentation.

4.3 Solution Architecture

Layer 1: Data Layer
Dataset loading and preprocessing

Layer 2: Model Layer
Training and evaluation
Model saved as random_forest_model.pkl

Layer 3: Presentation Layer
Streamlit frontend
User interaction
Prediction display

---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase 1 – Dataset Collection
Downloaded startup dataset from Kaggle

Phase 2 – Data Preprocessing
Handled missing values
Encoded categorical variables
Split into training and testing sets

Phase 3 – Model Building
Trained multiple models
Selected Random Forest
Achieved strong accuracy

Phase 4 – Model Evaluation
Evaluated using accuracy score
Checked confusion matrix

Phase 5 – Deployment
Saved model as .pkl
Integrated with Streamlit
Pushed project to GitHub

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Evaluation Metrics Used:

- Accuracy Score
- Confusion Matrix
- Precision
- Recall

The Random Forest model achieved high classification accuracy on test data and showed balanced performance without overfitting.

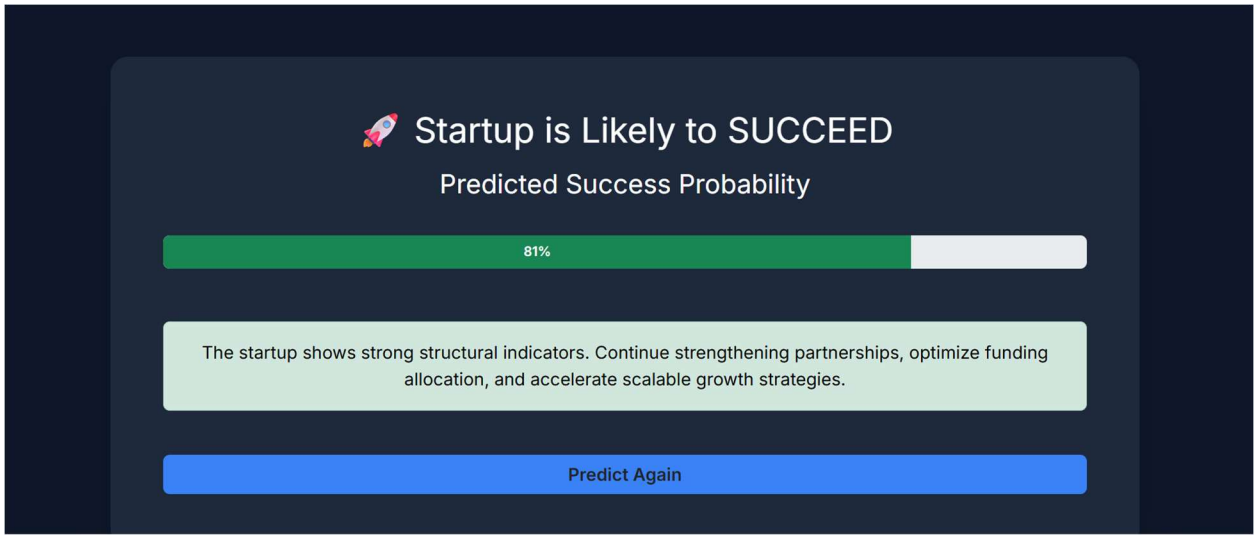
Functional Testing Included:

- Valid input testing
- Invalid input handling
- UI navigation testing
- Model loading verification

---

7. RESULTS

7.1 Output Screenshots



The system successfully displays:

- Home page
- Prediction input form
- Result page showing success or failure

The model generates predictions instantly after clicking the Predict button. The output is clearly displayed to the user.

---

8. ADVANTAGES & DISADVANTAGES

Advantages

- Data-driven decision support
- Easy-to-use web interface
- Scalable architecture
- Fast prediction time
- Demonstrates complete ML pipeline

Disadvantages

- Limited by dataset quality
- Predictions depend on historical trends
- Real-world business complexity not fully captured
- Model performance depends on feature quality

---

9. CONCLUSION

The Startup Success Prediction project successfully demonstrates the implementation of an end-to-end machine learning pipeline integrated with a web application. The system enables entrepreneurs and investors to make informed decisions based on predictive analytics rather than assumptions.

The project showcases practical skills in:

- Data preprocessing
- Machine learning model building
- Model evaluation
- Model deployment
- Web integration

The system meets its objectives and delivers reliable predictive performance.

---

10. FUTURE SCOPE

- Improve accuracy using XGBoost or Gradient Boosting
- Add probability score instead of binary output
- Deploy on cloud (AWS / Azure)
- Add analytics dashboard
- Integrate real-time economic indicators
- Convert into full SaaS product

---

11. APPENDIX

Source Code

Includes:

- Data preprocessing code
- Model training script
- Pickle model saving
- Streamlit app code

Dataset Link

Kaggle Startup Success Prediction Dataset :
Link: https://www.kaggle.com/datasets/manishkc06/startup-success-prediction

GitHub Repository

(https://github.com/harinagendra02/Startup-success-prediction-ml)

Project Demo Link

(https://drive.google.com/file/d/1lkFV_5WK9wiqASAYLjW0Sy_uUmfErSMs/view?usp=drive_link)