Prosperity Prognosticator

Machine Learning For Startup Success Prediction

Project Document – Full Reference

1. Executive Summary

The Prosperity Prognosticator is an end-to-end machine learning project that predicts the success or failure of startups using historical data. The system includes a complete ML pipeline — from raw data collection to a deployed Flask web application — and serves investors, entrepreneurs, and policymakers.

2. Project Objectives

- Develop a machine learning model with 80%+ accuracy for startup success prediction
- Compare multiple ML algorithms to select the best performer
- Apply hyperparameter tuning and feature selection for model optimization
- Deploy the trained model via a Flask web application
- Serve all three user scenarios: Investors, Entrepreneurs, Policy Makers

3. Dataset

Field	**Details**
Source	**Kaggle – Startup Success Prediction**
File Name	**startup_data.csv**
Format	**CSV (Comma-Separated Values)**
Target Column	**status (1 = success, 0 = failure)**
Key Features	**funding_total_usd, funding_rounds, age_first_funding_year, relationships, etc.**

4. ML Pipeline Summary

Stage	Key Activities	Output
Data Preparation	Load CSV, clean nulls, encode categoricals, split data	Cleaned X_train, X_test, y_train, y_test
EDA	Descriptive stats, class distribution, heatmap, pairplot	Visualizations and insights
Model Training	Train 6 algorithms, evaluate with accuracy + F1	Model comparison table
Tuning	GridSearchCV on Random Forest, select top 10 features	Optimized model
Deployment	Save .pkl, build Flask routes, create HTML templates	Working web application

5. Final Project Structure

File/Folder	Description
startup_data.csv	Raw dataset downloaded from Kaggle
startup-prediction-eda-model.ipynb	Jupyter Notebook with full ML pipeline
random_forest_model.pkl	Trained and saved Random Forest model
features.pkl	Saved list of top 10 selected features
app.py	Flask backend application with all routes
templates/index.html	Startup metrics input form
templates/result.html	Prediction result display page

6. How to Run the Project

- Step 1: Install requirements: pip install flask scikit-learn pandas numpy matplotlib seaborn

- Step 2: Open startup-prediction-eda-model.ipynb and run all cells to train and save the model

- Step 3: Ensure random_forest_model.pkl and features.pkl are generated in the project folder

- Step 4: Open terminal in the project folder and run: python app.py

- Step 5: Open browser and navigate to: http://127.0.0.1:5000

- Step 6: Enter startup metrics in the form and click Predict to see results

7. Results and Achievements

Achievement	Detail
Best Model	Random Forest Classifier
Model Accuracy	80%+ after hyperparameter tuning
Algorithms Compared	6 (LR, DT, RF, GB, SVM, KNN)
Features Selected	Top 10 from feature importance analysis
Web Application	Fully functional Flask app with 3 pages
Deployment	Localhost Flask server