Project Report Format

1. INTRODUCTION
   1.1 Project Overview

   Startups play a crucial role in driving innovation, employment, and economic growth. However, predicting whether a startup will succeed or fail is a complex task due to multiple influencing factors such as funding history, market conditions, business category, and operational strategy. Traditional evaluation methods rely heavily on intuition and manual analysis, which are often time-consuming and subjective. **Prosperity Prognosticator** is a machine learning–based solution designed to predict startup success using structured startup data. By leveraging classification algorithms such as Random Forest, this project provides an efficient, scalable, and data-driven approach for startup evaluation, helping investors and entrepreneurs make informed decisions.

   1.2 Purpose

   The core purpose of this project is to automate and improve startup success prediction using machine learning techniques.

   This project aims to:

   - Reduce dependency on manual startup evaluation.
   - Improve accuracy and consistency in predicting startup outcomes.
   - Enable quick and data-driven decision-making.
   - Provide a user-friendly web interface for real-time predictions.

   Ultimately, the project bridges artificial intelligence and business analytics to create an intelligent decision-support system that is practical, reliable, and scalable.

2. IDEATION PHASE
   2.1 Problem Statement
   2.2 Empathy Map Canvas
   2.3 Brainstorming
3. REQUIREMENT ANALYSIS
   3.1 Solution Requirement
   3.2 Data Flow Diagram
   3.3 Technology Stack
4. PROJECT DESIGN
   4.1 Problem Solution Fit
   4.2 Proposed Solution
   4.3 Solution Architecture
5. PROJECT PLANNING & SCHEDULING
   5.1 Project Planning
6. FUNCTIONAL AND PERFORMANCE TESTING
   6.1 Performance Testing

   Model Results (from your project):

   - Training Accuracy: **100%**
   - Testing Accuracy: **79%**
   - ROC-AUC Score: **0.745**
   - Precision-Recall AUC: **0.883**

   Classification Summary

   - Strong performance in predicting successful startups.
   - Slight overfitting observed due to high training accuracy.

7. RESULTS

7.1 Output Screenshots

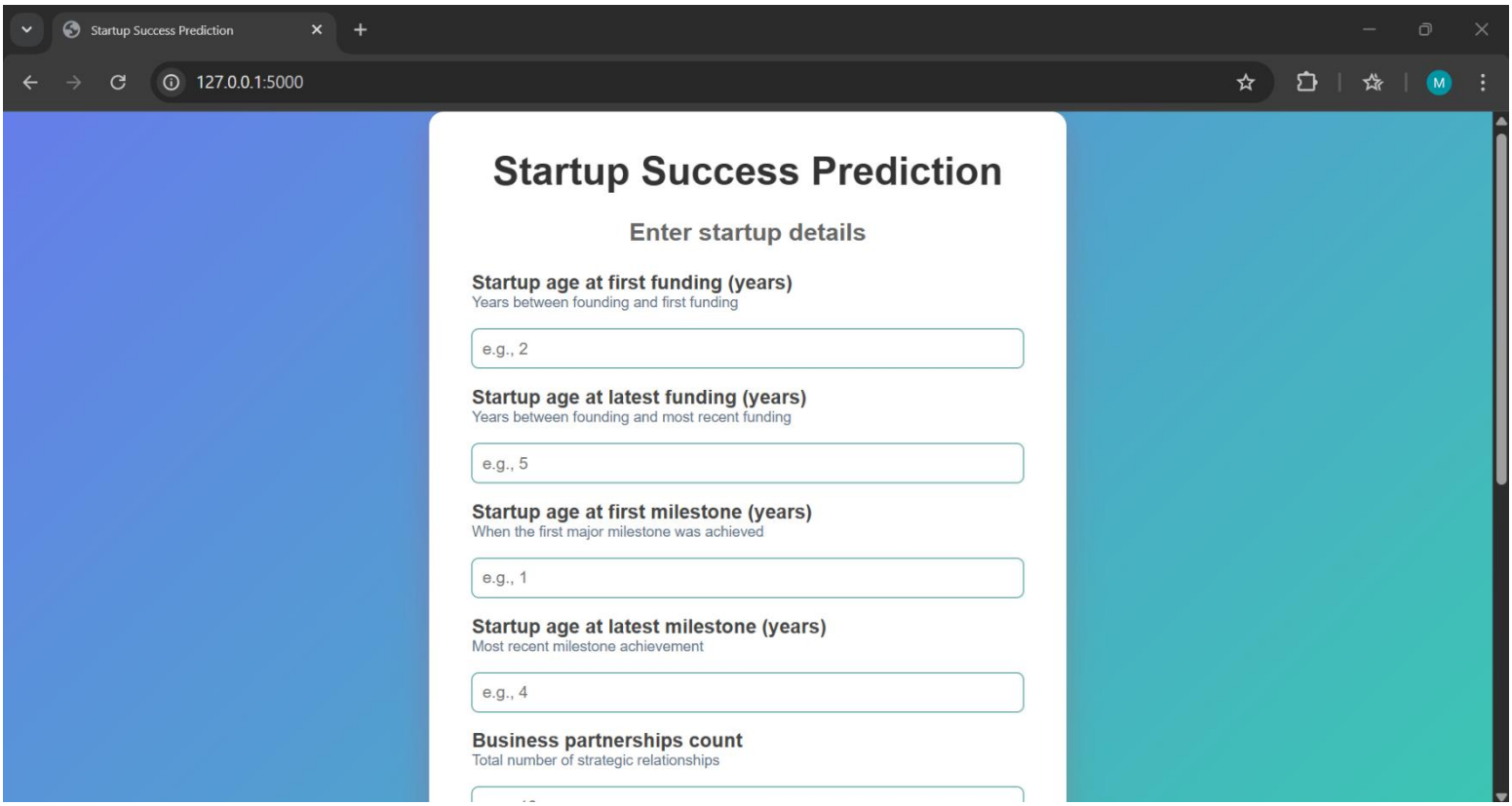


Figure 7.1

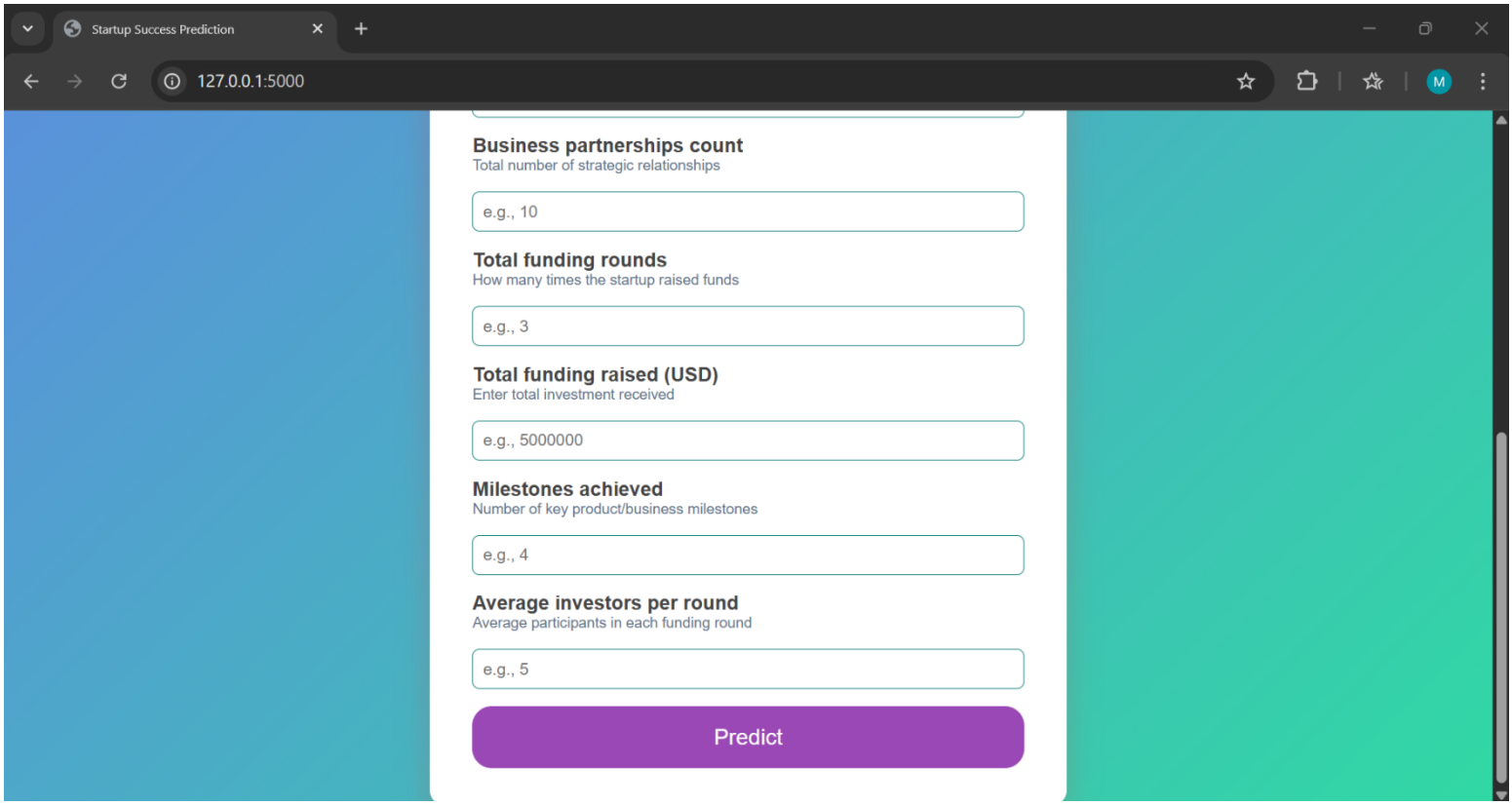


Figure 7.2

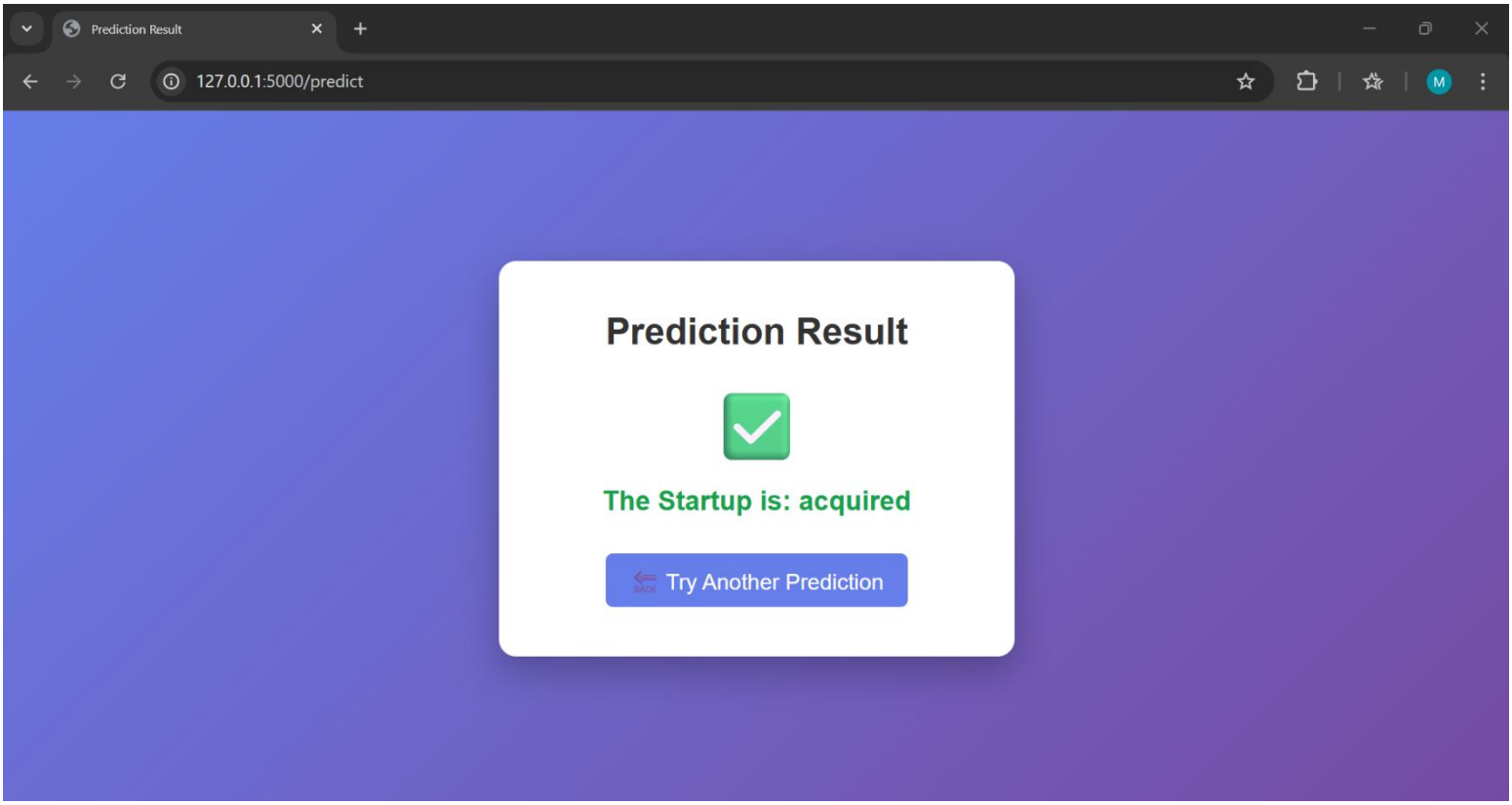


Figure 7.3

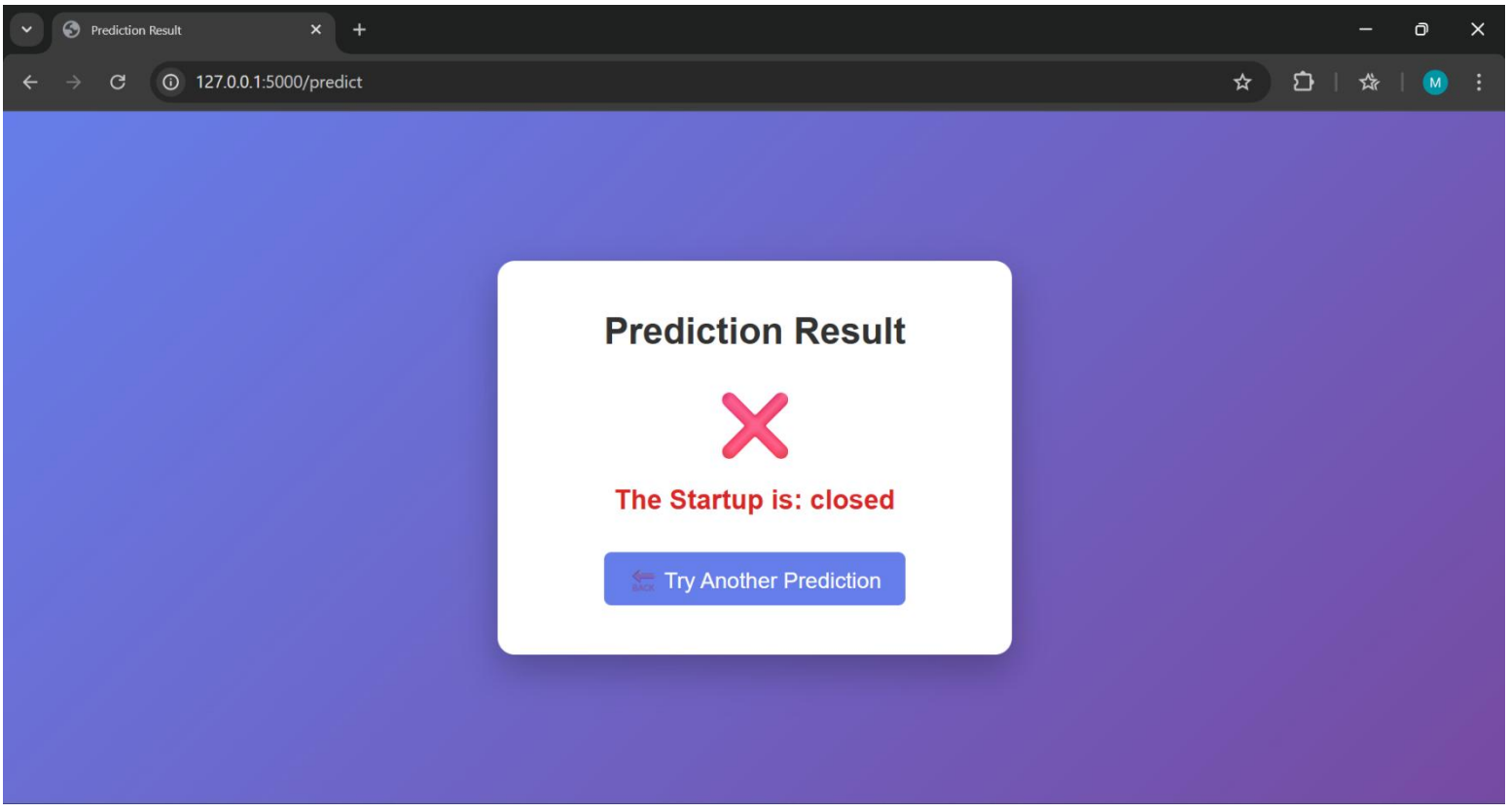


Figure 7.4

In [ ]:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()

rf.fit(X_train._get_numeric_data(),y_train)


y_pred_rf = rf.predict(X_test._get_numeric_data())

print("Training Accuracy :", rf.score(X_train._get_numeric_data(), y_train))
print("Testing Accuracy :", rf.score(X_test._get_numeric_data(), y_test))

cm = confusion_matrix(y_test, y_pred_rf)
plt.rcParams['figure.figsize'] = (3, 3)
sns.heatmap(cm, annot = True, cmap = 'YlGnBu', fmt = '.8g')
plt.show()

cr = classification_report(y_test, y_pred_rf)
print(cr)


print("----------------------------------------")

false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,y_pred_rf)
roc_auc = auc(false_positive_rate, true_positive_rate)
print("ROC Curves              =",roc_auc)

precision, recall, thresholds = precision_recall_curve(y_test, y_pred_rf)
f1 = f1_score(y_test, y_pred_rf)
Precision_Recall_rfs = auc(recall, precision)
print("Precision-Recall Curves =",Precision_Recall_rfs)
```

```
Training Accuracy : 1.0
Testing Accuracy : 0.7906137184115524
```

Figure 7.5

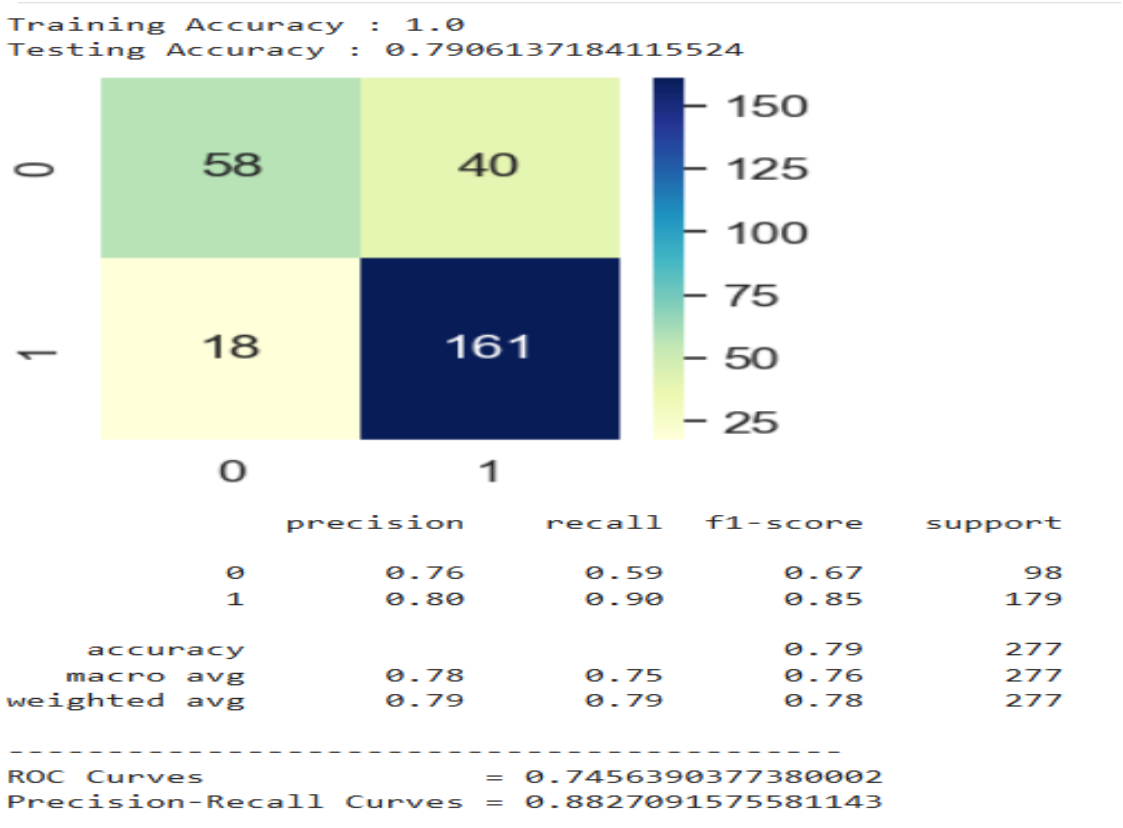


Figure 7.6

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

- Fast and data-driven decision support
- Easy-to-use web interface
- Scalable to larger startup datasets
- Reduces subjective investment decisions

8.2 Disadvantages

- Depends on dataset quality
- Slight overfitting observed
- Limited features may affect generalization

9. CONCLUSION

The project **Prosperity Prognosticator – Startup Success Prediction** demonstrates how machine learning can effectively support business decision-making. By using Random Forest classification, the system provides reliable predictions for startup success and failure. The solution bridges the gap between traditional analysis and AI-driven insights, enabling faster and smarter decisions for investors and entrepreneurs.

10. FUTURE SCOPE

- Integrate larger datasets for better accuracy
- Cloud deployment for public access
- Add user authentication
- Introduce advanced models like XGBoost or Neural Networks
- Provide downloadable analytical reports

11. APPENDIX

Source Code

This appendix contains key parts of the source code used in the project " **Prosperity Prognosticator – Startup Success Prediction”.** It includes the backend logic, model training script, and frontend form design.

A. app.py — Flask Backend

Handles:

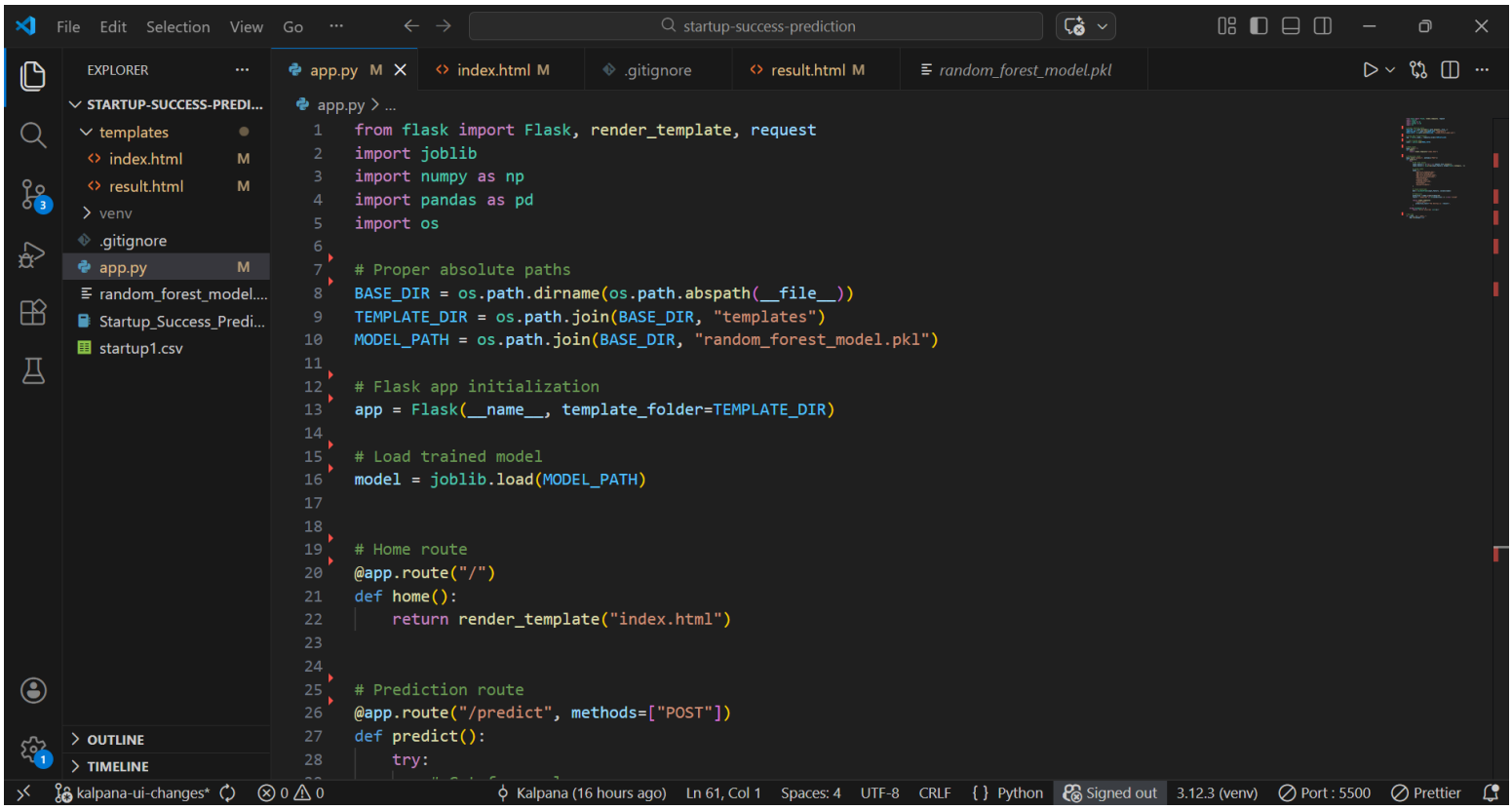
- Route definitions
- Form input
- Model loading
- Prediction and result rendering

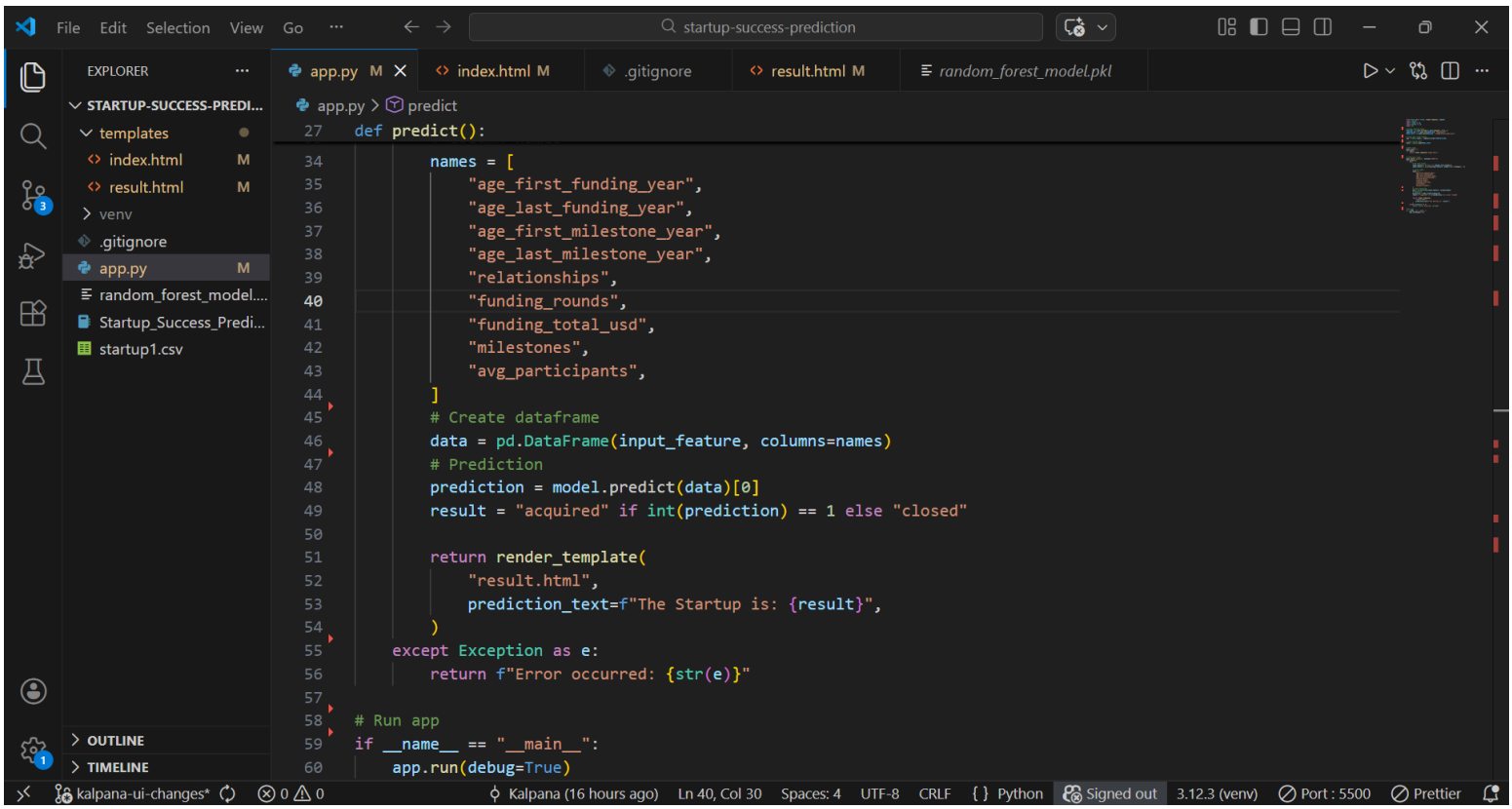
B. startup_model.ipynb — Model Training

Includes:

- **Data preprocessing**
- **Model building (Random Forest)**
- **Evaluation metrics**

- **Model saving**





Dataset Link: datsetlink

GitHub Link: https://github.com/Sowmyadiviti/startup-success-prediction

Project Demo Link: Videodemolink