Prosperity Prognosticator: Machine Learning for Startup Success Prediction

Project Description

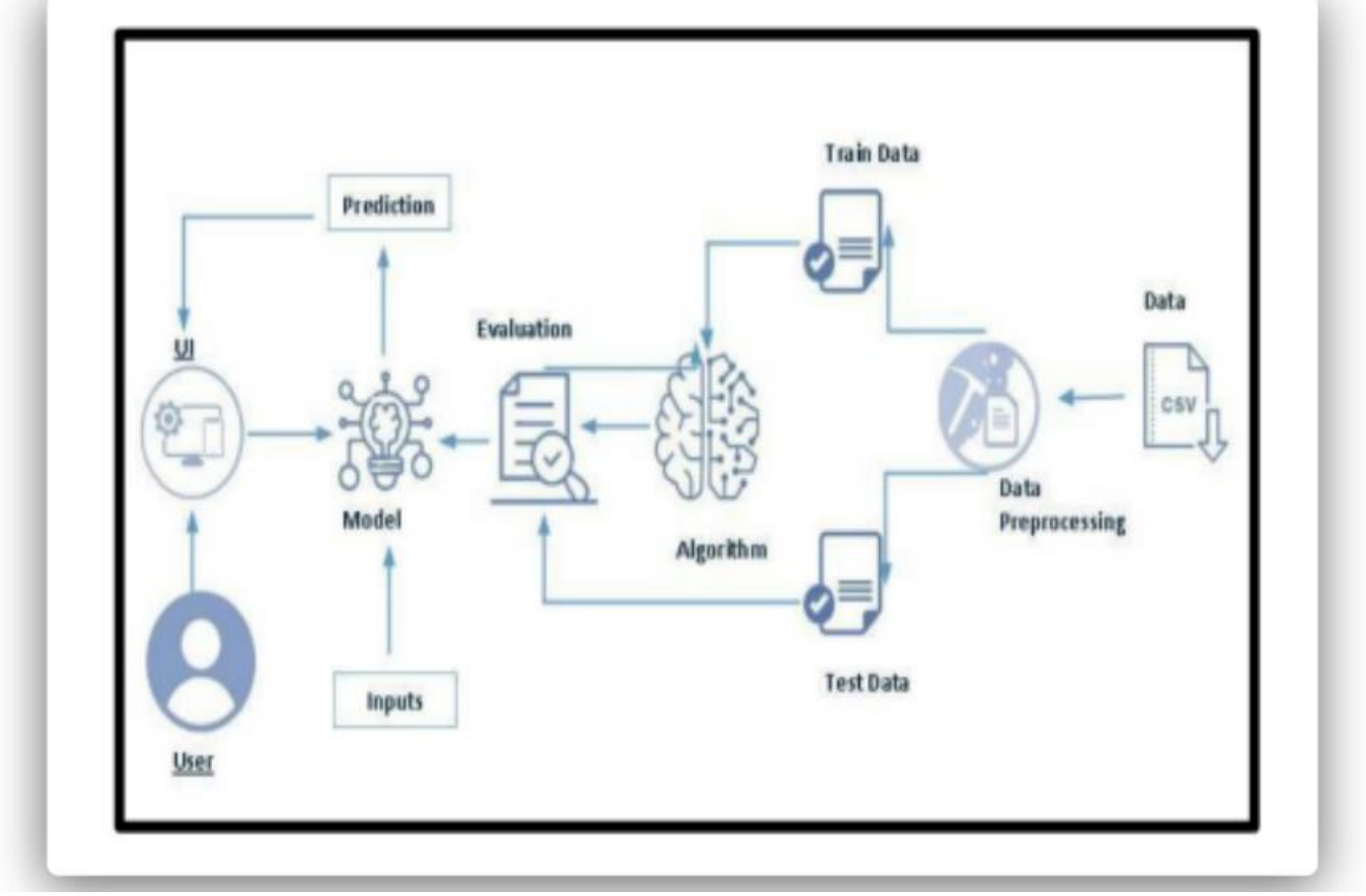
Develop a machine learning model for predicting the success of startups. By analyzing startup characteristics, market trends, and funding data, this project aims to provide insights into the potential success of new ventures, aiding investors, entrepreneurs, and policymakers in decision-making and resource allocation.

Scenario 1 (Investors): Utilize the prosperity prognosticator model to evaluate startup investment opportunities and assess the potential for returns. Improve investment decision-making, optimize portfolio management, and maximize investment profitability by identifying promising startups with high growth potential.

Scenario 2 (Entrepreneurs): Incorporate success prediction insights into business planning and strategy formulation for new ventures. Identify key success factors, mitigate risks, and enhance startup viability and sustainability through data-driven decision-making and resource allocation.

Scenario 3 (Policy Makers): Utilize machine learning predictions to inform entrepreneurship policies and support initiatives aimed at fostering startup growth and innovation. Identify factors influencing startup success, design targeted support programs, and stimulate economic development through the promotion of entrepreneurship and innovation ecosystems.

Technical Architecture:



To complete this project, you must required following software's, concepts and packages

- **Anaconda navigator and pycharm:**
  - Refer the link below to download anaconda navigator
  - Link : https://youtu.be/1ra4zH2G4o0
- **Python packages:**
- Open anaconda prompt as administrator
- Type "pip install numpy" and click enter.
- Type "pip install pandas" and click enter.
- Type "pip install scikit-learn" and click enter.
- Type "pip install matplotlib" and click enter.
- Type "pip install scipy" and click enter.
- Type "pip install pickle-mixin" and click enter.
- Type "pip install seaborn" and click enter.
- Type "pip install Flask" and click enter.

Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

- **ML Concepts**
  - Supervised learning: https://www.javatpoint.com/supervised-machine-learning
  - Unsupervised learning: https://www.javatpoint.com/unsupervised-machine-learning
  - Regression and classification
- Decision tree: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
- Random forest: https://www.javatpoint.com/machine-learning-random-forest-algorithm
- KNN: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
- Xgboost: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/
- Evaluation metrics: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/
- **Flask Basics** : https://www.youtube.com/watch?v=lj4I_CvBnt0

Project Objectives:

By the end of this project you will:

- Know fundamental concepts and techniques used for machine learning.
- Gain a broad understanding about data.
- Have knowledge on pre-processing the data/transformation techniques on outlier and some visualization concepts.

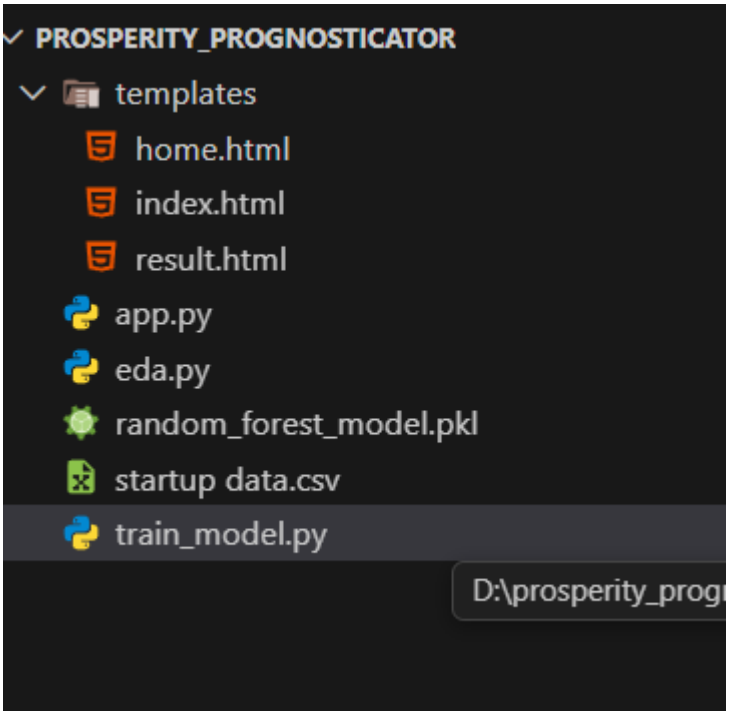
Project Flow:

- The user is shown the Home page. The user will browse through the Home page and go to predict my adaptivity and enter the specified engagement metrics.

- After clicking the Predict button the user will be directed to the Results page where the model will analyze the inputs given by the user and showcase the prediction of the Adaptivity level.
- To accomplish this we have to complete all the activities listed below:
    - Data Collection and Preparation
        - Collect the dataset
        - Data Preparation
    - Exploratory Data Analysis
        - Descriptive statistical
        - Visual Analysis
    - Model Building
        - Creating a function for evaluation
        - Training and testing the Models using multiple algorithms
        - Performance Testing & Hyperparameter Tuning
        - Testing model with multiple evaluation metrics
        - Comparing model accuracy before & after applying hyperparameter tuning
        - Comparing model accuracy for different numbers of features.
        - Building a model with appropriate features.
    - Model Deployment
        - Save the best model
        - Integrate with Web Framework

Project Structure:

- Create the Project folder which contains files as shown below



- We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
- random_forest_model.pkl is our saved model. Further we will use this model for flask integration.
- Training folder contains model training files and training.

Milestone 1: Data Collection

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

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

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

KLdkI/edit#gid=1373903219

Milestone 2: Visualizing and analysing the data

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

Note: There is n number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Activity 1: Importing the libraries

Import the necessary libraries as shown in the image.

```
import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
```

Activity 2: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

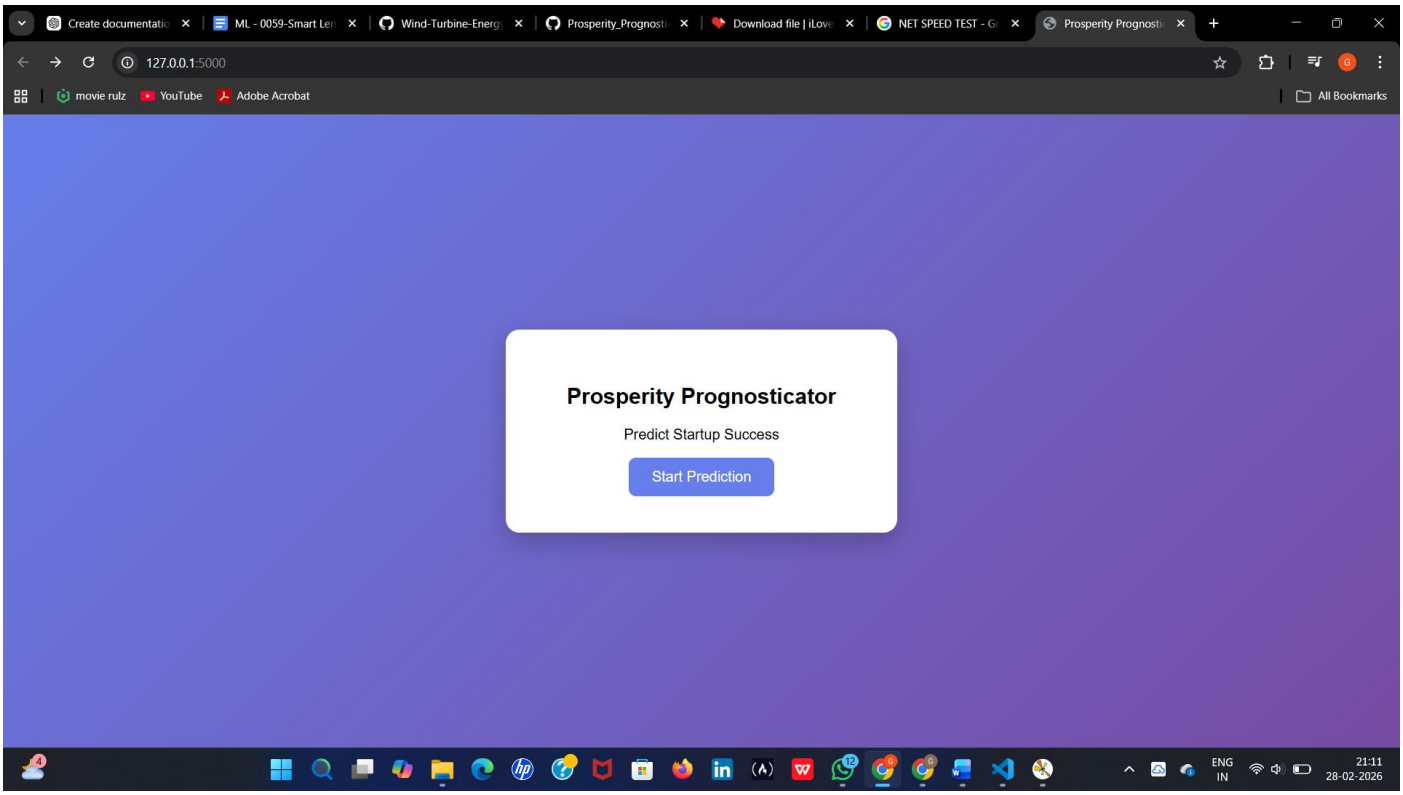
In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of csv file.

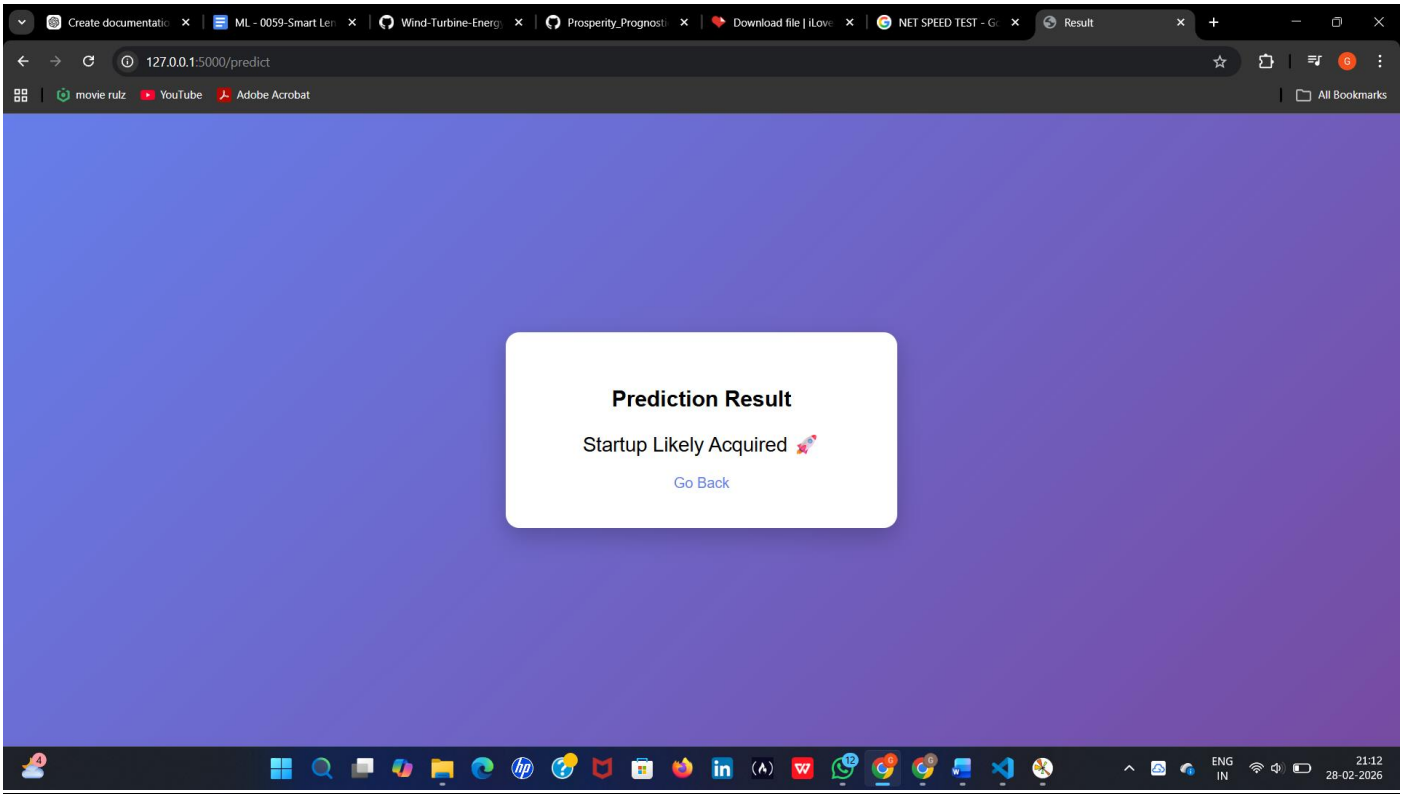
```
# ---------------- LOAD DATA ----------------
data = pd.read_csv('D:\\prosperity_prognosticator\\startup data.csv')


print("\nDataset Shape:", data.shape)
print("\nMissing Values:\n", data.isnull().sum())
```

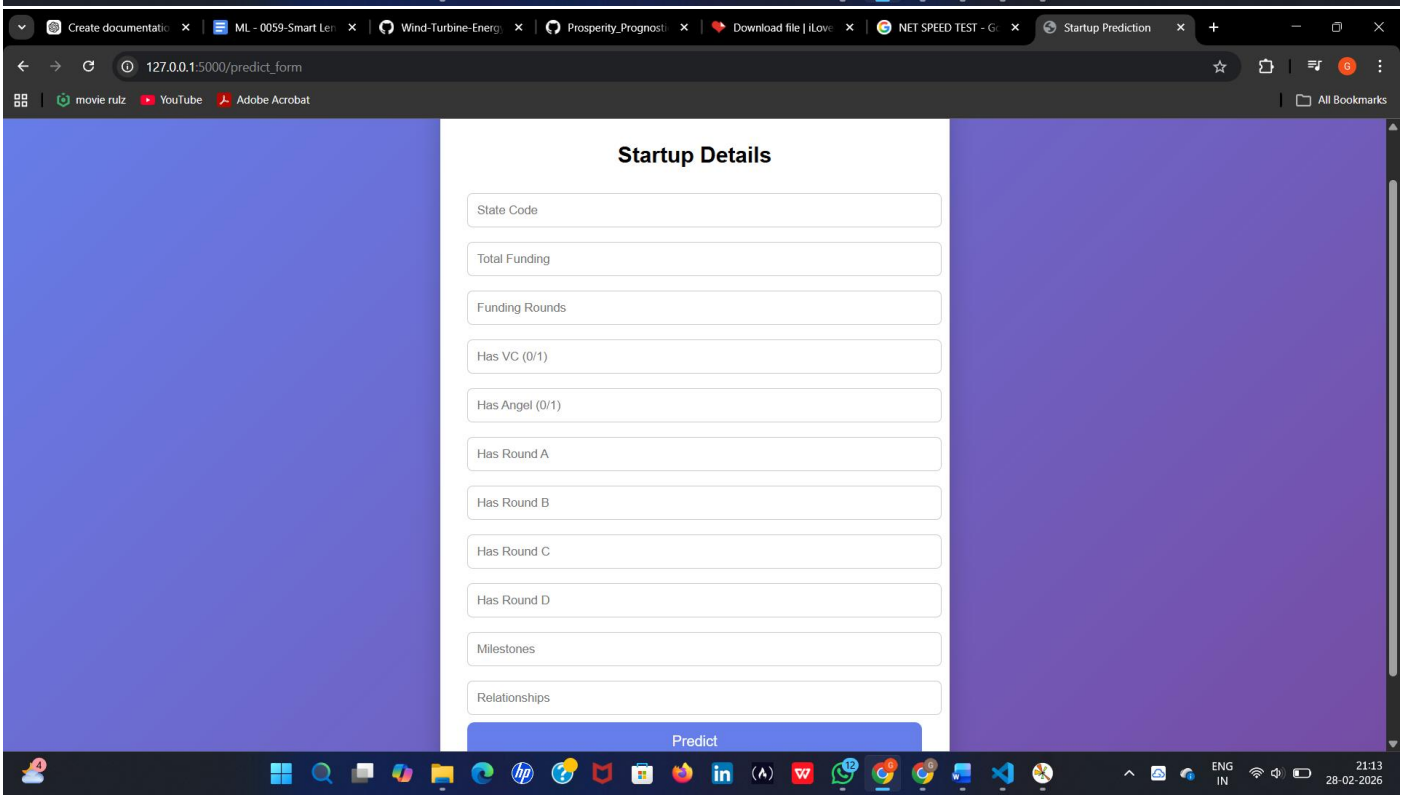
```
PS D:\prosperity_prognosticator> python train_model.py

Dataset Shape: (923, 49)

Missing Values:
 Unnamed: 0                0
state_code                 0
latitude                   0
longitude                  0
zip_code                   0
id                         0
city                       0
Unnamed: 6               493
name                       0
labels                     0
founded_at                 0
closed_at                588
first_funding_at           0
last_funding_at            0
age_first_funding_year     0
age_last_funding_year      0
age_first_milestone_year   152
age_last_milestone_year    152
relationships              0
funding_rounds             0
funding_total_usd          0
milestones                 0
state_code.1               1
is_CA                      0
is_NY                      0
is_MA                      0
is_
is_
category_code              0
is_software                0
is_web                     0
is_mobile                  0
is_enterprise              0
is_advertising             0
is_gamesvideo              0
is_ecommerce               0
```

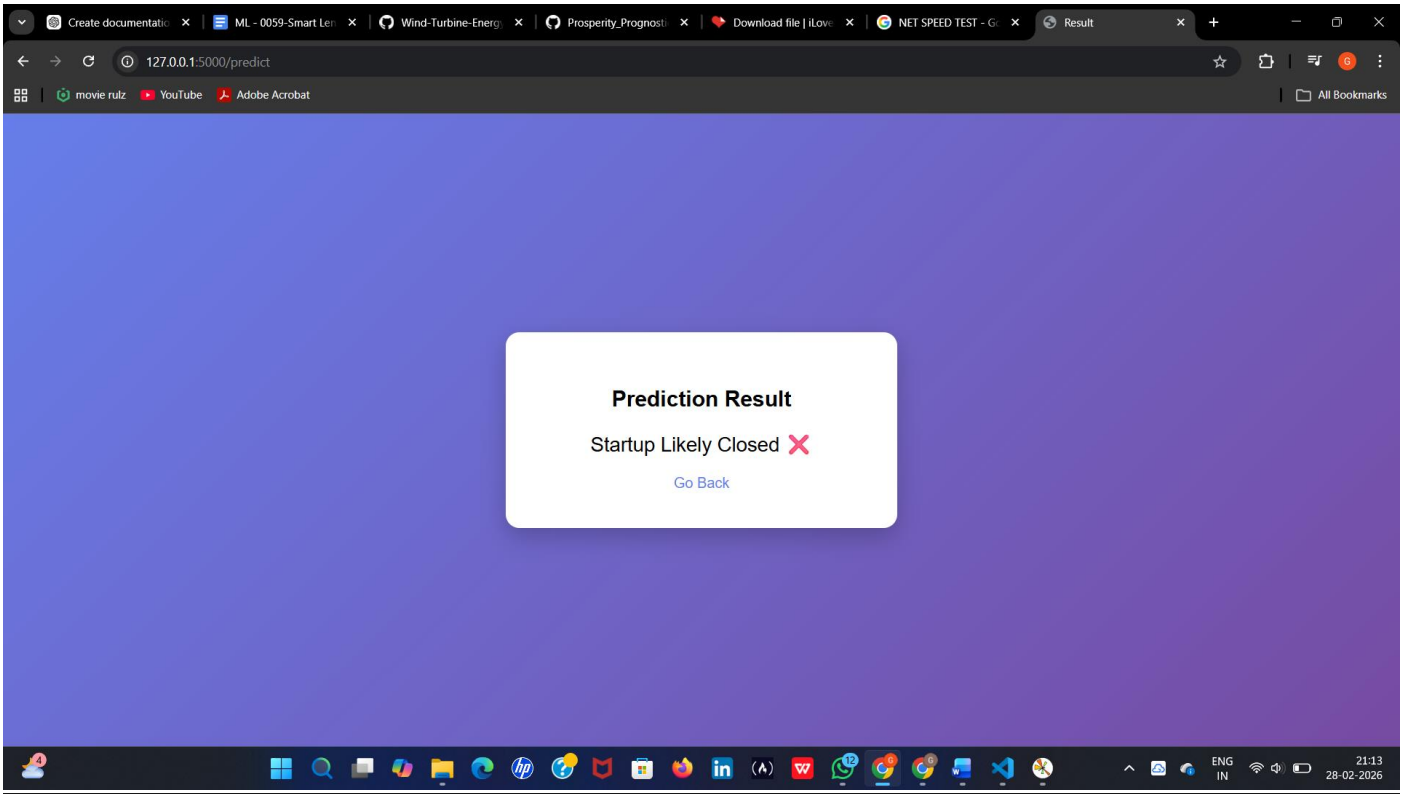
Focus folder in explorer (ctrl + click)

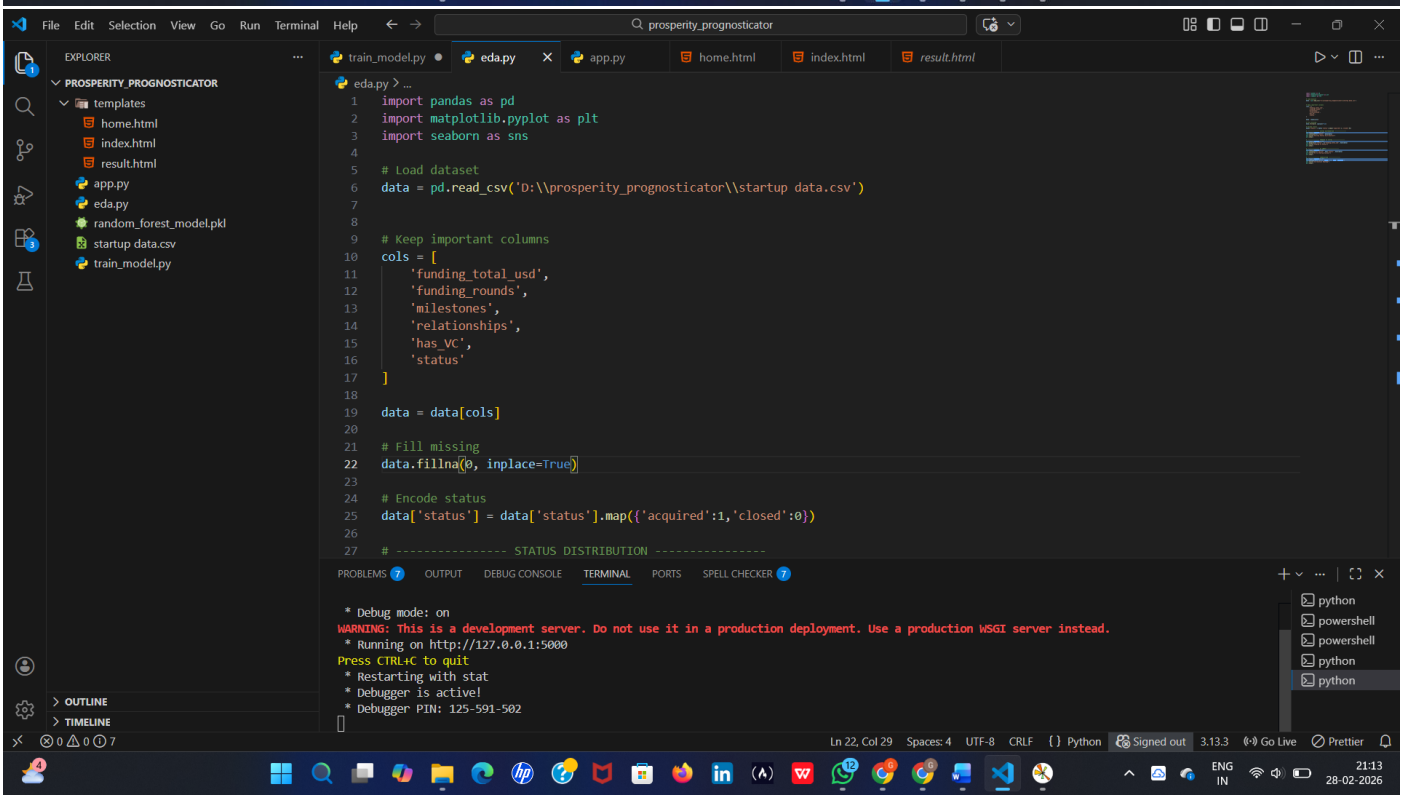
```python
# ---------------- MODEL ----------------
model = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    random_state=42
)

model.fit(X_train,y_train)

# ---------------- EVALUATION ----------------
train_acc = model.score(X_train,y_train)
test_acc = model.score(X_test,y_test)

print("\nTrain Accuracy:",train_acc)
print("Test Accuracy:",test_acc)

# ---------------- SAVE MODEL ----------------
joblib.dump(model,"random_forest_model.pkl")

print("\nModel Saved Successfully")
```

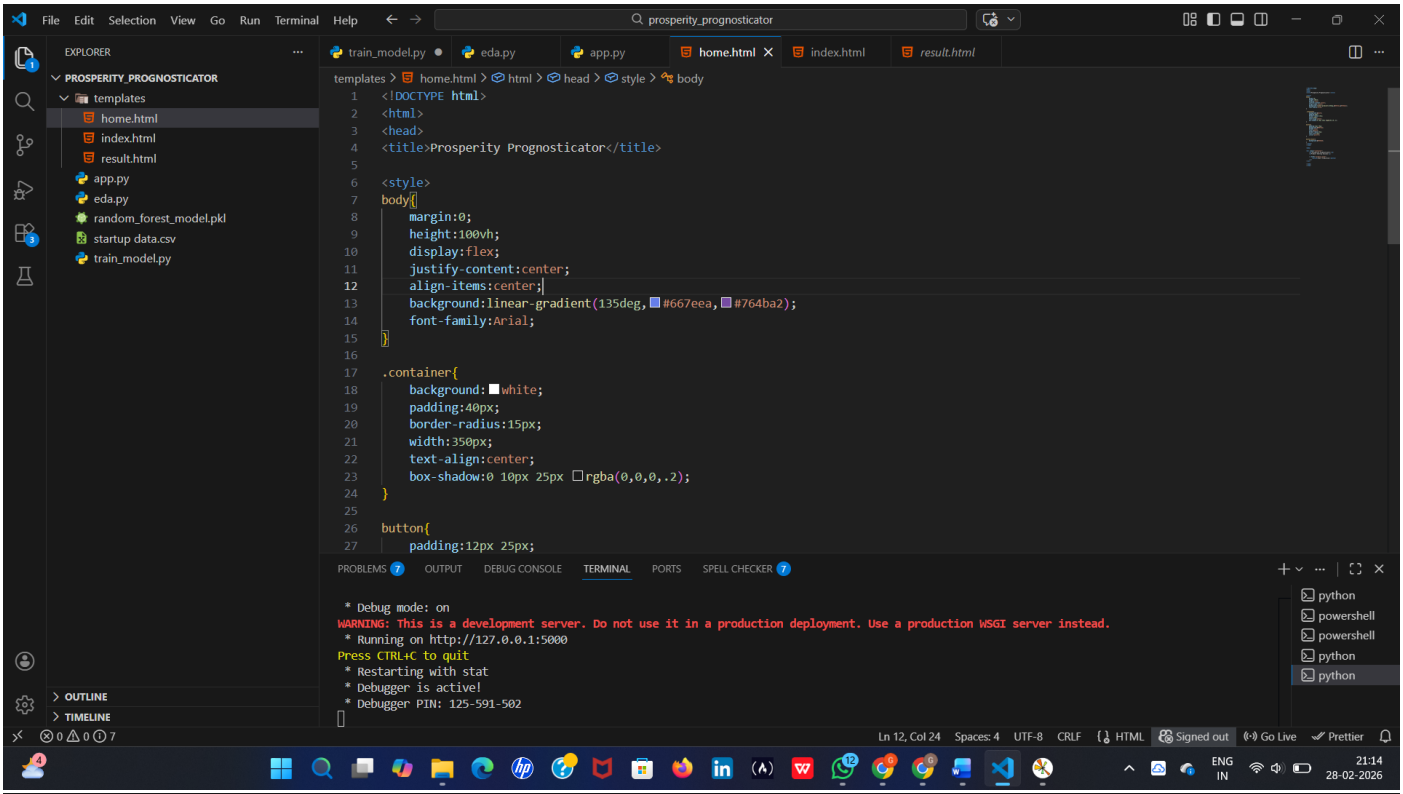


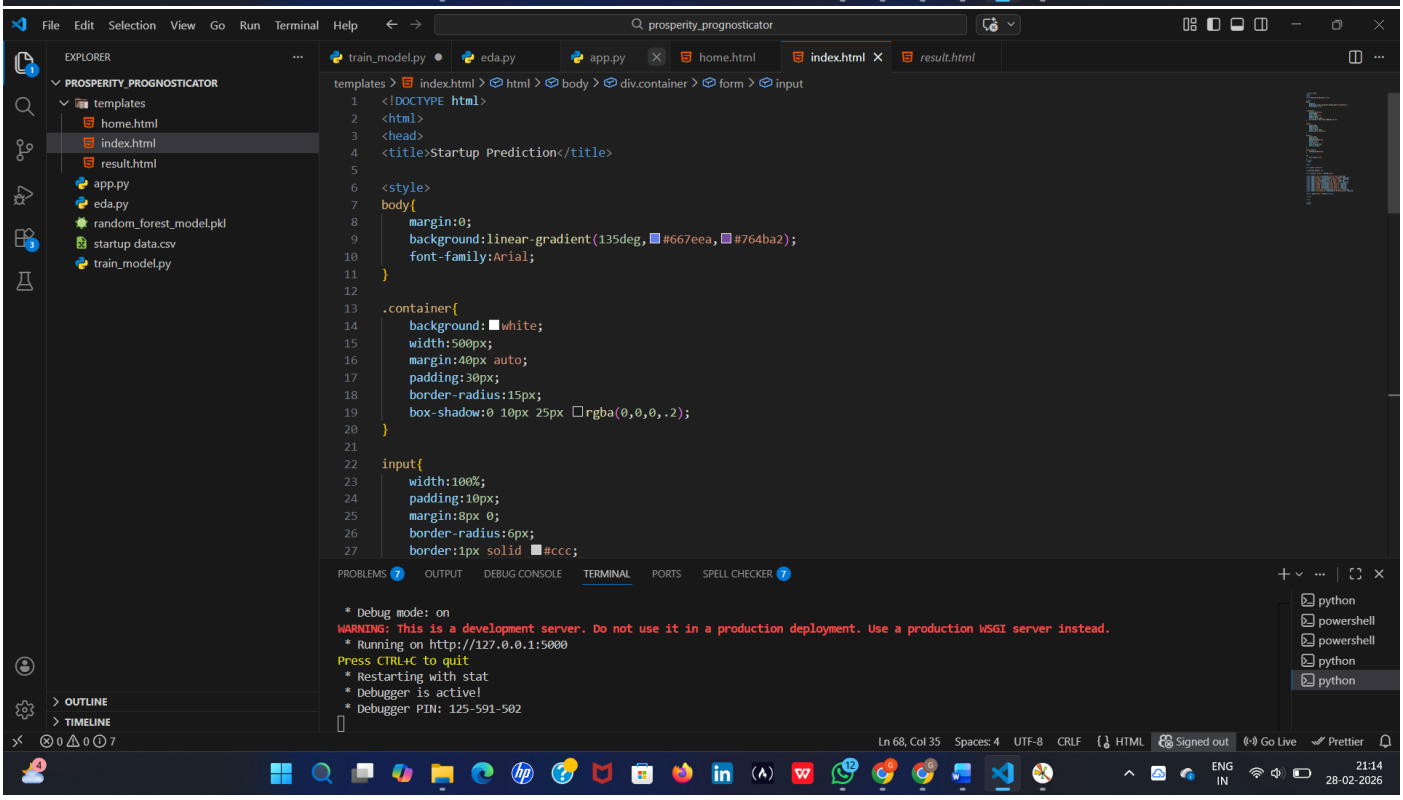

Result
127.0.0.1:5000/predict
movie rulz
YouTube
Adobe Acrobat
All Bookmarks
Prediction Result
Startup Likely Acquired
Go Back
ENG
IN
21:12
28-02-2026


Startup Prediction
127.0.0.1:5000/predict_form
movie rulz
YouTube
Adobe Acrobat
All Bookmarks
Startup Details
State Code
Total Funding
Funding Rounds
Has VC (0/1)
Has Angel (0/1)
Has Round A
Has Round B
Has Round C
Has Round D
Milestones
Relationships
Predict
ENG
IN
21:13
28-02-2026


Result
127.0.0.1:5000/predict
movie rulz
YouTube
Adobe Acrobat
All Bookmarks
Prediction Result
Startup Likely Closed
Go Back


File Edit Selection View Go Run Terminal Help
prosperity_prognosticator
EXPLORER
PROSPERITY_PROGNOSTICATOR
templates
home.html
index.html
result.html
app.py
eda.py
random_forest_model.pkl
startup data.csv
train_model.py
OUTLINE
TIMELINE
train_model.py
eda.py
app.py
home.html
index.html
result.html
eda.py > ...
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load dataset
data = pd.read_csv('D:\\prosperity_prognosticator\\startup data.csv')
# Keep important columns
cols = [
'funding_total_usd',
'funding_rounds',
'milestones',
'relationships',
'has_VC',
'status'
]
data = data[cols]
# Fill missing
data.fillna(0, inplace=True)
# Encode status
data['status'] = data['status'].map({'acquired':1,'closed':0})
---------- STATUS DISTRIBUTION ----------
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SPELL CHECKER 7
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 125-591-502
python
powershell
powershell
python
python
Ln 22, Col 29
Spaces: 4
UTF-8
CRLF
Python
Signed out
3.13.3
Go Live
Prettier


File Edit Selection View Go Run Terminal Help
prosperity_prognosticator
EXPLORER
PROSPERITY_PROGNOSTICATOR
templates
home.html
index.html
result.html
app.py
eda.py
random_forest_model.pkl
startup data.csv
train_model.py
train_model.py
eda.py
app.py
home.html
index.html
result.html
templates > home.html > html > head > style > body
<!DOCTYPE html>
<html>
<head>
<title>Prosperity Prognosticator</title>
<style>
body{
margin:0;
height:100vh;
display:flex;
justify-content:center;
align-items:center;
background:linear-gradient(135deg,#667eea,#764ba2);
font-family:Arial;
}
.container{
background:white;
padding:40px;
border-radius:15px;
width:350px;
text-align:center;
box-shadow:0 10px 25px rgba(0,0,0,.2);
}
button{
padding:12px 25px;
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SPELL CHECKER 7
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 125-591-502
python
powershell
powershell
python
python
OUTLINE
TIMELINE
Ln 12, Col 24
Spaces: 4
UTF-8
CRLF
HTML
Signed out
Go Live
Prettier
ENG
IN
21:14
28-02-2026


File Edit Selection View Go Run Terminal Help
prosperity_prognosticator
EXPLORER
PROSPERITY_PROGNOSTICATOR
templates
home.html
index.html
result.html
app.py
eda.py
random_forest_model.pkl
startup data.csv
train_model.py
train_model.py
eda.py
app.py
home.html
index.html
result.html
templates > index.html > html > body > div.container > form > input
<!DOCTYPE html>
<html>
<head>
<title>Startup Prediction</title>
<style>
body{
margin:0;
background:linear-gradient(135deg,#667eea,#764ba2);
font-family:Arial;
}
.container{
background:white;
width:500px;
margin:40px auto;
padding:30px;
border-radius:15px;
box-shadow:0 10px 25px rgba(0,0,0,.2);
}
input{
width:100%;
padding:10px;
margin:8px 0;
border-radius:6px;
border:1px solid #ccc;
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SPELL CHECKER 7
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 125-591-502
python
powershell
powershell
python
python
OUTLINE
TIMELINE
Ln 68, Col 35
Spaces: 4
UTF-8
CRLF
HTML
Signed out
Go Live
Prettier
ENG
IN
21:14
28-02-2026