# Risk Profiling Web Application

This repository contains a full-stack Flask web application that:

1. **Profiles user risk tolerance** based on survey inputs (Model 1).
2. **Classifies S&P 500 stocks by risk** (Model 2).
3. **Predicts future returns** and recommends top‑N stocks per risk bucket (Model 3).

All three models are packaged as pre‑trained joblib artifacts and integrated into a single web pipeline.

## Table of Contents

* [Features](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#features)
* [Project Structure](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#project-structure)
* [Installation & Setup](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#installation--setup)
* [Running the App Locally](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#running-the-app-locally)
* [Usage](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#usage)
  + [1. Risk Profiling Form](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#1-risk-profiling-form)
  + [2. Risk Prediction](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#2-risk-prediction)
  + [3. Simulation Dashboard](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#3-simulation-dashboard)
* [Model Artifacts](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#model-artifacts)
* [Inputs & Outputs](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#inputs--outputs)
* [Project Pipeline Flow](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#project-pipeline-flow)
* [Dependencies](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#dependencies)
* [License](https://chatgpt.com/c/68166885-7d4c-8003-8814-d2216472e4b1#license)

## Features

* **Step 1:** User enters financial and demographic data via a web form.
* **Step 2:** Backend preprocesses inputs and runs Model 1 to assign a risk bucket (Low/Medium/High).
* **Step 3:** User may proceed to a dashboard showing top‑N stock picks tailored to that bucket.
* **Step 4:** Models 2 & 3 handle stock‐risk classification and future‑return prediction.
* **Downloadable CSVs** available at each stage.

## Project Structure

Code/

├── app.py # Flask application entrypoint

├── train\_risk\_model.py # (Re)training script for Model 1

├── full\_pipeline.py # Standalone pipeline for Models 2 & 3

├── risk\_pipeline.joblib # Model 1 pipeline artifact

├── risk\_label\_encoder.joblib

├── stock\_classifier.joblib# Model 2 artifact

├── sp500\_features.csv # Universe features for Model 2

├── topreturn\_model.joblib # Model 3 artifact

├── top\_n\_per\_category.csv # Final pick list for dashboard

└── templates/ # HTML templates

├── index.html

├── predict.html

└── dashboard.html

Readme.docx # Original readme (superseded)

## Installation & Setup

1. **Clone the repo**
2. git clone <repo-url>
3. cd Code
4. **Create and activate a Python 3.8+ virtual environment**
   * Windows PowerShell:
   * python -m venv venv
   * .\venv\Scripts\Activate.ps1

**Note:** If activation is blocked by policy, run PowerShell as administrator and:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

* + macOS/Linux:
  + python3 -m venv venv
  + source venv/bin/activate

1. **Install dependencies**
2. pip install --upgrade pip
3. pip install -r requirements.txt

If you don’t have *requirements.txt*, run:

pip install flask pandas numpy scikit-learn joblib pandas\_datareader yfinance

## Running the App Locally

With your virtual environment active and dependencies installed:

python app.py

Navigate to: [http://localhost:5000](http://localhost:5000/)

## Usage

### 1. Risk Profiling Form

* Visit / and fill out all **19** fields.
* Each field name must match exactly (case & spacing) the feature names.

### 2. Risk Prediction

* Upon submission, /predict will display your assigned risk bucket.
* Uses risk\_pipeline.joblib (Model 1).

### 3. Simulation Dashboard

* Click **Proceed** to /dashboard.
* Displays top 5 stock recommendations for your bucket.
* Data sourced from top\_n\_per\_category.csv and topreturn\_model.joblib (Model 3).
* Download CSV via **Download CSV** link.

## Model Artifacts

1. **Model 1** (Risk Profiling)
   * risk\_pipeline.joblib
   * risk\_label\_encoder.joblib
2. **Model 2** (Stock Risk Classifier)
   * stock\_classifier.joblib
   * sp500\_features.csv
3. **Model 3** (Return Forecast & Picks)
   * topreturn\_model.joblib
   * top\_n\_per\_category.csv

## Inputs & Outputs

### Inputs (19 fields):

| **Feature** | **Type** | **Notes** |
| --- | --- | --- |
| Age Group | Numeric | 18–24, 25–34, … 65+ |
| Ethnicity | Categorical | Predefined categories |
| Education Level | Numeric | Highest completed education |
| Marital Status | Categorical | Married, Single, Divorced, Widowed |
| Financially dependent children | Integer | … |
| Annual Household Income | Numeric | Annual bracket or dollar amount |
| Spending vs Income Past Year | Categorical | Spend < |
| Difficulty covering expenses | Categorical | Very difficult … Not difficult |
| Emergency fund to cover 3 Months expenses | Categorical | None, 1 mo, 2 mo, 3+ mo |
| Current financial condition satisfaction | Numeric (1–5) | … |
| Thinking about FC frequency | Numeric (1–6) | Never → daily |
| Account ownership check | Integer | # of deposit/checking accounts |
| Savings/Money market/CD account ownership | Integer | # of savings-type accounts |
| Employer-sponsored retirement plan ownership | Integer | # of plans |
| Regular contribution to a retirement account | Binary (0/1) | Yes = 1, No = 0 |
| Non-retirement investments in stocks, bonds, mutual funds | Binary (0/1) | Yes/No |
| Homeownership | Binary (0/1) | Yes/No |
| Self-efficacy | Numeric (1–5) | Self-efficacy score |
| Self-rated overall financial knowledge | Numeric (1–5) | Literacy self-rating |

### Outputs:

* **Risk Profiling**: Low / Medium / High
* **Stock Picks**: table of top 5 tickers + predicted return

## Project Pipeline Flow

flowchart TD

A[Front-End Form] -->|POST form-data| B(API Server)

B --> C[Preprocessing]

C --> D[Risk Pipeline Prediction]

D --> E[Show Risk Level & Confirm]

E --> F[Load Picks CSV]

F --> G[Display Dashboard]

1. **Preprocessing**: coercion to numeric, median imputation, OHE with handle\_unknown='ignore'.
2. **Risk Prediction**: scikit-learn pipeline → numeric label → string via LabelEncoder.
3. **Picks**: precomputed CSV drives the dashboard; model‐3 artifact is saved as topreturn\_model.joblib.

## Dependencies

* **Python**: 3.8 or higher
* **Flask**: web framework
* **pandas**, **numpy**: data handling
* **scikit-learn**: model pipelines
* **joblib**: model serialization
* **pandas\_datareader**, **yfinance**: (for retraining and pipeline scripts)

Install via:

pip install -r requirements.txt

**requirements.txt** example:

flask>=2.0

pandas>=1.3

numpy>=1.20

scikit-learn>=1.0

joblib>=1.0

pandas\_datareader

yfinance

openpyxl

## License

This project is released under the **MIT License**.