**AD project: MoneyPort**

**Machine Learning**

Risk Assessment and Stock Recommendation

This project is Flask-based web application that assess and investor’s risk profile and recommends stocks based on that assessment.

**Project Structure**

traindataGenerator.py : to generate the features for training

risk\_assessment\_traintest.py: train-test using Decision Tree

app.py : Main Flask Application

Risk\_assessment\_model.pk1 : Pickled machine learning model for risk assessment

Label\_encoder.pk1 : Picked label encoder for risk levels

Populate\_Stock\_db.py: Script to populate the MYSQL database with stock data

Setup

Part 1:

Run traindataGenerator for generating the csv file (investment\_risk\_assessment.csv

)

Run risk\_assessment\_traintest to train the model using Decision Tree

Models will be created in a model folder:

1. risk\_assessment\_model.pkl
2. label\_encoder.pkl

1.install MySQL database :

**CREATE TABLE stocks(**

**Id INT AUTO\_INCREMENT PRIMARY KEY,**

**symbol VARCHAR(10) NOT NULL,**

**name VARCHAR(100) NOT NULL,**

**risk\_level ENUM(‘R1’,’R2’,’R3’) NOT NULL,**

**sector VARCHAR(50),**

**UNIQUE (symbol)**

**);**

Postman: the table features and query result

A screenshot of a computer

Description automatically generated

Browser:

A screenshot of a computer

Description automatically generated

2. Update the db\_config in both app.py and populate\_stock\_db.py with MySQL credentials

3. Run ‘populate\_stock\_db.py’ to populate the database with sample stock data

4. Ensure ‘risk\_assessment\_model.pkl’ and ‘label\_encoder.pkl’ are on the same directory as

app.py

Running the application

1. Run Flask application (app.py)
2. The application will start running on ‘http://localhost:8080’

Postman test result given below:

API endpoint

/predict\_and\_recommend (POST)

Input : JSON "investment\_goal",

    "time\_horizon",

    "risk\_tolerance",

    "income",

    "investment\_experience,

    "loss\_reaction",

    "age"

Output: JSON with risk\_level and recommended\_stocks

Test using postman

POST

<http://localhost:8080/predict_and_recommend>

Body

Raw

Json

{

    "investment\_goal": 3,

    "time\_horizon": 2,

    "risk\_tolerance": 4,

    "income": 2,

    "investment\_experience": 3,

    "loss\_reaction": 3,

    "age": 2

}

Result:[ will get 10 results ]

{

    "recommended\_stocks": [

        {

            "current\_price": 223.96,

            "name": "Apple Inc.",

            "sector": "Technology",

            "symbol": "AAPL"

        },

MySQL:

A screenshot of a computer

Description automatically generated

