

# **PORTFOLIO RISK ANALYSIS & PREDICTION**

**Prepared by: Prince Gupta**

# EXECUTIVE SUMMARY

**Goal:** Build automated portfolio risk & return analysis system.

Includes risk metrics, portfolio comparisons, stress tests & predictions.

Data-driven insights using BTC & ETH historical prices.

**End result:** Actionable insights for better investment decisions.

# PROBLEM STATEMENT

Investors lack quick insights into portfolio risk & performance.

Manual analysis of volatility, risk, and stress testing is time-consuming.

Need: Automated, data-driven, and easy-to-use solution.

# APPROACH & TOOLS

Python for calculations and automation.

Pandas, NumPy for data processing.

SQLite Database for storage of portfolio and risk metrics.

Matplotlib for graphs and visual insights.

SMTP email alerts for risk warnings.

# MILESTONE 1: DATA COLLECTION & SETUP

## Objective

Initialize the portfolio structure, assign weights, and store portfolio details in a database.

## Work Done

Created a **Python script (portfolio\_math.py)** that:

- Defines **currencies** (BTC and ETH)
- Assigns **weights** according to rule-based strategies:
  - **Equal-Weight Rule** (50%-50%)
  - **Risk-Based Rule** (lower risk → higher weight)
  - **Performance-Based Rule** (higher return → higher weight)

# MILESTONE 2: PORTFOLIO VS SINGLE ASSET

Compared returns of BTC, ETH, and portfolio.

Plotted line graph to visualize performance.

[Graph Placeholder: Portfolio vs BTC vs ETH returns over time]

Exported results to CSV for reporting.

## **Calculates:**

**Portfolio Return:** Weighted average of expected returns

**Portfolio Risk:** Standard deviation using covariance matrix

Implemented **SQLite database** to store:

- Portfolio details (portfolio table)
- Individual asset data (portfolio\_assets table)

## **Output**

Stored portfolio metrics in portfolio.db

Verified correct weights and calculations

Displayed “ Stored in DB” confirmation

# MILESTONE 2 INSIGHTS

## Objective

Enhance performance by running multiple rule tests in **parallel** and compare returns.

## Work Done

Added **ThreadPoolExecutor** to execute:

- Equal-weight
- Risk-based
- Performance-based rules **simultaneously**

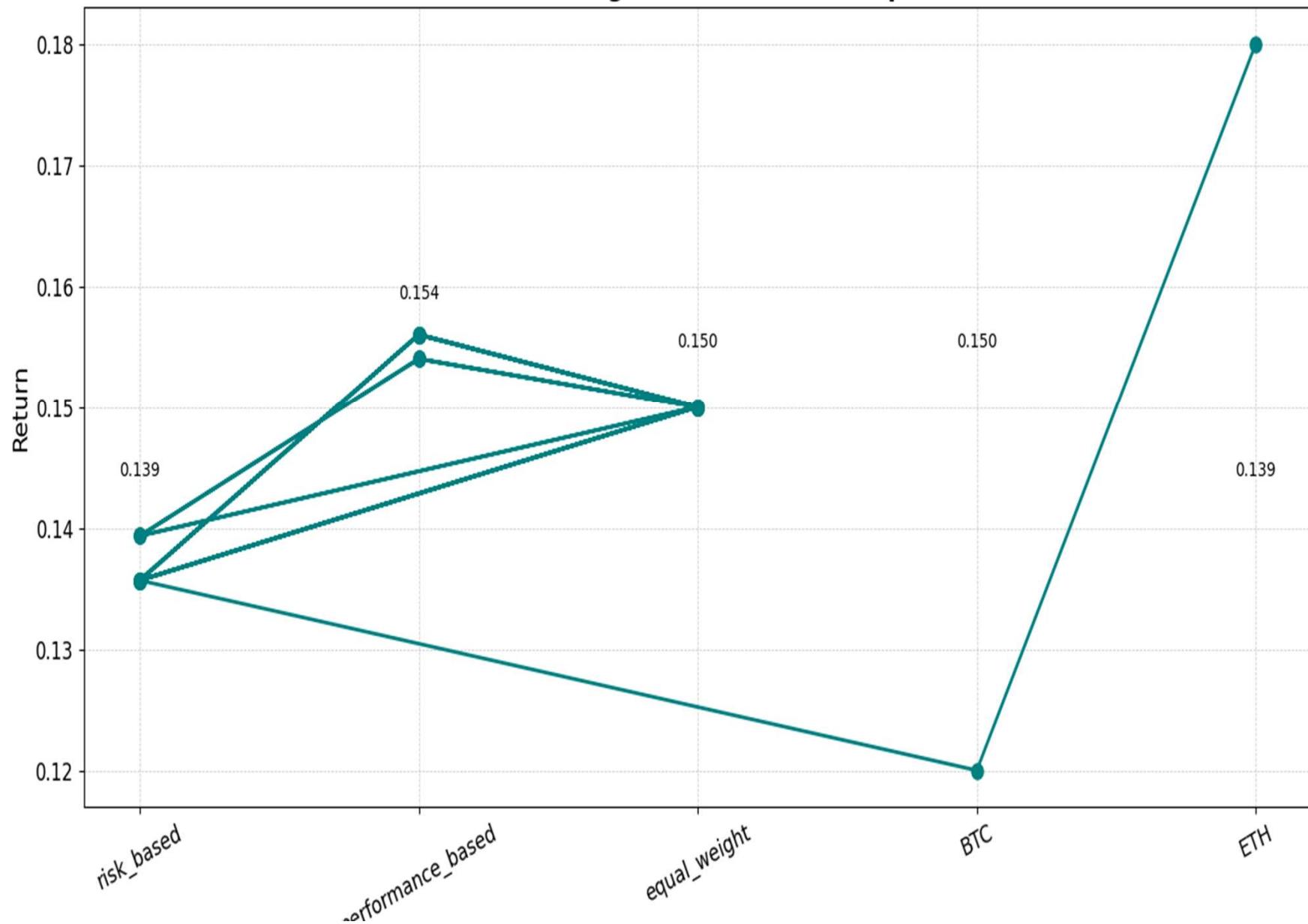
Stored each strategy result in the database.

Compared **portfolio vs. single asset returns** using real deterministic data.

Generated **line graph** :

- X-axis → Strategy/Asset Names
- Y-axis → Returns

## Portfolio vs Single Asset Return Comparison



## **Output**

Created and exported CSV file → portfolio\_vs\_assets.csv

Line graph showed:

- BTC and ETH individual returns
- Three portfolio strategies' returns

## **Insights:**

- ETH had higher returns but also more volatility.
- Balanced portfolios (risk-based) gave stable performance.

# MILESTONE 3 PART 1: RISK CHECKER

## 📍 Part 1: RiskChecker.py

### Objective:

Identify risk levels using specific financial risk metrics and send alert if any rule fails.

### Work Done:

Implemented **3 rules** (from 6 given options):

- **Volatility Risk**
- **Beta**
- **Asset Concentration**

Used **real BTC & ETH return data** (from CSV files).

# MILESTONE 3 PART 1:

Computed metrics and compared them with threshold limits.

Stored risk results in risk.db.

Sent **email alert** (SMTP-based) to client when any rule failed.

## **Output:**

Risk metrics with status (PASS/FAIL)

Database entries confirming checks

Email alert triggered for high-risk scenarios

# MILESTONE 3 PART 2: PREDICTOR

## **Objective:**

Predict future returns for portfolio and individual assets using a regression model.

## **Work Done:**

Loaded **historical BTC and ETH prices** from CSV.

Prepared training data ( $X$  = past returns,  $y$  = next-day returns).

Applied **Linear Regression** from scikit-learn.

Predicted **future returns** for both individual and portfolio assets.

## **Generated line chart:**

Real returns vs. Predicted returns

- Clear visualization of prediction accuracy.

# MILESTONE 3 PART 2: INSIGHTS

## **Output:**

Printed predicted returns on screen

Line chart displayed predicted trend

CSV export with predicted values

Option to extend with **Random Forest or LSTM** models

## **Insight:**

BTC showed smoother prediction than ETH due to lower volatility.

Portfolio predictions were more stable than single asset returns.

# MILESTONE 4: STRESS TESTING AND RISK PARITY RULE

## Objective

Simulate market stress and check how portfolio behaves under adverse conditions.

## Work Done

Implemented **Risk Parity Rule**:

- Assets weighted inversely to their volatility.

Conducted a **Stress Test** by simulating a sudden market drop (5–10%).

Calculated how portfolio return changed post-stress.

# MILESTONE 4:

Visualized stress test results with a **line chart** comparing:

- Normal scenario
- Stressed scenario

Stored results again in the database for traceability.

**Output:**

Chart clearly showed portfolio resilience to shocks.

Risk Parity portfolio performed best during stress conditions.

# KEY RESULTS & BUSINESS VALUE

Automated system saves analysis time and reduces manual effort.

Provides quick insights into risk and return for decision-making.

Helps investors adapt to market changes proactively.

# FUTURE INSIGHTS:

Integrate real-time data feeds.

Enhance prediction models with LSTM for accuracy.

Add interactive dashboards for portfolio visualization.

Expand to more assets beyond BTC & ETH.

**THANK  
YOU**