**Stock Portfolio and Sales Data Analysis**

**Team Members :**

1. Shrey Soni

2. Naisheel Patel

3. Suhani Patel

4. Dhyani Dave

5. Prince Patel

**Overview**

This application analyzes:

1. Stock portfolios for risk, return, and performance metrics.
2. Sales data for insights into total sales by item type.

It is built using Python libraries such as pandas, numpy, yfinance, matplotlib, and streamlit. The app allows users to input stock portfolios and upload sales data for analysis via an interactive interface.

**Features**

**1. Stock Portfolio Analysis**

* **Input**: Users provide a stock portfolio as ticker symbols and shares (e.g., AAPL:10, MSFT:5).
* **Metrics Calculated**:
  + Expected Annual Returns (per stock and portfolio)
  + Annualized Volatility (per stock and portfolio)
  + Sharpe Ratio (portfolio)
* **Risk-Return Visualization**: A scatterplot showing individual stock risk-return profiles and the overall portfolio performance.

**2. Sales Data Analysis**

* **Input**: Users upload a CSV file containing sales data.
* **Analysis Performed**:
  + Total sales aggregated by Item\_Type.
  + Visualization of sales distribution using a bar chart.
  + Tabular display of aggregated results.

**How to Use**

**Prerequisites**

Install the required Python packages:

pip install pandas numpy yfinance matplotlib streamlit

**Running the Application**

1. Save the script to a .py file (e.g., app.py).
2. Start the Streamlit application by running:
3. streamlit run app.py
4. Open the application in your browser (Streamlit will provide the link).

**Input Instructions**

**Stock Portfolio Analysis**

* Enter your portfolio in the following format:
* Ticker:Shares, e.g., AAPL:10, MSFT:5, GOOGL:8, AMZN:3
* Click the **Analyze Portfolio** button to view metrics and visualizations.

**Sales Data Analysis**

* Upload a sales data CSV file with at least the following columns:
  + Item\_Type
  + Item\_Outlet\_Sales
* View the analysis, which includes:
  + Bar chart of total sales by item type.
  + Tabular display of aggregated sales data.

**Code Details**

**Modules Used**

1. **pandas**: Data manipulation and analysis.
2. **numpy**: Numerical computations for portfolio analysis.
3. **yfinance**: Fetching historical stock data.
4. **matplotlib**: Risk-return scatterplot.
5. **streamlit**: Interactive user interface for inputs and outputs.

**Functions**

**1. analyze\_portfolio(portfolio)**

* Fetches historical stock data using yfinance.
* Calculates:
  + Daily returns.
  + Expected returns and volatility (annualized).
  + Portfolio return and volatility using weights.
  + Sharpe ratio (assuming a risk-free rate of 0.01).
* Returns:
  + Expected returns (per stock).
  + Volatility (per stock).
  + Portfolio metrics (return, volatility, Sharpe ratio).

**2. analyze\_sales\_data(sales\_data)**

* Aggregates total sales by Item\_Type.
* Returns:
  + Sorted sales totals.

**Notes**

**Portfolio Analysis Assumptions**

* Start Date: 2020-01-01
* End Date: 2023-01-01
* Risk-free rate for Sharpe Ratio: 0.01

**Known Issues**

* Ensure the portfolio input format is correct. Invalid inputs may cause errors.
* Uploaded sales data must include the required columns (Item\_Type, Item\_Outlet\_Sales).

**Example Usage**

**Portfolio Input**

AAPL:10, MSFT:5, GOOGL:8, AMZN:3

Expected output includes:

* Individual stock metrics (returns, volatility).
* Portfolio metrics (expected return, volatility, Sharpe ratio).
* Risk vs. Return scatterplot.

**Sales Data Input**

Example CSV structure:

| **Item\_Type** | **Item\_Outlet\_Sales** |
| --- | --- |
| Fruits | 12345.67 |
| Snacks | 9876.54 |

Output includes:

* Total sales bar chart.
* Tabular summary of total sales.