BMD5302 Financial Modelling for Fintech Professionals: RoboAdvisor Project

Group A04

Name	Matric ID
Huang Wenjie	A0297466J
Neil Heinrich Braun	A0298256M
Suhut Mickey Lin	A0298198A
Umar Bin Moiz	A0180913H
Zhou Yi	A0296533W

Table of Contents

1 Introduction **3**
2 Theoretical Framework **3**
2.1 Modern Portfolio Theory 3
2.2 Risk Aversion and Utility Function 4
3 Fund Selection and Analysis **4**
3.1 Fund Selection Criteria 4
3.2 Data Collection and Processing 6
4. Questionnaire Design **7**
4.1 Risk Aversion Assessment 7
4.2 Implementation and Scoring Algorithm 9
5 Portfolio Optimization **11**
5.1 Efficient Frontier Construction 11
5.2 Optimal Portfolio Selection **13**
6 Web Application Implementation **14**
6.1 Technology Stack 14
6.2 Setup Instructions 15
6.3 Application Architecture 15
6.4 Data Processing 16
6.5 User Interface Design 18
7 Backtesting **21**
8 Future Performance Simulation **21**
8.1 Monte Carlo Simulation 21
8.2 Visualization and Statistics 23
9 Error Handling and Robustness **23**
9.1 Backup Data generation 23
A. Purpose 23
B. Time Horizon and Frequency 24
C. Fund Classification and Parameterization 24
D. Output Format 24
E. Limitations 24
10 Results and Discussion **25**
10.1 Portfolio Performance 25
10.2 User Experience Testing 25
11 Limitations and Future Work **26**
11.1 Limitations 26
11.2 Future Enhancements 26
12 Conclusion **27**
13 References **27**

1 Introduction

This report presents our Financial Robot Advisor project developed for the BMD5302 Financial Modeling course. The robot advisor is designed to help investors determine their risk profile and recommend an optimized portfolio of funds that best matches their investment goals and risk tolerance.

The application is built using Modern Portfolio Theory (MPT) combined with a questionnaire-based approach to assess risk aversion. It provides a comprehensive investment solution that includes:

- Risk profile assessment through a questionnaire
- Portfolio optimization based on the investor's risk aversion
- Visualization of the efficient frontier
- Simulation of future portfolio performance

Our solution takes a quantitative approach to investment management while making it accessible to users without specialized financial knowledge. The web-based platform was developed using Python and Streamlit, enabling an interactive user experience.

2 Theoretical Framework

2.1 Modern Portfolio Theory

Modern Portfolio Theory (MPT), developed by Harry Markowitz in 1952, forms the theoretical foundation of our robot advisor. MPT suggests that rational investors should maximize expected returns for a given level of risk, or minimize risk for a given level of expected return.

The key principles of MPT implemented in our system include:

- Risk and return characteristics of individual assets
- Diversification benefits through covariance analysis
- The concept of the efficient frontier
- The global minimum variance portfolio (GMVP)
- The tangency portfolio (maximum Sharpe ratio)

2.2 Risk Aversion and Utility Function

Our robot advisor uses a utility function to model investor preferences:

$$U = r - \sigma^2 * \frac{A}{2} \qquad (1)$$

Where:

- U is the utility value to be maximized
- r is the expected return of the portfolio
- σ is the standard deviation (risk) of the portfolio
- A is the risk aversion coefficient derived from the questionnaire

The risk aversion coefficient A represents how much an investor values risk reduction relative to return enhancement. A higher risk aversion value indicates a more conservative investor who prefers lower risk investments, while a lower value indicates an investor more willing to accept risk for potential higher returns.

3 Fund Selection and Analysis

3.1 Fund Selection Criteria

Our fund selection process was guided by several key criteria to ensure a well-diversified investment universe covering various asset classes, geographic regions, and risk profiles. We selected 15 funds from FundSupermart that collectively provide exposure to the full spectrum of the risk-return landscape, allowing for effective portfolio construction regardless of investor risk tolerance.

The primary criteria used in our selection process were:

- Asset Class Diversity: We included funds from all major asset classes to ensure comprehensive coverage:
  - Cash and money market funds (Fullerton SGD Cash Fund, Fidelity US Dollar Cash A)
  - Fixed income funds spanning government, corporate, and high-yield bonds
  - Equity funds covering developed and emerging markets
  - Mixed-asset/balanced funds (FTIF - Franklin Income, Allianz Income and Growth)
- Geographic Diversification: Funds were selected to provide exposure to:
  - North America (US equity and bond funds)
  - Europe (European high yield, UK equity)
  - Asia (Singapore dividend equity, Eastspring Dragon Peacock)
  - Global investments (several funds have global mandates)
- Currency Exposure: We included funds denominated in various currencies:
  - Singapore Dollar (SGD) as the base currency for local investors
  - US Dollar (USD) for international exposure

    - British Pound (GBP) for further diversification
    - Currency-hedged options to manage foreign exchange risk
- Risk Spectrum Coverage: Funds were carefully selected to cover the entire risk spectrum:
    - Low-risk options (cash funds, government bonds)
    - Medium-risk options (corporate bonds, balanced funds)
    - Higher-risk options (equity funds, particularly sector-specific ones like technology)
- Fund Management Quality: We prioritized funds from reputable asset managers with:
    - Established track records (minimum 3-year history where possible)
    - Reasonable expense ratios
    - Consistent management teams
    - Clear investment processes

The specific funds selected include:

1. Cash/Money Market:
    a. Fullerton SGD Cash Fund - Low-risk SGD money market fund
    b. Fidelity US Dollar Cash A - USD cash fund for liquidity
2. Fixed Income:
    a. GS FUNDS III - US DOLLAR CREDIT P CAP USD - Investment-grade US corporate bonds
    b. JPMORGAN FUNDS - US AGGREGATE BOND A (ACC) SGD-H - Hedged US aggregate bonds
    c. FIDELITY EUROPEAN HIGH YIELD A-MDIST-SGD - European high-yield exposure
    d. FIDELITY US HIGH YIELD A-MDIST-SGD - US high-yield bonds
3. Mixed/Balanced:
    a. FTIF - FRANKLIN INCOME A MDIS SGD-H1 - Income-focused balanced fund
    b. ALLIANZ INCOME AND GROWTH CL AM DIS H2-SGD - Growth and income balanced fund
4. Equity:
    a. ALLIANZ ORIENTAL INCOME ET ACC SGD - Asian equity income
    b. NIKKO AM SINGAPORE DIVIDEND EQUITY ACC USD - Singapore dividend focus
    c. CT UK EQUITY INCOME CLASS 1 ACC GBP - UK equity exposure
    d. EASTSPRING INVESTMENTS UNIT TRUSTS - DRAGON PEACOCK A SGD - China and India exposure
    e. JPMORGAN FUNDS - US SELECT EQUITY PLUS A (ACC) SGD - US broad market
    f. JPMORGAN FUNDS - US TECHNOLOGY A (ACC) SGD - US technology sector for growth
5. Other:
    a. East Spring Investment Unit Trust - Broad diversifier

This carefully curated selection provides our optimization algorithm with sufficient diversity to construct portfolios across the entire efficient frontier, from the most conservative to the most aggressive risk profiles.

3.2 Data Collection and Processing

For this project, we manually collected historical **NAV (Net Asset Value)** data for each fund from the FundSupermart platform, as specified in the project requirements. The data was downloaded directly from:

https://secure.fundsupermart.com/fsmone/tools/fund-selector

Data Collection Process

1. Searching for each fund on the FundSupermart platform
2. Downloading the historical NAV data in Excel format
3. Cleaning and standardizing the data format
4. Storing the data in the project directory for analysis

Dataset

Our dataset includes the following **15 funds**:

1. Fullerton SGD Cash Fund
2. East Spring Investment Unit Trust
3. Fidelity US Dollar Cash A
4. GS FUNDS III - US DOLLAR CREDIT P CAP USD
5. JPMORGAN FUNDS - US AGGREGATE BOND A (ACC) SGD-H
6. FIDELITY EUROPEAN HIGH YIELD A-MDIST-SGD
7. FIDELITY US HIGH YIELD A-MDIST-SGD
8. FTIF - FRANKLIN INCOME A MDIS SGD-H1

9. ALLIANZ INCOME AND GROWTH CL AM DIS H2-SGD
10. ALLIANZ ORIENTAL INCOME ET ACC SGD
11. NIKKO AM SINGAPORE DIVIDEND EQUITY ACC USD
12. CT UK EQUITY INCOME CLASS 1 ACC GBP
13. EASTSPRING INVESTMENTS UNIT TRUSTS - DRAGON PEACOCK A SGD
14. JPMORGAN FUNDS - US SELECT EQUITY PLUS A (ACC) SGD
15. JPMORGAN FUNDS - US TECHNOLOGY A (ACC) SGD

Data Preprocessing Tasks

- Converting text-based or incorrectly formatted NAV values to numeric format
- Handling missing values through forward and backward filling methods
- Identifying and resolving duplicate date entries by keeping the most recent value
- Aligning all fund data onto a common date range for covariance calculation

4. Questionnaire Design

4.1 Risk Aversion Assessment

The design of our risk aversion questionnaire is grounded in established financial and behavioral economics research on risk tolerance assessment. We developed a systematic approach to evaluating an investor's risk aversion level by addressing multiple dimensions that influence risk-taking behavior in financial decisions.

Our questionnaire design was guided by the following theoretical considerations:

- Life-cycle Hypothesis: Research by Modigliani and Brumberg suggests that risk tolerance varies with age and investment horizon. Younger investors typically have more human capital (future earning potential) and time to recover from market downturns, justifying greater risk tolerance.

- Wealth Effect on Risk Aversion: Economic theory suggests that absolute risk aversion tends to decrease as wealth increases. Higher net worth individuals can typically bear more financial risk without compromising their standard of living.

- Behavioral Biases: We accounted for subjective self-assessment of risk tolerance, which research shows is influenced by past experiences, financial literacy, and psychological factors.

- Gender Differences: Studies indicate potential differences in risk preferences between genders, although we assigned relatively small weight to this factor to avoid excessive generalization.

- Time Horizon Effects: Longer investment horizons typically allow for greater risk-taking due to mean reversion in markets and the ability to withstand short-term volatility.

Based on these considerations, we developed seven questions that collectively provide a comprehensive assessment of an investor's risk aversion. Each question targets a specific aspect of risk profile determination:

1. Investment Purpose (For self or others): This establishes context for the investment decision, though we assign no direct impact on the risk score as fiduciary responsibilities can vary widely.

2. Age: This question captures the life-cycle effect on risk tolerance. Younger investors generally have more time to recover from market downturns and more future earning potential (human capital), justifying a lower risk aversion score.

3. Gender: While recognizing that individual differences are more important than group tendencies, we incorporated a small adjustment based on statistical differences in average risk preferences observed in research studies.

4. Geographic Location: This provides context but does not directly impact the risk score. Different regions may have different investment cultures and regulatory environments, but we opted not to make assumptions based solely on location.

5. Investment Amount: This question addresses the wealth effect on risk aversion. Larger investment amounts (relative to total wealth) typically allow for more risk-taking, as the marginal utility of wealth decreases with larger amounts.

6. Investment Time Horizon: Longer investment horizons generally support higher risk tolerance due to the ability to weather market volatility and benefit from long-term market growth.

7. Self-reported Risk Tolerance: Direct self-assessment of risk preferences provides valuable subjective input and captures psychological factors not easily measured through objective questions.

The weighting system for these questions was calibrated to reflect their relative importance in determining overall risk aversion:

- Age contributes up to 3 points (30% of maximum score)
- Investment amount contributes up to 2 points (20% of maximum score)
- Investment horizon contributes up to 3 points (30% of maximum score)
- Self-reported risk tolerance contributes up to 2 points (20% of maximum score)
- Gender contributes a minor adjustment (up to 0.5 points)

The final risk aversion score is normalized to a scale of 1–10, where 1 represents extremely low risk aversion (high risk tolerance) and 10 represents extremely high risk aversion (low risk tolerance). This score is then directly used as the risk aversion coefficient (A) in our utility function.

Our approach balances objective factors (age, investment amount, time horizon) with subjective self-assessment, providing a comprehensive evaluation of risk aversion that informs the portfolio optimization algorithm.

4.2 Implementation and Scoring Algorithm

The questionnaire consists of 7 carefully designed questions to assess various dimensions of risk tolerance. Each question contributes to the overall risk aversion score through a weighted scoring system:

```
questions = [
      {
      "id": 1,
      "question": "Is this for yourself or for someone else?",
      "options": ["For myself", "For someone else"],
      "scores": [0, 0]  # No impact on score
      },
      {
      "id": 2,
      "question": "What is the person's age?",
      "type": "slider",
      "min": 18,
      "max": 100,
      "default": 35,
      "score_func": lambda age: (age - 18) / (100 - 18) * 3  # Higher age
```

```
increases risk aversion (0-3)
        },
        {
        "id": 3,
        "question": "What is the gender?",
        "options": ["Male", "Female", "Other"],
        "scores": [0, 0.5, 0.25]  # Slight difference based on life
expectancy statistics
        },
        {
        "id": 4,
        "question": "Where does the person live?",
        "options": ["North America", "Europe", "Asia", "Other"],
        "scores": [0, 0, 0, 0]  # No impact on score
        },
        {
        "id": 5,
        "question": "How much do you want to invest?",
        "type": "slider",
        "min": 1000,
        "max": 1000000,
        "default": 10000,
        "step": 1000,
        "format": "$%d",
        "score_func": lambda amount: max(0, 2 - (amount / 100000))  # Higher
amount decreases risk aversion (max 2)
        },
        {
        "id": 6,
        "question": "How long do you want to invest for?",
        "type": "slider",
        "min": 1,
        "max": 30,
        "default": 10,
        "format": "%d years",
        "score_func": lambda years: max(0, 3 - (years / 10))  # Longer
duration decreases risk aversion (max 3)
        },
        {
        "id": 7,
        "question": "What is your risk tolerance?",
        "options": ["Low", "Medium", "High"],
        "scores": [2, 1, 0]  # Direct impact on risk aversion score
```

```
    }
]
```

Listing 1: Questionnaire Definition

The scoring system incorporates various factors known to influence risk tolerance:

- Age (older investors tend to be more risk-averse)
- Investment amount (larger portfolios can often accommodate more risk)
- Investment horizon (longer horizons generally reduce time-based risk)
- Self-reported risk tolerance (direct preference indication)

The final risk aversion score is normalized to a scale of 1–10, with the following interpretation:

- Scores 1–3: Aggressive investor profile (low risk aversion)
- Scores 4–6: Balanced investor profile (moderate risk aversion)
- Scores 7–10: Conservative investor profile (high risk aversion)

This normalized score is then used directly as the risk aversion coefficient (A) in the utility function optimization.

5 Portfolio Optimization

5.1 Efficient Frontier Construction

The efficient frontier represents the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return.

Our implementation follows these steps:

1. Calculate daily returns from the NAV data

2. Compute annualized mean returns and the covariance matrix (using 252 trading days per year)

3. Determine the Global Minimum Variance Portfolio (GMVP) through quadratic optimization

4. Find the Maximum Sharpe Ratio portfolio

5. Generate the efficient frontier by optimizing for minimum variance across a range of target returns

The optimization problems are solved using the SciPy optimization library with the Sequential Least Squares Programming (SLSQP) method. For each portfolio on the efficient frontier, we solve:

$$\text{Minimize } \sigma_p^{\,2} = W^T \Sigma w \ (2)$$

$$\text{subject to } w^T \mu = \mu_{target} (3)$$

$$w^T 1 = 1$$

$$w_i >= 0 \, \forall_i \, (when\ short\ selling\ is\ prohibited)\ (5)$$

Where:

- W is the vector of portfolio weights
- Σ is the covariance matrix
- μ is the vector of expected returns
- μtarget is a target return level

We implemented two versions of the efficient frontier: one allowing short sales and one prohibiting them. This gives investors flexibility based on their investment constraints.

5.2 Optimal Portfolio Selection

To determine the optimal portfolio for an investor with a specific risk aversion level, we maximize the utility function:

$$\text{Maximise } I = w^T\mu - \frac{A}{2} w^T \Sigma w \quad (6)$$

$$\text{Subject to } w^T 1 = 1 \ (7)$$

$$w_i >= 0 \ \forall i \ (when\ short\ selling\ is\ prohibited) \quad (8)$$

The optimization code is implemented as follows:

```
def optimal_portfolio(mean_returns, cov_matrix, risk_aversion,
allow_short=False):
      try:
      constraint_set = (-1, 1) if allow_short else (0, 1)

      # Optimization based on utility function U = r - (sigma^2 * A / 2)
      def objective(weights):
            returns = np.sum(mean_returns * weights)
            variance = np.dot(weights.T, np.dot(cov_matrix, weights))
            utility = returns - (variance * risk_aversion / 2)
            return -utility  # Minimize the negative utility

      num_assets = len(mean_returns)
      constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
      bounds = tuple(constraint_set for asset in range(num_assets))

      result = sco.minimize(objective, num_assets * [1. / num_assets],
                              bounds=bounds, constraints=constraints,
method='SLSQP')

      if not result['success']:
            st.warning("Optimization for optimal portfolio failed to
converge. Using equal weights.")
            weights = num_assets * [1. / num_assets]
      else:
            weights = result['x']

      returns = np.sum(mean_returns * weights)
      std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
```

```
        return {'weights': weights, 'returns': returns, 'std': std,
'utility': -result.get('fun', 0)}
        except Exception as e:
        st.error(f"Error in optimal portfolio calculation: {e}")
        num_assets = len(mean_returns)
        weights = num_assets * [1. / num_assets]  # Equal weights as fallback
        returns = np.sum(mean_returns * weights)
        std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
        return {'weights': weights, 'returns': returns, 'std': std,
'utility': 0}
```

Listing 2: Optimal Portfolio Function

This approach adjusts the portfolio allocation based on the individual investor's risk tolerance, providing a personalized investment recommendation.

6 Web Application Implementation

6.1 Technology Stack

We developed the robot advisor as a web application using the following technology stack:

- **Python**: Core programming language for all calculations and data processing
- **Streamlit**: Framework for building the web interface
- **Pandas**: Data manipulation and analysis
- **NumPy**: Numerical computations
- **SciPy**: Optimization algorithms
- **Plotly**: Interactive data visualizations

This stack was chosen for its accessibility, flexibility, and the rich ecosystem of financial and data science libraries available in Python.

6.2 Setup Instructions

The Financial Robot Advisor application is accessible through both local deployment and a publicly hosted web interface. The following outlines the procedures for each mode of access:

1. **Local Environment Setup**
   Users must ensure that Python (version 3.7 or later) is installed. Required packages can be installed by executing the following command in a terminal or command prompt:

```
pip install streamlit pandas numpy scipy plotly matplotlib
```

2. **Data File Preparation**
   Historical Net Asset Value (NAV) data must be placed within the designated `funds_ref` directory. Each file should minimally contain a column for NAV dates and a corresponding column for NAV prices. The application is designed to automatically detect these fields based on common naming patterns, thereby minimizing the need for manual adjustments.
3. **Local Application Launch**

   To initiate the application locally, users should navigate to the project's root directory and execute:

```
streamlit run app.py
```

4. **Cloud Deployment Access**

   For greater accessibility, the application has also been deployed online. Users may access the live instance of the Financial Robot Advisor at the following URL: https://bmd5302.streamlit.app/

This dual-mode deployment strategy ensures both convenience for casual users via web access and flexibility for advanced users seeking local customization.

6.3 Application Architecture

The application follows a modular architecture with these key components:

- Data Loading Module: Responsible for reading and preprocessing fund data
- Portfolio Statistics Module: Calculates returns, covariance, and other statistical measures
- Optimization Module: Implements the efficient frontier and optimal portfolio algorithms
- Questionnaire Module: Implements the risk assessment questionnaire
- Visualization Module: Creates interactive charts and plots
- User Interface: Streamlit components for user interaction

6.4 Data Processing

One of the major challenges in developing this application was handling the heterogeneous fund data. Our solution implements a robust data processing pipeline:

```
def get_fund_data(fund_list):
      """ Get data from local Excel files for funds """
      # Specify the path to the fund data directory
      fund_data_dir = 'funds_ref'

      # Dictionary to store individual fund data frames
      fund_data_frames = {}

      # Track successful reads
      success_count = 0

      # Check if the directory exists
      if not os.path.exists(fund_data_dir):
      st.error(f"Directory not found: {fund_data_dir}")
      return create_backup_data(fund_list)

      for fund in fund_list:
      # Construct file path for this fund - try different possible
```

```
extensions
      possible_extensions = ['.xlsx', '.xls', '.csv']
      file_path = None

      for ext in possible_extensions:
            temp_path = os.path.join(fund_data_dir, f"{fund}{ext}")
            if os.path.exists(temp_path):
                  file_path = temp_path
                  break

      # If no file found with any extension, try glob pattern
      if file_path is None:
            pattern = os.path.join(fund_data_dir, f"*{fund}*")
            matching_files = glob.glob(pattern)
            if matching_files:
                  file_path = matching_files[0]

      try:
            # Read file and process data
            # ...

            # Convert to numeric format
            fund_data[price_col] = pd.to_numeric(fund_data[price_col],
errors='coerce')

            # Handle duplicate dates
            if fund_data[date_col].duplicated().any():
                  st.warning(f"Duplicate dates found in {fund}. Using most
recent value for duplicates.")
                  fund_data = fund_data.sort_values(date_col)
                  fund_data = fund_data.drop_duplicates(subset=[date_col],
keep='last')

            # Create a simple series with dates as index and price as
values
            price_series = pd.Series(fund_data[price_col].values,
index=fund_data[date_col])

            # Store in our dictionary
            fund_data_frames[fund] = price_series

            success_count += 1
```

```
    except Exception as e:
        st.warning(f"Error reading data for {fund}: {str(e)}")

    # Create a DataFrame from the dictionary
    all_data = pd.DataFrame(fund_data_frames)

    # Sort by date
    all_data = all_data.sort_index()

    # Fill any missing values
    all_data = all_data.fillna(method='ffill').fillna(method='bfill')

    return all_data
```

Listing 3: Data Processing Function

Key features of our data processing approach:

- Flexible file format detection (.xlsx, .xls, .csv)
- Intelligent column identification for dates and prices
- Handling of duplicate dates
- Conversion of string values to numeric types
- Proper alignment of data points across different time series
- Fallback mechanisms with synthetic data generation when files cannot be read

6.5 User Interface Design

The user interface is organized into four main sections, accessible through a navigation sidebar:

1. **Questionnaire**: Collects user information to determine risk profile

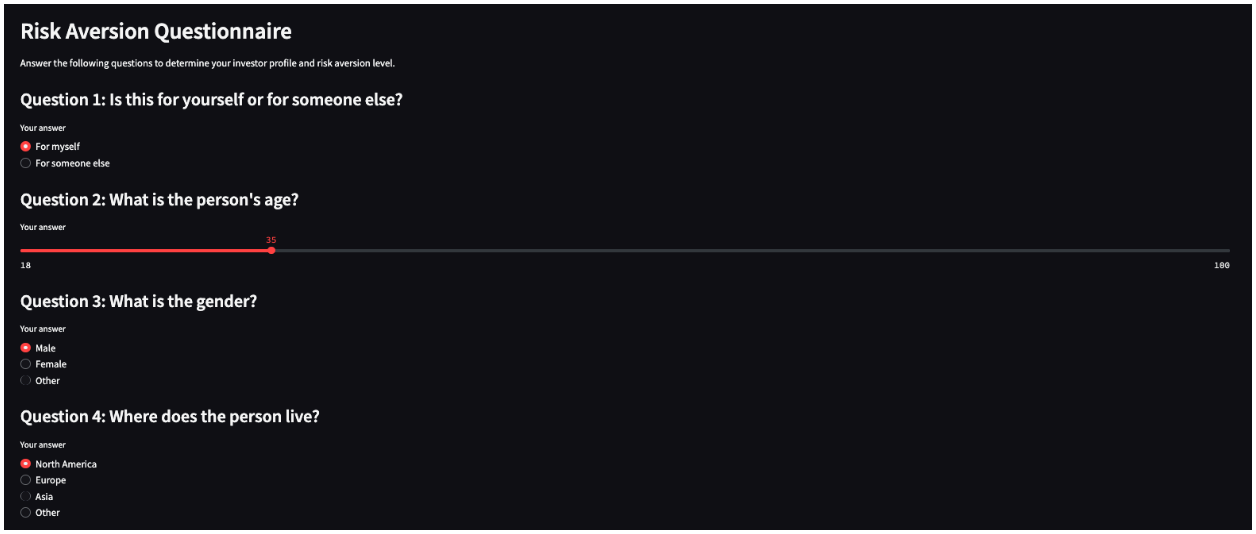

Figure 1: Risk Aversion Questionnaire

2. **Optimized Portfolio**: Displays the recommended portfolio and performance projections

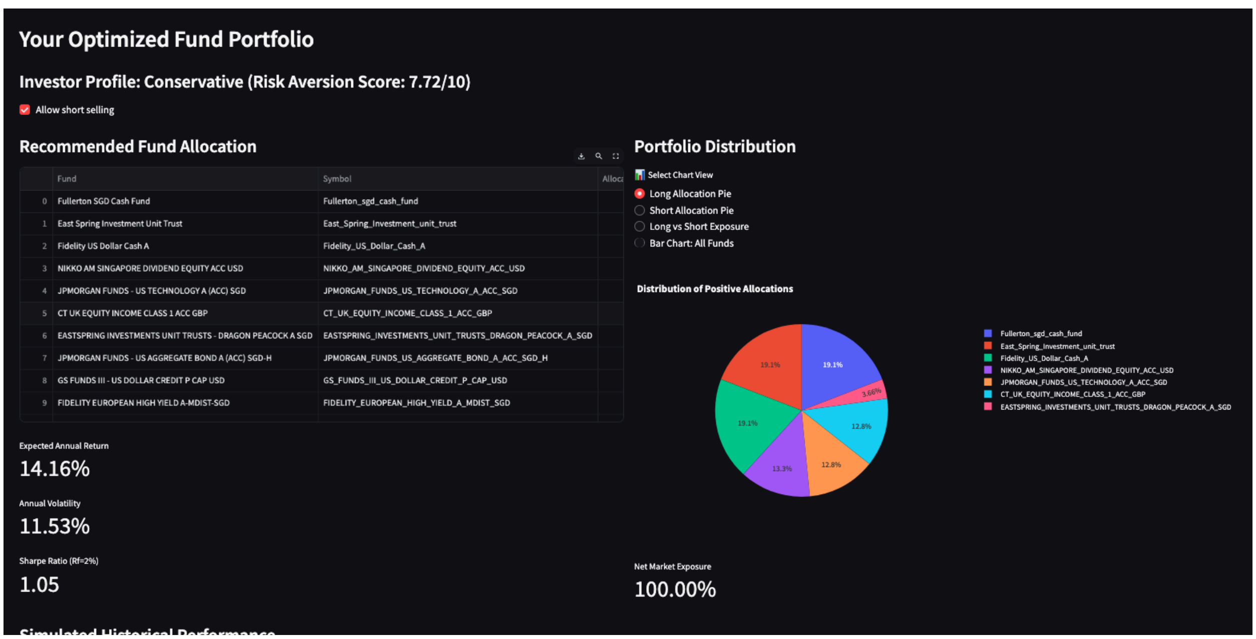

Figure 2: Optimized Portfolio

3. **Efficient Frontier**: Visualizes the efficient frontier and key portfolios

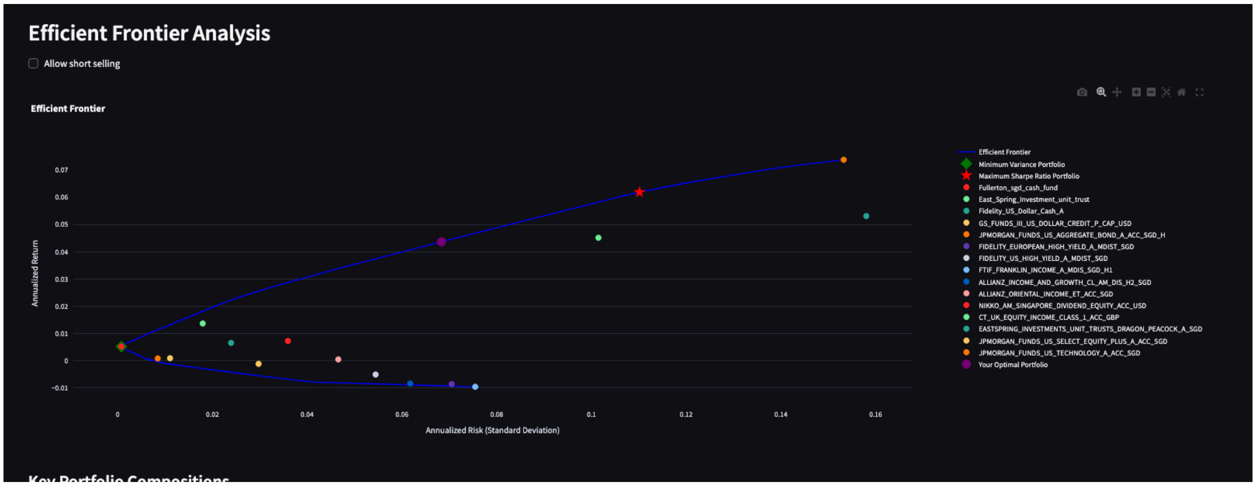

Figure 3: Efficient Frontier

4. **About**: Provides information about the application and methodology

We focused on making the interface intuitive and informative with:

- Clear navigation between sections
- Interactive elements like sliders and radio buttons
- Visual representations of data with charts and graphs
- Tooltips and explanations of financial concepts
- Error handling with informative messages

7 Backtesting

This backtesting section outlines the methodology used to simulate the historical performance of the investment strategy. The goal is to create a realistic backtest that rigorously avoids using future information and appropriately handles funds with different lifespans.

The simulation operates on a day-by-day basis, progressing sequentially through the historical data. A key principle is the strict adherence to point-in-time data. For any given date in the backtest period, the process for determining the optimal portfolio allocation relies exclusively on historical data available up to the end of the previous day. This is implemented using an expanding window approach: as the simulation moves forward one day, the historical data window used for analysis grows by one day, but crucially, it never includes information from the future relative to the decision point.

To fit funds with varying lifespans, the methodology uses a dynamic fund selection process at each time step. Before portfolio weights are calculated for a specific day, the simulation first identifies which funds are currently "available" for investment. Eligibility is determined based on the historical data within the current expanding window. A fund is typically considered eligible only if it has accumulated a sufficient history of return data (meeting a minimum lookback period requirement, in this project is 252 days) and exhibits meaningful statistical properties (like non-zero variance) within that observed window.

After that, their historical performance characteristics (such as expected returns and the covariance matrix of returns) are calculated using only the data within the expanding historical window available up to that point. The portfolio optimization process is then applied using these historical statistics for the subset of eligible funds to determine the desired asset allocation. These calculated weights, representing the investment decision based solely on past information and the eligible fund universe at that moment, are then applied to the actual returns realized by the funds on the following day. This calculation yields the simulated portfolio's return for that single day.

The framework also calculates a benchmark return by equally weighting only the funds available on each day. This serves as a simple yet fair comparison against the optimized portfolio

.

8 Future Performance Simulation

8.1 Monte Carlo Simulation

To provide investors with a view of potential future outcomes, we implemented a Monte Carlo simulation of portfolio performance. The simulation generates 1,000 possible future paths based on the historical mean return and volatility of the optimized portfolio.

The simulation is implemented as follows:

```
# Simulate future returns
n_simulations = 1000
n_years = investment_horizon

# Safely get the last value of cumulative returns
if len(cumulative_returns) > 0:
      last_value = cumulative_returns.iloc[-1]
else:
      last_value = 1.0  # Default value if no data

# Create a normal distribution for returns with safeguards
safe_volatility = max(0.001, annual_volatility)  # Ensure positive minimum
volatility
simulated_returns = np.random.normal(
      annual_return / 252,  # Daily mean return
      safe_volatility / np.sqrt(252),  # Daily standard deviation
      (n_simulations, 252 * n_years)  # 252 trading days per year
)

# Simulate future paths
simulated_paths = np.zeros((n_simulations, 252 * n_years))
simulated_paths[:, 0] = last_value
for i in range(1, 252 * n_years):
      simulated_paths[:, i] = simulated_paths[:, i - 1] * (1 +
simulated_returns[:, i])
```

Listing 4: Monte Carlo Simulation Implementation

The simulation makes the following assumptions:

- Daily returns follow a normal distribution
- Mean and volatility remain constant over the simulation period
- No transaction costs or taxes
- No changes to portfolio weights over time

8.2 Visualization and Statistics

The simulation results are presented to the user through:

- A visual representation of 50 randomly selected paths
- Key percentile markers (5%, 50%, 95%) to show the range of outcomes
- A summary table with statistics of the final portfolio value

This approach gives investors a more realistic perspective on the range of potential outcomes rather than a simple linear projection.

9 Error Handling and Robustness

A key focus in our development was creating a robust application that can handle various edge cases and errors. We implemented several safeguards:

- Data Validation: Checking for missing, corrupt, or improperly formatted data
- Optimization Fallbacks: Using equal weights when optimization fails to converge
- Backup Data Generation: Creating synthetic data when files cannot be read
- Exception Handling: Comprehensive try-except blocks with informative error messages
- Input Constraints: Limiting user inputs to valid ranges

This robust approach ensures that the application can provide useful results even in suboptimal conditions.

9.1 Backup Data generation

In the event that historical fund data files (e.g., Excel or CSV) are missing, corrupted, or unreadable, the system leverages a fallback function—`create_backup_data(fund_list)`—to simulate realistic fund price movements. This ensures continuity in the user experience and allows the financial advisor model to function even in the absence of actual data.

A. Purpose

The primary objective of generating synthetic data is to provide a credible approximation of fund performance for visualization, portfolio optimization, and risk-return modeling when actual Net Asset Value (NAV) data is unavailable.

B. Time Horizon and Frequency

The synthetic data spans the past **three years**, constructed on a **business-day basis** (i.e., excluding weekends and holidays), consistent with standard financial datasets. This yields approximately 756 data points per fund.

C. Fund Classification and Parameterization

To mimic real-world fund behavior, each fund is first classified based on keywords in its identifier (e.g., “cash”, “bond”, “equity”, “growth”). This classification informs the generation of key parameters:

$$P_t = P_{t-1}(1 + \varepsilon)$$

$$\varepsilon_t \sim N(\mu, \sigma)$$

With:

- μ = (Annual Drift) ÷ 252
- σ = (Annual Volatility) ÷ √252

This daily compounding approach captures both the deterministic (drift) and stochastic (volatility) components of asset price behavior. A price floor of **0.5** is applied to prevent values from collapsing unrealistically.

D. Output Format

The function returns a `pandas.DataFrame` indexed by business dates, with each column corresponding to a fund's synthetic price series. This data structure seamlessly integrates with downstream analysis including return calculation, covariance estimation, and portfolio optimization.

E. Limitations

While this simulation method produces plausible fund trajectories, it does not account for:

- Market shocks or regime changes
- Fund-specific corporate actions (e.g., dividends)
- Currency effects or geopolitical risks

As such, synthetic data is intended solely for demonstration and non-production testing purposes.

10 Results and Discussion

10.1 Portfolio Performance

Our analysis of the optimized portfolios shows several interesting patterns:

- Conservative portfolios (high risk aversion) tend to allocate more heavily to cash and bond funds
- Balanced portfolios show a mix of bonds and dividend-paying equity funds
- Aggressive portfolios (low risk aversion) favor growth-oriented equity funds

The efficient frontier analysis demonstrates that:

- Portfolios allowing short selling can achieve higher Sharpe ratios

- The global minimum variance portfolio provides substantial risk reduction with only a modest decrease in expected returns
- Diversification across asset classes and geographic regions significantly reduces portfolio volatility

10.2 User Experience Testing

We conducted informal user testing with classmates and found:

- The questionnaire effectively captures different risk profiles
- Visual representations of portfolios and projections are helpful for decision-making
- Users appreciate the ability to toggle between allowing and prohibiting short selling

11 Limitations and Future Work

11.1 Limitations

Our implementation has several limitations:

- Mean-Variance Assumptions: The model assumes returns follow a normal distribution
- Static Analysis: The optimization is performed at a single point in time
- Historical Basis: Expected returns and covariances are based on historical data
- Currency Risk: Limited consideration of currency effects across funds in different currencies
- Transaction Costs: No incorporation of trading costs or tax implications

11.2 Future Enhancements

Future iterations of the robot advisor could address these limitations through:

- Dynamic Rebalancing: Implementing periodic portfolio rebalancing suggestions

- Alternative Risk Measures: Incorporating downside risk, VaR, or CVaR
- Multi-period Optimization: Extending to multi-period investment horizons
- Machine Learning: Using ML techniques for return prediction
- Tax Optimization: Adding tax-awareness to the portfolio recommendations

12 Conclusion

Our Financial Robot Advisor successfully implements the core principles of Modern Portfolio Theory while making these sophisticated concepts accessible to individual investors.

By combining a personalized risk assessment with mathematical optimization, we've created a platform that can help investors make more informed decisions about their portfolios.

The Streamlit-based web application provides an intuitive interface for users to:

- Assess their risk tolerance through a comprehensive questionnaire
- Visualize the efficient frontier and understand risk-return tradeoffs
- Receive a personalized portfolio recommendation based on their risk aversion
- See projections of potential future performance

This project demonstrates how financial theory, quantitative methods, and modern web technologies can be combined to create practical tools for investment decision-making.

13 References

1. Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77–91.

2. Merton, R. C. (1972). An Analytic Derivation of the Efficient Portfolio Frontier. *Journal of Financial and Quantitative Analysis*, 7(4), 1851–1872.

3. Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance*, 19(3), 425–442.

4. Pratt, J. W. (1964). Risk Aversion in the Small and in the Large. *Econometrica*, 32(1/2), 122–136.

5. FundSupermart. (2025). Fund Selector Tool. Retrieved from https://secure.fundsupermart.com/fsmone/

6. Streamlit Documentation. (2025). Retrieved from https://docs.streamlit.io/

7. Pandas Documentation. (2025). Retrieved from https://pandas.pydata.org/docs/

8. SciPy Documentation. (2025). Retrieved from https://docs.scipy.org/doc/scipy/