**Software Requirements Specification**

**InsightPredict: A Smart Stock Recommendation System for Personalized Investment Guidance**

**Authors:**

* Bayan Yahia
* Adeeb Ganadry

**1. Introduction**

**1.1 Purpose**

This document outlines the comprehensive software requirements for the InsightPredict project. InsightPredict is a stock recommendation system designed to assist investors in optimizing their portfolios using machine learning techniques, financial analysis algorithms, and user-defined preferences. The system provides recommendations based on financial metrics, historical data, risk tolerance levels, and market trends.

**1.2 Scope**

InsightPredict is designed to simplify and enhance the portfolio management process for investors by leveraging advanced machine learning models and financial optimization techniques. The system enables users to:

* Access and analyze historical stock data from reliable sources including Yahoo Finance
* Predict stock performance using statistical models and machine learning algorithms
* Optimize portfolio allocations based on Modern Portfolio Theory principles
* Incorporate personalized preferences including risk tolerance, investment horizon, and portfolio size
* Visualize portfolio performance through interactive dashboards and graphs
* Receive tailored stock recommendations based on financial metrics and user preferences

**1.3 Intended Audience**

This document is intended for:

* Developers implementing the InsightPredict system
* Project mentor Professor Iris Reinhartz-Berger
* Project customer helping with defining project scope and features

**1.4 Definitions, Acronyms, and Abbreviations**

* **API**: Application Programming Interface
* **MPT**: Modern Portfolio Theory
* **SRS**: Software Requirements Specification
* **UI**: User Interface
* **TFT**: Temporal Fusion Transformer
* **YFinance**: Yahoo Finance API
* **MVP**: Minimum Viable Product

**2. System Overview**

InsightPredict comprises several key components that work together to provide comprehensive investment guidance:

**2.1 Data Processing Module**

* Retrieves historical stock data from Yahoo Finance API
* Processes and cleans financial data to remove inconsistencies
* Calculates financial metrics including returns, volatility, and covariance
* Handles missing data through appropriate statistical methods
* Prepares datasets for model training and portfolio optimization
* Automatic alignment of stock price data with quarterly financial reports
* Noise reduction in stock price fluctuations to highlight medium-term trends
* Advanced missing data handling for improved model accuracy

**2.2 Analysis and Forecasting Module**

* Implements statistical models for stock trend analysis
* Uses machine learning algorithms to predict future stock performance
* Identifies market trends and potential investment opportunities
* Multi-horizon forecasting using the Temporal Fusion Transformer (TFT) model
* Identification of key input features that impact stock performance
* Provides interpretable outputs indicating the impact of different input features
* Fine-tuning of prediction models with new data for enhanced accuracy

**2.3 Portfolio Optimization Module**

* Implements Markowitz's Mean-Variance Optimization algorithm
* Calculates efficient frontier for optimal risk-return tradeoffs
* Generates recommended portfolio allocations based on user preferences
* Evaluates portfolio performance metrics (Sharpe ratio, returns, etc.)

**2.4 User Preferences Module**

* Captures user investment goals and constraints
* Stores risk tolerance profiles and investment horizons
* Allows customization of portfolio size and composition preferences
* Adapts recommendations based on user feedback and selections
* Manages user authentication and profile settings

**2.5 Visualization and UI Module**

* Provides intuitive dashboard for portfolio monitoring
* Displays interactive charts for performance visualization
* Shows portfolio allocation and diversification metrics
* Enables user interaction with system features through web interface

**3. Functional Requirements**

| **ID** | **Requirement Description** | **Category** |
| --- | --- | --- |
| F1.1 | The system shall retrieve historical stock data from Yahoo Finance API | Data Acquisition and Processing |
| F1.2 | The system shall update stock price information on a per-minute basis | Data Acquisition and Processing |
| F1.3 | The system shall support data retrieval for 503 different stocks comprising the S&P 500 | Data Acquisition and Processing |
| F1.4 | The system shall process and store at least 15 years of historical data | Data Acquisition and Processing |
| F1.5 | The system shall detect and handle missing values in stock data | Data Cleansing and Preparation |
| F1.6 | The system shall normalize data for consistent analysis | Data Cleansing and Preparation |
| F1.7 | The system shall calculate derived metrics including returns and volatility | Data Cleansing and Preparation |
| F1.8 | The system shall identify and filter outliers in stock price data | Data Cleansing and Preparation |
| F2.1 | The system shall evaluate stock performance against the S&P benchmark | Performance Analysis |
| F2.2 | The system shall detect significant trends in stock price movements | Performance Analysis |
| F2.3 | The system shall analyze stock volatility and trading volume patterns | Performance Analysis |
| F2.4 | The system shall implement machine learning models for stock return prediction | Prediction Models |
| F2.5 | The system shall update prediction models with new data quarterly | Prediction Models |
| F3.1 | The system shall calculate the efficient frontier using Modern Portfolio Theory | Efficient Frontier Calculation |
| F3.2 | The system shall determine optimal portfolio allocations for different risk levels | Efficient Frontier Calculation |
| F3.3 | The system shall incorporate constraints including maximum allocation per asset | Efficient Frontier Calculation |
| F3.4 | The system shall optimize for user-specified objectives (max return, min risk, etc.) | Efficient Frontier Calculation |
| F3.5 | The system shall calculate expected returns for recommended portfolios | Portfolio Evaluation |
| F3.6 | The system shall calculate risk metrics including volatility and drawdown | Portfolio Evaluation |
| F3.7 | The system shall compute Sharpe ratio and other performance indicators | Portfolio Evaluation |
| F3.8 | The system shall compare portfolio performance against market benchmarks | Portfolio Evaluation |
| F4.1 | The system shall assess users' risk tolerance level | Risk Profile Management |
| F4.2 | The system shall provide at least three risk profiles (conservative, moderate, aggressive) | Risk Profile Management |
| F4.3 | The system shall adjust recommendations based on user risk preference | Risk Profile Management |
| F4.4 | The system shall enable users to update their risk profile at any time | Risk Profile Management |
| F4.5 | The system shall allow users to set their investment horizon (short, medium, long-term) | Investment Constraints |
| F4.6 | The system shall enable users to specify maximum portfolio size (15, 20, 25) | Investment Constraints |
| F4.7 | The system shall respect minimum and maximum allocation constraints per asset | Investment Constraints |
| F5.1 | The system shall provide a main dashboard with portfolio overview | Dashboard Components |
| F5.2 | The system shall display chosen portfolio stocks based on predicted performance | Dashboard Components |
| F5.3 | The system shall include interactive charts for performance visualization | Dashboard Components |
| F5.4 | The system shall present allocation recommendations in graphical format | Dashboard Components |
| F5.5 | The system shall enable users to select stocks for detailed analysis | User Interaction |
| F5.6 | The system shall allow portfolio customization through direct manipulation | User Interaction |
| F5.7 | The system shall provide filtering and sorting options for stock lists | User Interaction |
| F5.8 | The system shall support comparison of multiple portfolio strategies | User Interaction |

**4. Non-Functional Requirements**

| **ID** | **Requirement Description** | **Type** |
| --- | --- | --- |
| NF1.1 | The system shall finetune and analyze data for 500 stocks within 6 hours on a quarterly basis | Performance |
| NF1.2 | The system shall generate portfolio recommendations within 30 seconds of request | Performance |
| NF1.3 | The system shall update stock data daily with minimal downtime | Performance |
| NF1.4 | The system shall support at least 100 concurrent users | Performance |
| NF2.1 | The system shall maintain 99% uptime during market hours | Reliability |
| NF2.2 | The system shall perform data validation to prevent incorrect recommendations | Reliability |
| NF2.3 | The system shall implement error handling for API failures | Reliability |
| NF2.4 | The system shall maintain data integrity through proper validation | Reliability |
| NF3.1 | The system shall provide an intuitive interface requiring minimal training | Usability |
| NF3.2 | The system shall include tooltips and help documentation for all features | Usability |
| NF3.3 | The system shall be accessible to users with various financial expertise levels | Usability |
| NF3.4 | The system shall provide clear visualization of complex financial data | Usability |
| NF4.1 | The system shall implement secure user authentication | Security |
| NF4.2 | The system shall protect user preference data and portfolio information | Security |
| NF4.3 | The system shall comply with data protection regulations | Security |
| NF4.4 | The system shall provide appropriate access controls for different user roles | Security |
| NF5.1 | The system shall scale to accommodate additional data sources | Scalability |
| NF5.2 | The system shall support the addition of new prediction models | Scalability |
| NF5.3 | The system shall handle increasing numbers of users and stocks | Scalability |
| NF5.4 | The system shall maintain performance with growing data volume | Scalability |

**5. Interface Requirements**

**5.1 User Interfaces**

* **Stock Ranking List**: Displays stocks ranked based on predicted performance
* **Stock Information Table**: Shows essential details for each stock including current price, weight in portfolio, and price change
* **Profit/Loss Graph**: Visualizes profit over time with adjustable timeframes
* **Stock Details View**: Provides in-depth information about selected stocks
* **Preference Settings**: Allows users to configure risk profile and constraints
* **Portfolio Visualization**: Displays allocation and diversification metrics

**5.2 External Interfaces**

* **Yahoo Finance API**: Primary source for stock price and historical data
* **Financial News APIs**: Provides market news and sentiment information
* **Data Export Interfaces**: Allows export of recommendations and analysis
* **Twelve Data API** and Finnhub API: For gathering additional stock details and financial insights
* **NewsAPI** For tracking the latest news related to stocks and market trends

**6. System Constraints**

**6.1 Hardware Constraints**

* The system requires server capacity to process data for 500+ stocks
* Machine learning models require sufficient computational resources

**6.2 Software Constraints**

* The system is implemented primarily in Python
* The system uses web technologies for user interface (HTML, CSS, JavaScript)
* The system relies on specific libraries for financial analysis and visualization

**6.3 Environmental Constraints**

* The system requires internet connectivity for API access
* The system depends on the availability of external data sources

**7. Acceptance Criteria**

The system shall be considered acceptable when:

* All functional requirements are implemented and tested
* Performance meets or exceeds specified requirements
* User interface is evaluated for usability and accessibility
* Data processing accuracy is validated against benchmark datasets
* Portfolio recommendations demonstrate improved risk-adjusted returns

**8. Appendices**

**8.1 Data Dictionary**

* **Stock**: A share of ownership in a company
* **Portfolio**: A collection of stocks with specific allocation weights
* **Return**: The profit or loss on an investment, expressed as a percentage
* **Risk**: The volatility or uncertainty of stock returns
* **Efficient Frontier**: The set of optimal portfolios that offer the highest expected return for a defined level of risk
* **Sharpe Ratio**: A measure of risk-adjusted return