OPTI_WEALTH

STOCK PORTFOLIO RECOMMENDATION BASED ON RISK TOLERANCE AND CASH COMPOUNDING.

Authors:

1. Fanice Andayi
2. Jimmy Mumo
3. Kevin Riungu
4. Phyllis Kiruki
5. Cynthia Chelangat
6. Edwin Mwai

- BACKGROUND OF THE INDUSTRY

The investment landscape has evolved significantly over the years, driven by advancements in technology, changing market dynamics, and shifting investor preferences. Traditional investment approaches often relied on static asset allocation strategies that did not adapt to individual risk profiles or market conditions. However, with the advent of data analytics, machine learning, and financial modeling techniques, there is a growing opportunity to provide personalized investment advice tailored to investors' unique needs.'

- INTRODUCTION

In a world where data and algorithms drive financial decisions, envision having a personalized financial advisor at your fingertips. Instead of a magic wand, this system harnesses the power of data and algorithms to craft investment recommendations tailored specifically to you. Imagine inputting your risk tolerance and investment objectives, and presto! You receive customized portfolio suggestions designed not only to maximize returns but also to incorporate compounding strategies for long-term wealth growth. It's akin to having a financial genie fulfilling your wishes, but in the language of data and analytics!
Amidst the intricate landscape of financial markets today, investors grapple with the challenge of optimizing their portfolios to meet long-term financial objectives while effectively managing risk. Without personalized guidance, investors may find it daunting to navigate the array of investment choices available, potentially leading to

less-than-ideal outcomes. This highlights the necessity for a sophisticated recommendation system capable of analyzing individual risk preferences, investment timelines, and market dynamics to offer tailored portfolio suggestions.

- <u>PROBLEM STATEMENT</u>

1. Prevailing Circumstance

Currently, many investors lack access to personalized investment advice tailored to their risk tolerance levels and financial objectives. They often resort to generic investment strategies that may not align with their individual needs, leading to potential underperformance or excessive risk exposure.

2. Problem We're Trying to Solve

Our project aims to address this gap by developing an intelligent recommendation system that leverages machine learning algorithms and financial modeling techniques to provide personalized portfolio recommendations. By considering an investor's risk tolerance, investment horizon, and market conditions, we aim to optimize portfolio allocations and compounding strategies to maximize returns while mitigating risk.

3. How the Project Aims to Solve the Problem

Through data-driven analysis and advanced algorithms, our project seeks to empower investors with actionable insights that align with their unique financial goals and risk preferences. By harnessing historical stock price data, financial statements, and market indices, we aim to build a robust recommendation system capable of dynamically adjusting portfolio allocations and compounding strategies to optimize long-term wealth accumulation.

- <u>OBJECTIVES</u>

1. Main Objective:

The primary goal is to create a tailored recommendation system that enhances portfolio allocations and compounding strategies by considering individual risk tolerances and investment timelines.

2. Specific Objectives:

1.Employ machine learning algorithms to evaluate historical stock prices, financial records, and market indicators for risk assessment and the identification of appropriate investment prospects.
2.Create adaptive portfolio optimization methods that modify asset distributions according to evolving market dynamics and individual choices.
3.Integrate compounding tactics to optimize wealth growth over time and boost portfolio returns.

- <u>NOTEBOOK STRUCTURE</u>

1. Business Understanding
Provide a comprehensive overview of the investment landscape and the need for personalized investment recommendations
Clearly define the problem statement and the objectives of the project
2. Data Understanding
Identify and describe the relevant data sources, including historical stock prices, financial statements, and market indices
Understand the key features and characteristics of the data that will be used for the recommendation system
3. Data Cleaning
Perform necessary data cleaning and preprocessing steps to ensure the data is ready for analysis
Handle missing values, outliers, and any other data quality issues
4. Exploratory Data Analysis
Conduct in-depth analysis of the data to gain insights into the relationships between different variables
Identify patterns, trends, and potential drivers of investment performance
5. Data Preparation
Engineer relevant features from the raw data to be used as inputs for the machine learning models
Perform any necessary data transformations, scaling, or normalization
6. Modeling
Develop and train machine learning models, such as portfolio optimization algorithms and compounding strategies, to provide personalized investment recommendations
Experiment with different algorithms and techniques to find the most effective approach
7. Evaluation
Assess the performance of the recommendation system using appropriate metrics, such as risk-adjusted returns, portfolio diversification, and investor satisfaction
Identify areas for improvement and fine-tune the models accordingly
8. Conclusion, Recommendations, and Next Steps
Summarize the key findings and insights from the project
Provide actionable recommendations for implementing the personalized investment recommendation system
Outline the next steps and potential areas for future development and enhancement

BUSINESS UNDERSTANDING

As a prominent financial services provider in Kenya, we understand the crucial role of guiding clients towards informed investment decisions in a complex and volatile market setting. Clients seek tailored investment advice aligned with their financial objectives, risk preferences, and investment timelines. To meet these demands, our goal is to develop an advanced Stock Portfolio Recommendation System utilizing machine learning algorithms to analyze market data and optimize portfolio allocations. By integrating compounding strategies and dynamic optimization techniques, we aim to enhance long-term wealth accumulation for clients while fostering trust and loyalty. This initiative reflects our dedication to innovation, client-centricity, and market leadership in the financial services sector.

1. Stakeholders:

Investors: Individuals in search of personalized investment guidance tailored to their risk tolerance and financial goals.
Financial Advisors: Professionals seeking to strengthen client relationships by providing sophisticated portfolio recommendations based on data-driven insights.
Financial Institutions: Entities aiming to set themselves apart and attract clients through innovative investment solutions.

2. Metrics of Success:

Portfolio Return vs. Benchmark: Targeting a 3% annual outperformance against a relevant benchmark like the S&P 500.
Risk-adjusted Returns: Aiming for a Sharpe ratio exceeding 0.8 to ensure superior risk-adjusted performance compared to the benchmark.
Portfolio Diversification: Ensuring that at least 90% of portfolios meet diversification standards by being well-spread across various asset classes and industries.

DATA UNDERSTANDING

1.Data Source:

The dataset used for this project is obtained from Yahoo Finance through web scraping. It contains historical stock market data for the top 10 companies listed on the Nasdaq exchange, including Apple (AAPL), Starbucks (SBUX), Microsoft (MSFT), Cisco Systems (CSCO), Qualcomm (QCOM), Meta (META), Amazon.com (AMZN), Tesla (TSLA), Advanced Micro Devices (AMD), and Netflix (NFLX).

2. Data Size:

The dataset consists of 54,801 instances, each with 7 features. Each row represents the stock market data for a specific company on a given date. The features include 'Ticker', 'Date', 'Close/Last', 'Volume', 'Open', 'High', and 'Low'.

- ☐ Dataset Columns and Description:

Column Name	Description
Ticker	The name of the company whose stock market data is recorded.
Date	The date in yyyy-mm-dd format representing the trading date.
Close/Last	The closing price of the stock at the end of the trading day.
Volume	The number of shares traded in the day.
Open	The opening price of the stock at the beginning of the trading day.
High	The highest price of the stock reached during the trading day.
Low	The lowest price of the stock reached during the trading day.

- ☐ Relevance to the Project:

The dataset provides valuable insights into the historical performance of top companies listed on the Nasdaq exchange. It includes essential features such as stock prices (open, close, high, low) and trading volumes, enabling analysis of market trends, price movements, and trading activity over time. The 'Date' column serves as a temporal reference, allowing for time-series analysis and trend identification. By leveraging this dataset, the project aims to develop a portfolio recommendation system that utilizes historical stock market data to generate informed investment recommendations for users, aligning with the project's objective of providing valuable insights for investment decision-making.

DATA CLEANING

a) Importing necessary libraries for data manipulation, analysis, visualization, modeling, and evaluation. Here's a breakdown of the libraries that we will use;

- yfinance: Used for fetching historical stock market data from Yahoo Finance.
- pandas: Essential for data manipulation and analysis, providing data structures and functions to work with structured data.
- numpy: Used for numerical computations and operations on arrays and matrices.
- matplotlib.pyplot: Enables data visualization, particularly for creating plots and charts.
- seaborn: A library for advanced data visualization, offering a high-level interface for attractive and informative statistical graphics.
- train_test_split: From scikit-learn, used to split data into training and testing sets for machine learning models.
- LinearRegression: Also from scikit-learn, for implementing linear regression models.
- mean_squared_error: A metric from scikit-learn to evaluate the performance of regression models.
- datetime: Allows working with dates and times in Python.
- statsmodels.api: Useful for time series analysis and modeling.
- warnings: Enables the handling of warnings in Python.

By importing these libraries, the code sets up the environment for data retrieval, manipulation, visualization, modeling, and evaluation, laying the foundation for a comprehensive data analysis and modeling workflow.

b) We download the historical stock market data for the top ten companies from Yahoo Finance, specifically Apple (AAPL), Microsoft (MSFT), Amazon (AMZN), Starbucks (SBUX), Advanced Micro Devices (AMD), Meta (META), Tesla (TSLA), Cisco Systems (CSCO), Qualcomm (QCOM), and Netflix (NFLX). The data is fetched for the period from January 1, 2000, to April 24, 2024.

The download_top_ten_data function takes a list of tickers, a start date, and an end date as parameters. It downloads the data for each ticker using Yahoo Finance's API and then combines the data into a single DataFrame using the pd.concat function.

The resulting DataFrame, combined_data, contains the stock prices and trading volumes for the top ten companies over the specified period. The DataFrame has a MultiIndex with two levels: 'Ticker' and 'Date'. The columns include 'Open', 'High', 'Low', 'Close', 'Adj Close', and 'Volume'.

The code then resets the index of the DataFrame, converts the 'Date' column to datetime format, and drops the index column. The final output is a DataFrame with a datetime index and the same columns as before.

EXPLORATORY DATA ANALYSIS(EDA)

Exploratory Data Analysis (EDA) using univariate, bivariate, and multivariate analysis techniques, you gain significant insights into our data, using opening and closing stock prices.
a) Using Univariate: This technique allows us to examine individual variables in isolation, to provide insights into the trend analysis, seasonality and cyclicality, volatility and risk, correlation between opening and closing prices, the outliers and anomalies and finally comparison across the companies.
b) Using Bivariate: By exploring relationships between pairs of variables, bivariate analysis helps uncover correlations, dependencies, and interactions between two variables. Checking correlations between stock price and trading volumes, then we create a figure with two subplots, one for the distribution of stock prices and another for the distribution of trading volumes. Each subplot uses a kernel density estimation (KDE) plot to visualize the distribution of the respective variable across different companies. The KDE plots are stacked by company, allowing for a comparison of the distributions across companies. The plots provide a visual representation of the density of the data, helping to identify patterns and trends in the stock prices and trading volumes.
c) Using Multivariate: This involves examining relationships between three or more variables simultaneously. We investigated the trends over time for individual companies and across the entire dataset.
The top-performing companies based on stock price growth are:

TSLA (Tesla) with a growth rate of 6.72%

The top-performing companies based on trading volumes are:

TSLA (Tesla) with a trading volume of 337,112,476,700

These companies have demonstrated strong growth and trading activity, indicating their potential for future success. However, it's essential to consider various factors, including market trends, economic conditions, and company performance, before making investment decisions.

DATA PREPARATION

Calculation of various techniques thus indicates moving averages for stock data. The possible findings from the calculations include;

1. Moving Averages:
   Identifying trends and potential reversal points in stock prices using moving averages like the 30-day moving average.
   Analyzing the relationship between short-term and long-term moving averages to gauge momentum and trend strength.
2. Standard Deviation (Price Volatility):
   Assessing the volatility of stock prices using the standard deviation, which can help in risk management and identifying periods of high volatility.
3. Relative Strength Index (RSI):
   Evaluating overbought or oversold conditions in the market based on the RSI values.
   Identifying potential trend reversals or continuation patterns in stock prices.
4. Moving Average Convergence Divergence (MACD):
   Understanding the relationship between short-term and long-term moving averages to signal potential buy or sell opportunities.
   Analyzing momentum and trend changes in stock prices.
5. Stochastic Oscillator:
   Assessing the momentum and strength of stock price movements based on the Stochastic Oscillator values.
   Identifying potential trend reversals or continuation patterns.
6. Volume Moving Averages:
   Analyzing trends in trading volumes using moving averages to understand changes in market participation and interest.
7. Bollinger Bands:
   Identifying potential overbought or oversold conditions in stock prices based on the upper and lower Bollinger Bands.
   Assessing price volatility and potential breakout points.
8. Lagged Variables:
   Creating lagged variables like CloseLagged_1 to analyze the relationship between current and past stock prices.
   Understanding how past price movements may influence future price trends.

We can gain insights into stock price trends, volatility, momentum, and potential trading opportunities in the market. These findings can help in making informed investment decisions and developing trading strategies based on our historical stock data analysis.

MODELING

- ☐ Our starting point for analyzing stock market data using Python and the 'GOOG' ticker using Yahoo Finance API. We downloaded the data to be further analyzed, visualized, and used in various financial modeling and forecasting tasks.
- ☐ We did a comprehensive data processing workflow demonstrating a structured approach to handling historical stock price data for Google, including cleaning, analysis, visualization, and data export for future reference or analysis.
- ☐ Three models were implemented:-

1.Decision tree

-Classifier

-Regression

2. Random Forest

3. Long Short Term Memory(LSTM)

1. **Decision Tree**
   - Training a decision tree classifier model to predict the direction of stock price changes for Google, evaluates the model's performance, and saves the trained model for future use. Creating strategies for stock trading based on classification models and backtesting the strategies,analyzing the results, and visualizing the performance. It combines machine learning with financial strategies to automate decision-making in stock trading.

   - Evaluating a regression model for stock price prediction using DecisionTreeRegressor and assessing its performance metrics,this process involves training a regression model, evaluating its performance using metrics like RMSE, analyzing prediction errors, and saving the model for future predictions in stock price forecasting, the process of building a regression-based trading strategy, backtesting it, and optimizing the strategy parameters to improve the overall performance.
   - The results;

The RMSE of the decision tree regressor is 1.9415211371227128
sharpe ratio on test data is 0.426582
sharpe ratio on the train data is 0.939722

2. **Random Forest**

- A Random Forest Classifier model is trained with specific parameters on the training data for stock trading.
- The model predicts the 'buy' values for the test data and visualizes the actual 'buy' values against the predicted values.

- A function is defined to predict 'buy' values using the trained model on given data.
- A function is created to perform backtesting by making predictions at different intervals and combining the results.
- Backtesting is conducted using the model and predictors, generating predictions for different horizons.
- The results of the backtesting are displayed, showing the predicted 'buy' values.
- Additional predictors are created by calculating rolling averages and trends based on different horizons.
- The Random Forest Classifier model is updated with new parameters.
- The prediction function is modified to predict probabilities and convert them to binary 'buy' decisions.
- Backtesting is performed with the updated model and new predictors, generating and displaying predictions.
- The mean squared error and mean squared error mean are calculated for the prediction, error analysis.

3. Long Short Term Memory(LSTM)

Demonstrating the process of building, training, tuning, and evaluating an LSTM model for stock price prediction, along with potential findings related to the model's predictive accuracy and performance.

- The data is preprocessed using MinMaxScaler and split into training and testing sets.
- A function is defined to create a dataset with a specified look-back window for time series forecasting.
- An LSTM model is constructed with specific architecture and compiled using mean squared error loss.
- The model is trained on the training data and evaluated using the mean squared error loss on both training and testing sets.
- Predictions are made on the training and testing data, and the results are inverse scaled back to the original data range.
- The true values, train predictions, and test predictions are plotted to visualize the model performance.
- Grid search is performed to find the best LSTM model hyperparameters using custom scoring based on mean squared error.
- The best model obtained from grid search is selected and evaluated on the test data, calculating the Root Mean Squared Error (RMSE).
- The RMSE value provides insights into the model's performance in predicting stock prices based on the LSTM architecture and hyperparameters.

FINDINGS

- Training Loss: The training loss is consistently decreasing over epochs, which is a good sign. It indicates that the model is learning from the data
- Test Loss: The test loss is also relatively low, indicating that the model is generalizing well to unseen data.
- Evaluation: Your training loss is significantly lower than your test loss, which suggests that the model might be overfitting slightly to the training data. This could be mitigated by using techniques like dropout and regularization.
  1. The RMSE (Root Mean Squared Error) value is approximately 0.013, which indicates that the average deviation of the predictions from the actual values is around 0.013 units. This suggests the model is making very accurate predictions.
  2. The R-squared value is 0.9969, which means the model explains approximately 99.69% of the variance in the target variable.
  3. An R-squared value of 0.9969 is exceptionally high, indicating that the model fits the data extremely well. This suggests the model is highly effective at predicting the target variable based on the input features.
- The low RMSE value and the extremely high R-squared value both point to the model having excellent predictive performance and being a very good fit for the data. These results suggest the model is highly effective at forecasting the target variable.

EVALUATION

1. Performance metrics results

The (RMSE) Root Mean Squared Error from the models:-
The RMSE of the decision tree regressor is 1.9415211371227128
The RMSE of the Random forest regressor is 1.6145925269605346
The RMSE of the Long Short Term Memory is 0.013287148207474278
The key metrics to calculate the Sharpe Ratio shows:

1. Mean Daily Return: 0.02406208791223002

This is the average daily return of the investment or portfolio.

2. Standard Deviation of Daily Returns: 0.09337389664626591

This is the standard deviation of the daily returns, which measures the volatility or risk of the investment.

3.Sharpe Ratio: 4.090798580372617

The Sharpe Ratio is calculated as:
Sharpe Ratio = (Mean Daily Return - Risk-Free Rate) / Standard Deviation of Daily Returns

In this case, the Sharpe Ratio of 4.090798580372617 is considered excellent. A Sharpe Ratio of 1 or better is good, 2 or better is very good, and 3 or better is excellent. This means the investment or portfolio has generated a very high return relative to the amount of risk (volatility) it has taken on. The higher the Sharpe Ratio, the better the risk-adjusted performance of the investment.

RESULTS

Calculating evaluation metrics to assess the performance of the portfolio recommendation system.

RMSE

Lower RMSE values indicate better model performance.The Tuned LSTM had the lowest RMSE value.

2. Backtesting Results

Conduct backtesting to assess the historical performance of the recommended portfolios. Simulate portfolio returns based on the recommendations and compare them with benchmark portfolios or market indices,here are the results we got:

Decision Tree Regressor

In sample returns (Train)9145.116981

Out of sample returns (Test) 137.838117

Random forest regressor

Return 13346.592751

Long Short Term Model

3. Risk Analysis

The provided Sharpe Ratios for different models are as follows:

- Decision Tree Regressor: Sharpe Ratio of 0.434264
- Sortino Ratio of 0.739555
- Random Forest Regressor: Sharpe Ratio of 0.711592
- Long Short Term Memory: Sharpe Ratio of 4.090798

The Sharpe Ratio is a measure of risk-adjusted return, indicating how much excess return an investment generates relative to the risk taken. A higher Sharpe Ratio implies better risk-adjusted performance. In this case, the Long Short Term Memory model stands out with a Sharpe Ratio of 4.090798, indicating excellent risk-adjusted returns compared to the other models.

RESULTS

A higher Sharpe ratio is preferred because it indicates a higher return per unit of risk. A Sharpe ratio of 0.8 was the threshold considered acceptable, as it indicates that the investment's returns are sufficient to compensate for the risk taken. So the LSTM was the selected model as it had a sharpe ratio of 4.090798580372617.

MINIMUM VIABLE PRODUCT

A minimum viable product (MVP) for a stock portfolio recommender system could include the following features:

1. Basic Recommendation Engine: Implement a simple recommendation engine that suggests a diversified portfolio of stocks based on the user's risk profile, investment goals, and time horizon. Initially, the recommendations can be based on basic heuristics or rules-based strategies.
2. Portfolio Performance Tracking: Provide functionality to track the performance of recommended portfolios over time. Users should be able to see their portfolio's returns, volatility, and other relevant metrics.
3. User Authentication: Implement user authentication to allow users to create accounts, save their preferences, and access their portfolio recommendations.
4. Basic UI/UX: Develop a user-friendly interface where users can input their preferences, view recommended portfolios, and track portfolio performance.

Regarding the LSTM model, it's claimed to be the best due to having the lowest RMSE, highest Sharpe ratio, and resulting in the highest returns, it implies the following:
1. Lowest RMSE: RMSE (Root Mean Square Error) measures the deviation between predicted and actual values. A lower RMSE suggests that the LSTM model's predictions are closer to the actual stock prices, indicating higher accuracy in forecasting. For an investor, this means they can have more confidence in the model's predictions, potentially leading to better investment decisions.
2. Highest Sharpe Ratio: The Sharpe ratio measures the risk-adjusted return of an investment. A higher Sharpe ratio indicates better risk-adjusted performance, meaning the investment is generating more return per unit of risk taken. For investors, a higher Sharpe ratio suggests that the LSTM model's recommended portfolio has provided superior returns relative to the risk it has taken, making it an attractive investment option.
3. Highest Returns: Ultimately, investors are primarily interested in maximizing returns on their investments. If the LSTM model's recommended portfolio resulted in the highest returns among other models or strategies, it implies that the model's predictions and recommendations have been successful in identifying profitable investment opportunities.

In summary, for investors, a recommendation system with an LSTM model boasting the lowest RMSE, highest Sharpe ratio, and highest returns signifies accuracy in predictions, superior risk-adjusted performance, and the ability to generate significant profits, respectively.

CONCLUSION

In summary, the stock portfolio recommendation system developed using decision tree, Long Short Term Models (LSTM), and random forest models has demonstrated promising performance in predicting the future trends of stocks from the selected companies: AAPL, MSFT, AMZN, SBUX, AMD, META, TSLA, CSCO, QCOM, GOOG, and NFLX. Through backtesting, the effectiveness of the models was evaluated in simulating real-world investment scenarios, providing valuable insights into potential returns and risks.

Key Findings:

1. Model Performance

The LSTM model exhibited the most robust performance in predicting stock price movements and identifying profitable investment opportunities. Its ability to capture complex relationships within the data resulted in reliable forecasts.

2. Backtesting Results

Backtesting the portfolio strategies revealed satisfactory returns over the specified historical period. By simulating investment decisions based on the model's recommendations, the viability of the approach in a practical investment setting was assessed.

3. Diversification Consideration

While the system focused on individual stock predictions without explicit industry diversification, the selected companies span various sectors, offering a degree of diversification in terms of market exposure. Future iterations could explore incorporating industry diversification strategies to further enhance portfolio resilience.

RECOMMENDATIONS

Inclusion of Additional Sectors: To enhance the comprehensiveness of our recommendation system, it is recommended to incorporate additional sectors such as healthcare, technology, energy, etc. This would provide investors with a more diverse set of investment options to choose from.

Data Enrichment: Consider enriching the existing dataset with more comprehensive data sources that cover a broader spectrum of sectors and industries. This could involve accessing additional financial data repositories or utilizing alternative data sources for a more holistic view of the market.

User Feedback Integration: Solicit feedback from users of the recommendation system to understand their preferences, investment goals, and areas for improvement. Incorporating user feedback into the recommendation process can lead to more personalized and relevant stock portfolio suggestions.

NEXT STEPS

Implement Sector-based Filters: Implement sector-based filters or preferences within the recommendation system, allowing users to specify their sector preferences and receive tailored recommendations based on their investment criteria.

Monitor and Evaluate Performance: Continuously monitor the performance of the recommendation system over time and evaluate its effectiveness based on key performance indicators (KPIs) such as portfolio returns, risk-adjusted metrics, and user satisfaction.

Stay Updated: Stay updated with market trends, regulatory changes, and technological advancements in the field of investment recommendation systems. Incorporate relevant updates and enhancements to ensure the recommendation system remains competitive and relevant in the dynamic financial landscape.