Performing Machine Learning on Portfolio Attribution Analysis

Mohit Sonwane, *SJSU Student*

***Abstract*—This project uses machine learning to enhance portfolio attribution analysis to improve portfolio accuracy, strengthen risk management, and enhance stock price predictions.**

I. Introduction

THIS project proposal outlines the plan to apply machine learning techniques to enhance portfolio/performance attribution analysis in asset management. Portfolio attribution analysis is a critical aspect of managing investment portfolios, allowing asset managers to understand the sources of performance deviation from benchmarks. By leveraging machine learning algorithms this project aims to improve the accuracy and insights gained from performance attribution analysis.

II. Theory

In the world of managing investments, looking into how well a portfolio is doing compared to a standard measure (benchmark) is crucial. This process is called portfolio attribution analysis which helps us figure out why a portfolio might be doing better or worse than expected. The main goal is to break down the influences from different regions, types of assets, and business sectors to understand what's making the portfolio perform the way it does. To get a clearer picture, we use performance attribution, a method that measures and identifies where a portfolio's returns are coming from.

- Stock Selection: This looks at how individual stocks affect the overall performance. It helps us see if picking certain stocks adds to the portfolio's success
- Stock Allocation: This metric helps us see how the performance is affected by where the investments are located. Different sectors/stocks can have different economic situations, and this metric helps us understand those impacts For investors, understanding what's driving a portfolio's returns is key to keeping their investors happy by helping them obtain high returns on the portfolio with less volatility. It helps in making smart decisions to improve performance, manage risks, and align strategies with financial goals. To help make sense of our goals, here are some important numbers for benchmark indices
- NASDAQ 100:
  - Returns: 17.31
  - Risk: Higher
- SP 500:
  - Returns: 12.39
  - Risk: Moderate
- Russel 2000:
  - Returns: 8.3
  - Risk: Higher

III. Methodology

To predict future stock prices, we employed a combination of machine learning (ML) models, leveraging cutting-edge technologies for robust analysis. My approach included the use of the following models:

- Prophet Model : Utilizing the Prophet model, a sophisticated forecasting tool, to enhance the accuracy of stock price predictions.
- Long Short-Term Memory: Implementing LSTM, a specialized neural network architecture, for its ability to capture long-term dependencies in stock price data, contributing to more nuanced predictions.
- Custom-Made Models for Time series date: Developing and deploying custom-made ML models tailored to specific characteristics of the stock data, providing a personalized and fine-tuned approach to stock selection. In the realm of stock allocation, we adopted a multifaceted strategy to make informed decisions:
- Sentiment Analysis of Stock News: Employing sentiment analysis on stock-related news to gauge market sentiment and incorporate qualitative information into the allocation process.
- Computed Risk Factors: Utilizing quantitative methods to compute risk factors associated with each stock, enhancing our ability to assess and manage risk in the portfolio allocation. These techniques collectively aim to not only predict stock prices accurately but also to strategically allocate assets based on a holistic understanding of market sentiment and risk factors. The synergy between these methodologies is anticipated to contribute to the overall enhancement of portfolio performance.

IV. Training different Stock Prediction Algorithms

- **Prophet Model**
  - **Overview**: Prophet is a forecasting tool for time series data. It's great for data with clear patterns and works well even with missing information or changes in trends.
  - **Model Training**: The training data is split into a training set (before 2022) and a test set (from 2022 onward). The model is initialized, fitted with the training data, and then used to make predictions. Mean Absolute Error (MAE) is calculated to evaluate the model's performance on both training and test data. The visualizations show a comparison between predicted and actual price actions. Finally, the model

is applied to forecast future prices, and the results are visualized using Matplotlib.

  – **Model's performance**: Evaluation reveals a Test Mean Absolute Error (MAE) of 91.698 and a Train MAE of 2.041. The notably high MAE for the test dataset suggests that the model may not fit well with the data. Despite the use of a substantial dataset, the discrepancy between predicted and actual values indicates room for improvement. It is advisable to explore alternative data sets for more accurate predictions in the context of stock price forecasting.

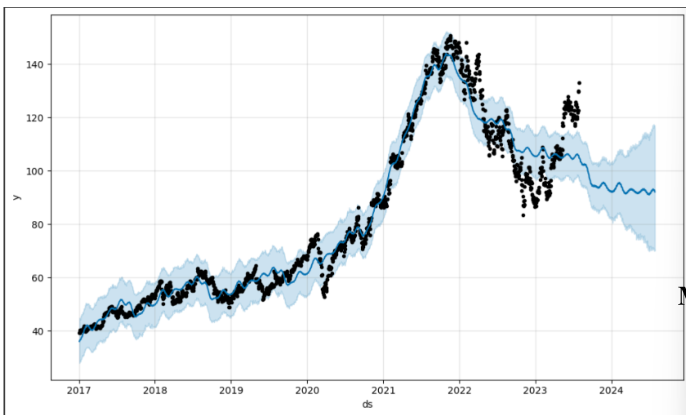


Fig. 1. Future Predicted Stock Prices

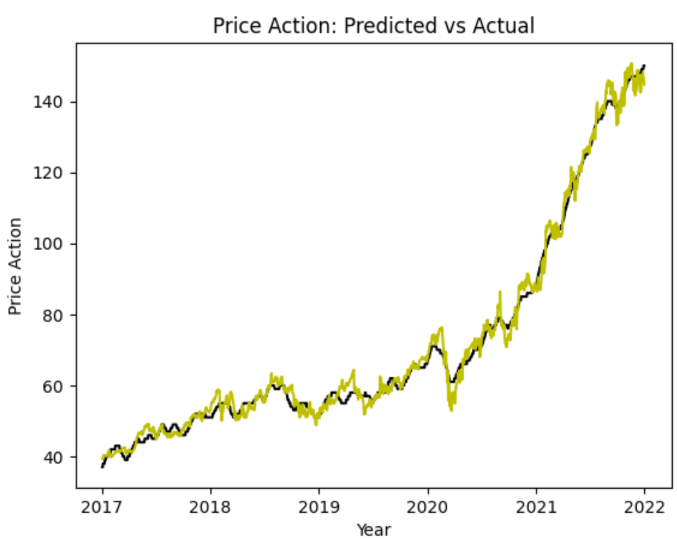


Fig. 2. Performance on Test set

- **LSTM Model**
  – **Overview**: LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data, overcoming the vanishing gradient problem. Its specialized memory cells make it well-suited for analyzing complex patterns in stock data, enabling more accurate predictions by retaining and utilizing crucial historical information.
  – **Model Training:** The data is preprocessed and split using a Time Series Split for training and testing.

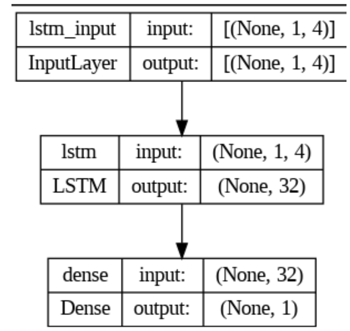


Fig. 3. Enter Caption

The LSTM model, comprising an LSTM layer with 32 units and a Dense layer, is trained on the training set for 100 epochs. The model is then used to predict stock prices on the test set. The visual representation compares the true stock values with the LSTM-predicted values, providing insights into the model's predictive performance.

Model Performance: The LSTM model demonstrated strong performance in stock price prediction, yielding an impressive Mean Squared Error (MSE) of 5.24919 and an R-squared (R^2) score of 0.9992. Outperforming the Prophet model, the LSTM exhibited minimal systematic deviations, as indicated by a residual plot showing residuals randomly scattered around zero. While the model effectively captures the data's structure, there remains a goal to devise a model with an actual loss less than 1 for further refinement.

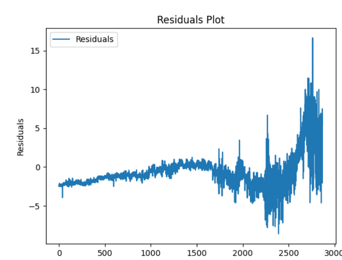


Fig. 4. Residual Factor in stock prediction

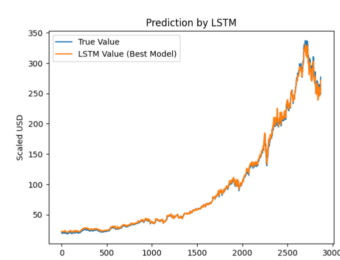


Fig. 5. Performance on Test dataset

Custom RNN GRU Models

- **RNNs** process sequential data by maintaining internal memory and capturing temporal dependencies. However, they face challenges with vanishing gradients, limiting their ability to capture long-term dependencies.

- **GRUs** are a type of RNN designed to address vanishing gradient issues. They incorporate gating mechanisms for better memory management, allowing more effective learning of long-term dependencies in sequential data.
- Stock data trained with several custom models:
  – **Single-Layer SimpleRNN Model**: -A single-layer SimpleRNN model with 6 units.
  – **Single-Layer RNN Model**: - A single-layer SimpleRNN model with 32 units.
  – **Multi-Layer RNN Mode**l: - A multi-layer SimpleRNN model with 32 units in each layer.
  – **GRU Model**: - A GRU (Gated Recurrent Unit) model with four layers and 50 units in each layer.
  – **GRU Model with Dropout**: - A GRU model with dropout layers between the GRU layers.
- **Model Performace**: The custom RNN & GRU models were evaluated based on Mean Squared Error (MSE), R-squared (R^2) scores, and loss values. **RNN_Single:**
  – MSE: 18858.7305
  – R^2: -18.5308
  – Loss: 0.0013

  RNN:
  – MSE: 18845.2830
  – R^2: -18.5168
  – Loss: 0.0017

  GRU:
  – MSE: 18848.0005
  – R^2: -18.5197
  – Loss: 6.4508e-04

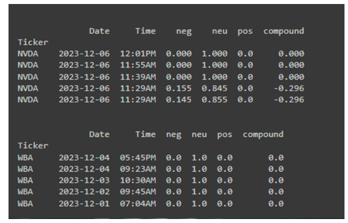
  GRU_Dropout:
  – MSE: 18846.3747
  – R^2: -18.5180
  – Loss: 0.0019

  While all models exhibit similar MSE and R^2 scores, the GRU model with dropout regularization achieved the lowest MSE (18846.3747) and the smallest loss (0.0019). Therefore, the GRU model with dropout appears to outperform the other custom models in terms of predictive accuracy.

V. Sentiment Analysis on Stock News

In my project, sentiment analysis played a crucial role in determining stock weights based on current news. We parsed news headlines, extracting date, time, and text information. Using the Sentiment Intensity Analyzer, we assessed sentiment, categorizing scores as positive, neutral, or negative, with an overall compound score. Mean sentiment scores were calculated for each stock based on compound scores, offering a concise measure of prevailing sentiment. This approach informed the allocation of optimal weights within the investment portfolio, leveraging real-time market sentiments extracted from news headlines.

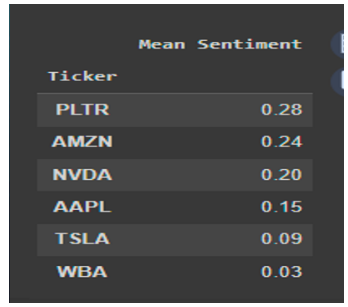
The sentiment analysis process involves using the VADER sentiment analysis tool from the NLTK library to evaluate the sentiment of a given text. Finally, the mean sentiment score for each stock ticker is calculated based on the compound scores, providing a concise measure of prevailing sentiment for informed decision-making in stock sector-wise allocation.



Ticker	Date	Time	neg	neu	pos	compound
NVDA	2023-12-06	12:01PM	0.000	1.000	0.0	0.000
NVDA	2023-12-06	11:55AM	0.000	1.000	0.0	0.000
NVDA	2023-12-06	11:39AM	0.000	1.000	0.0	0.000
NVDA	2023-12-06	11:29AM	0.155	0.845	0.0	-0.296
NVDA	2023-12-06	11:29AM	0.145	0.855	0.0	-0.296

Ticker	Date	Time	neg	neu	pos	compound
WBA	2023-12-04	05:45PM	0.0	1.0	0.0	0.0
WBA	2023-12-04	09:23AM	0.0	1.0	0.0	0.0
WBA	2023-12-03	10:30AM	0.0	1.0	0.0	0.0
WBA	2023-12-02	09:45AM	0.0	1.0	0.0	0.0
WBA	2023-12-01	07:04AM	0.0	1.0	0.0	0.0

Fig. 6. Sentiment score for different stock news



Ticker	Mean Sentiment
PLTR	0.28
AMZN	0.24
NVDA	0.20
AAPL	0.15
TSLA	0.09
WBA	0.03

Fig. 7. the weight allocated to the stock based on sentiment analysis performance

VI. Risk Analysis on Stock Prices

In the project, risk factor analysis is conducted by assessing the volatility of stock returns for selected assets. This approach aids in informed decision-making by quantifying historical volatility and visually representing the distribution of returns for each asset.

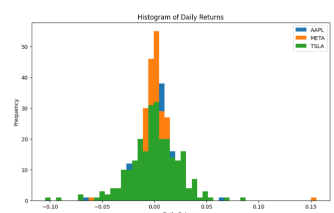


Fig. 8. Risk Factors of Different Stocks

Average Volatility for Each Stock in the above figure:
- **AAPL:** 0.014731
- **META:** 0.017840
- **TSLA:** 0.024073

VII. Conclusions

In summary, the GRU model with dropout consistently outperforms other models in stock prediction. Future work involves fine-tuning parameters for further optimization and exploring various ML RNN models. I aim to develop an integrated model combining real-time stock news sentiment analysis and risk factor analysis, providing investors with a comprehensive decision-making tool. Additionally, I plan to explore unsupervised learning for analyzing interaction effects within portfolios, contributing to a more nuanced understanding of portfolio dynamics. This project marks a step towards sophisticated machine learning applications in finance, enhancing stock prediction and supporting strategic decision-making in investment portfolios.

REFERENCES

[1] "Performance Attribution," DWBI1, Dec. 09, 2018. [Online]. Available: https://dwbi1.wordpress.com/2018/12/09/performance-attribution/

[2] "Machine Learning and Portfolio Performance Analysis," RiskSpan. Available: https://riskspan.com/machinelearning-and-portfolio-performance-analysis/