Financial Portfolio Management

Authors: Sricharran Ramaswamy, Atharva Pandit, Athulya Anand

Abstract— In today's era, with everyone's schedule being so busy it can be difficult to keep up with regular updates and speed up with everything going on around us. One important factor out of which could include managing and getting timely updates on the ever-changing financial affairs, specifically because investment products are unlikely to remain the same over a timeline. Considering such a scenario, even meaningful financial advice can sometimes not serve the purpose, becoming invaluable. Thus, a customized financial portfolio holding stocks can exclusively cater to the investment needs of individuals, considering their financial interest, financial goals, requirement, willingness, and ability to deal with risk. Additionally, in most cases, investment portfolios must be designed in such a way where it can minimize an individual's tax burden. Through this project, we predominantly focus on building a customized financial portfolio to hold best performing stocks for investors and individuals who are interested in keeping up with financial affairs given their need and interests.

Index Terms— Financial Portfolio, stocks, LSTM, Machine Learning, User Interface, Cloud, AWS, S3, Lambda

INTRODUCTION

In alignment with our objective, a more detailed version of the proposed system is as explained below:

- Our proposed system will aim on building a **cloud-based** financial portfolio management system using **Machine Learning** algorithms such as ***Long Short-Term Memory (LSTM) algorithm*** to help users maximize their profit for a given time for which he/she chooses to invest.

- Secondly, this system will also focus on the total amount of money which the individual is willing to invest in, based on which our system will distribute the invested amount optimally on basis of the performances of stocks in the given time.

- Lastly, through this project, we mainly aim to bring out an answer to our ***research question*** as follows:
    a. What is the performance of this financial portfolio system on a local machine Vs that on cloud.
    b. Which has a better performance and why?

1 RELATED WORK AND GAP ANALYSIS

After conducting in depth research and analysis on the topic, we collectively accumulated abundant work found on two strands of the market timing through the expected returns and volatilities. Although, there is some work combining both the factors and its integration with machine learning, we have not come across any approach including the integration of cloud resources in it. Some of the studies conducted in this field are as mentioned:

- Kandel and Stambaugh (1996) study the predictability of equity returns and conclude that a predictor variable like dividend yield may have a significant impact on the best stock-to-cash allocation.

- Goyal and Welch (2008) thoroughly assess the performance of variables that have been recommended by the academic literature as being effective predictors of the equity premium and come to conflicting conclusions.

- Johannes et al. (2014) provides compelling evidence that, if investors include estimation risk and time-varying volatility in their optimum portfolio issues, they can leverage predictability to enhance out-of-sample performance.

- Gu et al. (2020) demonstrated the advantage of applying machine learning for empirical asset pricing and linked the improved predictive performance to the acceptance of non-linear predictor interactions. The most accurate methods for predicting returns were trees and neural networks.

- Nystrup et al. (2016)16 offer a dynamic asset allocation method based on hidden Markov models that is based on change point detection rather than fitting a model with a given number of regimes to the data, estimating any parameters, or assuming a certain distribution of the data.

Referring to the important and significant aspects from each of the studies listed above, we have decided to implement machine learning models like Long Short-Term Memory algorithm to predict the best stock that the user can invest in as per his constraints. Additionally, we will be focusing on incorporating the usage of cloud resources into this system to understand the overall improvements and impacts of cloud on the system.

2 WORKFLOW AND METHODOLOGY

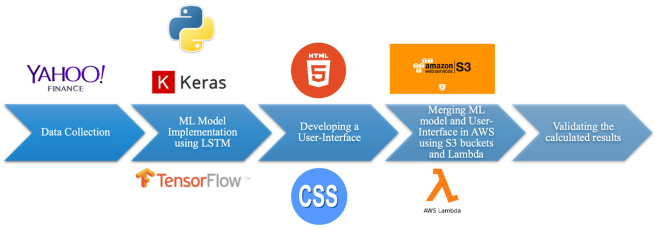

Fig.1.: Workflow Diagram

As shown in Fig.1., our project was sub divided into the following phases.

2.1. Data Collection

The first step was Data Collection. Using the pandas 'datareader' module the data was extracted from Yahoo Finance, a free resource for stock analysis that is part of the Yahoo! network. It provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content. Using Yahoo Finance data allows us to access the performance of the stock from 2012 to today with almost zero latency. This ensures that our model outputs the prediction based on the latest available data which in turn ensures higher accuracy and lowest latency.

2.2. ML Model implementation

Next, analysis has been carried out on the data which is received from Yahoo finance using the LSTM (Long Short-Term Memory) network. It is a special kind of recurrent neural network (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it can process the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc.

Since the only required data was the name of the stock along with the date and closing price, the rest of attributes were eliminated from the dataset, which implies that data cleaning was deemed unnecessary. Next, the data was scaled to a 0-1 scale and was then divided into training and testing dataset with 95% data being used as training data and the rest 5% being used as testing dataset. After which the training data was passed through the LSTM with 128 input layers and 64 hidden layers. The number of output layers was set to 25. Next, we used 'Adam' Optimizer to optimize our output. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first order and second-order moments. The model was then fitted on the training data keeping the batch size as 4 and epoch to 1. Since using multiple epochs was resulting in little to no improvement in the model's performance, the epoch was restricted to 1.

The result of the model was then validated with the actual data. The performance of the model with respect to every stock was as given in Table 1.

Stock Name	Epoch Runtime (in ms)	Root Mean Square Error in %
AAPL	3660	7.814
MSFT	3095	10.997
DIS	3762	5.000
JPM	3441	4.926
MU	3602	3.231

Table 1: Performance of LSTM model on each stock

It can be observed that, although the model was performing quite accurately, it was taking up to 3700 milliseconds to run the epoch.

2.3. User interface

The user interface for the project was designed with simplicity and user requirement in mind. On visiting the website, the user is asked to login to his account.

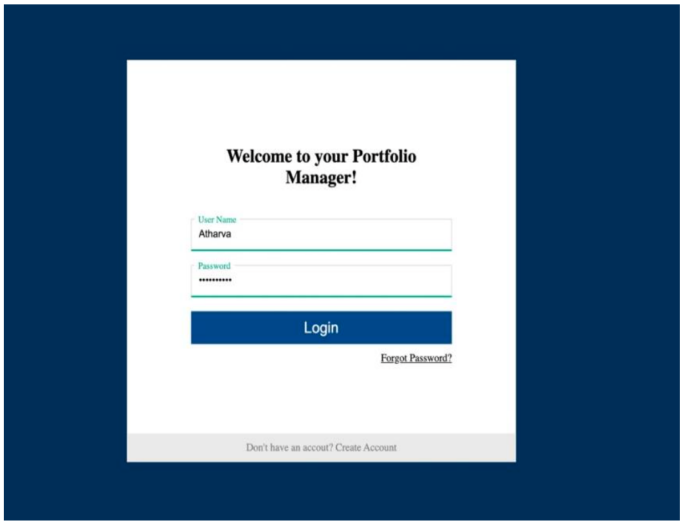

Fig.2.1.: UI Login Page

For new users, the website has a new user registration page which takes the first name, last name and asks the user to create new username and password.

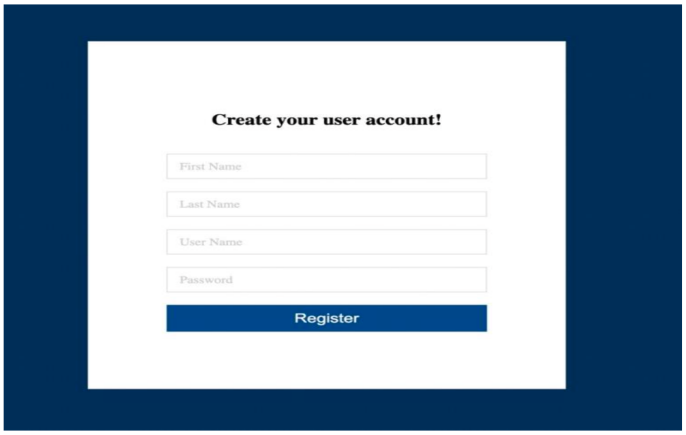

Fig 2.2.: UI Registration Page

The user profile was then saved in a database. Once the user has logged in, he or she can choose the name of the stock to be analysed. Once, the stock has been analysed, the user can see whether it is a good idea to invest in the given stock.

Fig 2.3.: Displaying Stock Performance

2.4. Cloud Integration

The next step was to upload the model along with the user interface to the Amazon Web Services cloud platform. In order to achieve this, AWS Lambda along with S3 (Simple Storage Services) was used. AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. Since Lambda is a scalable service and allows users to build custom backend services lambda was chosen to host the machine learning model. Furthermore, lambda allows the user to connect to the regional database such as S3 which hosted the frontend. Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

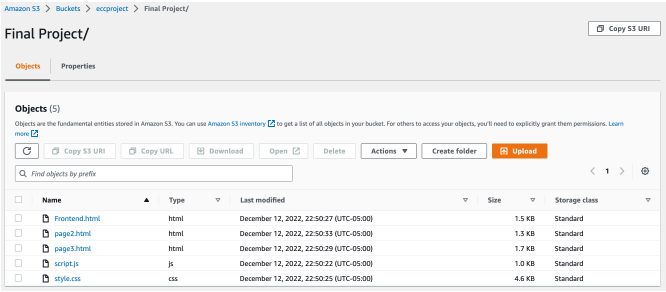

Fig 3.1.: Frontend on S3 bucket

For the Frontend, a new S3 bucket was created with the permission set to public. The files were then uploaded to the S3 bucket, and the viewing permissions were set to public since the viewing permission for objects is different to that of buckets.

To upload the machine learning model, a new lambda function was created. To avoid the model running after every stock name input by the user, one lambda function was created for some of the few selected stocks.

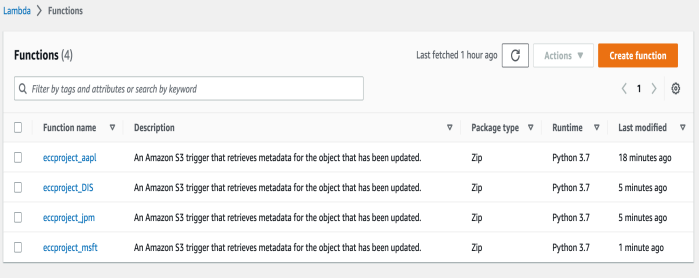

Fig 3.2.: Lambda functions for individual stocks

The lambda function was then connected to the frontend stored in the S3 by calling the Flask API. Thus, suppose the user enters that he wants to analyse the performance of the 'Apple' stock, the S3 will make an API call to the Lambda function for the 'Apple' stock and will display the graph output given by the function. This reduces the latency required to run the machine learning model every time the user enters a new stock name.

3 EXPERIMENTAL RESULTS AND CONCLUSION

The performance of the machine learning algorithm is compared with the data available.

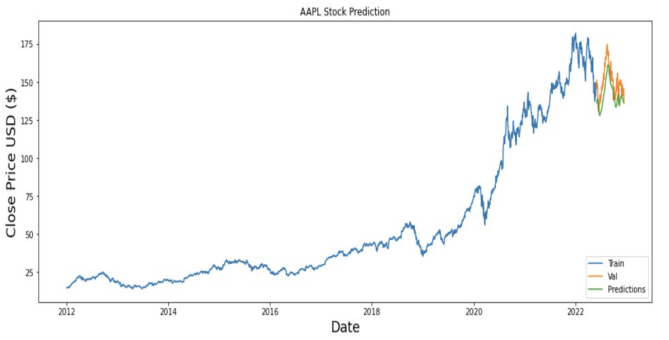

Fig 4.1.: Stock Performance for AAPL ('Apple stock')

As per Fig 4.1., for the year 2022, the blue line represents the test data, the orange line represents the validation i.e., the actual closing rate of each stock and the green line represents the predictions i.e., the predictions made by our model for the stock. It can be inferred that since the orange and green lines are approximately coinciding each other, the actual closing rate and the model predicted rate is nearly the same as shown in Fig 4.2.

Thus, the model developed was quite accurate to the actual data.

Date	Close	Predictions
2022-05-03	159.479996	165.074448
2022-05-04	166.020004	164.702194
2022-05-05	156.770004	165.340469
2022-05-06	157.279999	165.127457
2022-05-09	152.059998	164.671280
...	...	...
2022-11-08	139.500000	149.637512
2022-11-09	134.869995	148.254761
2022-11-10	146.869995	146.654755
2022-11-11	149.699997	146.750076
2022-11-14	148.985001	147.929581

136 rows x 2 columns

Fig 4.2.: Comparison between actual closing rate and predicted closing rate

Fig.4.2., represents a table consisting of the closing rates and the LSTM model predicted dates for each given date. Note that just as shown in Fig.4.1., the predicted values are aligning with that of the actual closing rate values in the table of Fig.4.2. Confirming that the model has performed well.

Thus, the LSTM model has predicted values in the similar fashion for the rest of the company stocks as well.

After the deployment on cloud was completed, the model runtime for the system went down rapidly (as shown Table No. 1). Since for a few well-known stocks, there was a predefined lambda function and saved model, the average epoch runtime went down by a huge margin.

Stock Name	Model Runtime (in ms)	Root Mean Square Error in %
AAPL	235	7.814
MSFT	773	10.997
DIS	362	5.000
JPM	265	4.926
MU	1203	3.231

Table 2: Performance after uploading on cloud

4 LIMITATIONS AND FUTURE SCOPE

The main challenge was the fact that there was no way to validate the result for predicting the future value of a particular stock. Thus, validation was only possible on results on past data. Furthermore, the saving the model using the 'pickle' module is quite difficult since the data for every stock will be difference.

As a part of future scope, we aim to refine the user interface by inferring more results from the machine learning model. Furthermore, we intend to use the AWS Elastic Block Store (EBS) which is a high-performance block storage service which can be used for transaction intensive workloads at any scale.

REFERENCES

[1] https://reader.elsevier.com/reader/sd/pii/S2405918821000155?token=CCD9E085F771337BFCBEA1C890E816C7257037D4528FF4DB48519A577CB418C77ECD289063473D72ACAB97F3AEFC7BE4&originRegion=us-east-1&originCreation=20221008010354

[2] https://www.grayside.co.uk/factsheets12/InvestmentPortfolio.pdf

[3] https://www.raseedinvest.com/learn/what-is-the-importance-of-an-investment-portfolio

[4] https://www.capestart.com/resources/blog/active-portfolio-management-using-ml/

[5] https://towardsdatascience.com/deepdow-portfolio-optimization-with-deep-learning-a3ffdf36eb00?source=collection_tagged---------5----------

[6] https://www.kdnuggets.com/2022/03/build-machine-learning-web-app-5-minutes.html