NOVA IMS Information Management School

STOCK EXCHANGE DATA with Python: Hang Seng Index (HSI)

PG DATA SCIENCE FOR FINANCE

DEEP LEARNING METHODS IN FINANCE 2022/2023

Group 9: Report

Bernardo Machado – M20221001

David Gonçalves – M20221236

Frederico Bravo – M20221231

João Mascarenhas – M20221537

Nelson Lima – M20221539

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Table of Contents

1. Abstract 3
2. Introduction 4
3. Data Exploration & Modelling 4
4. Discussion 7
5. Conclusion 8
6. Bibliography 8

TITLE:

STOCK EXCHANGE INDEX with Python: Hang Seng Index (HSI)

KEYWORDS: Deep Learning Models, Neural Networks, Machine Learning, Financial Markets.

1. ABSTRACT

This report is a comprehensive data analysis of the Stock Exchange Data conducted on the Hang Seng Index (HSI) dataset using Python. Predicting stock prices accurately is a challenging task due to the dynamic nature of financial markets. In this study, we propose the development of a machine learning (ML) model to predict the price of the (HSI), a widely recognized stock index representing the performance of the Hong Kong stock market. By leveraging historical stock market data, our model aims to provide valuable insights for investors and traders. We collected a dataset comprising daily historical HSI prices, from 1986 to 2021. The dataset was pre-processed to ensure data integrity and consistency, and feature engineering techniques were applied to extract meaningful information. Subsequently, we employed machine learning algorithms to train and evaluate our predictive model. The successful development of a robust predictive model for the HSI price has significant implications for investors and financial institutions. By accurately predicting stock market movements, investors can make informed decisions regarding portfolio management and trading strategies. Moreover, financial institutions can utilise our model to enhance risk assessment, asset allocation, and hedging strategies.

2. INTRODUCTION

As part of the Data Science for Finance postgraduate course at NOVA IMS, we were asked to develop a data modelling in finance for Stock Exchange Data to estimate the movement of the stock market. We downloaded our dataset from Kaggle, and the name of the dataset is “Stock Exchange Data”. In this study, the data set contains information about fourteen stock exchange indexes. To predict the closing price, we are going to use the ‘CloseUSD’ column, since these indexes have different currencies, this last column has the Close prices of every index in United States Dollars. The first step of our work was to analyse the dataset and decide which variable(s) to use. We decided to use the variable ‘CloseUSD’, this variable represents the closing price of the HSI Index in USD, which is a critical indicator of the market performance. The other variables do not provide much relevant information to estimate the future price of the indexes. We then pre-processed the data, we separated the indexes and removed the index data from 1986 until the year 2000. This longitudinal analysis allows a deeper understanding of the trends and patterns in the data over this period. The total return value, which is a key metric in financial analysis, is also calculated, providing a summary measure of the dataset’s overall performance. Beyond the initial data exploration and analysis, we also decided to predict the evolution of the HSI index with the application of ML models to the dataset. Our approach and concern to this data modelling in finance was to apply Recurrent Neural Networks to better capture the behaviour of the indexes over time, we also used Long Short-Term Memory as well as Gated Recurrent Unit. These models have shown remarkable success in dealing with time-series data, making them particularly relevant for financial data analysis. Each model’s performance is evaluated, allowing a direct comparison and selection of the best performing model. This report serves as a comprehensive guide to data analysis using Python, from initial data exploration and analysis to the application and evaluation of machine learning models applied to Deep Learning. It provides valuable insights and a practical example that can be applied to a wide range of data analysis scenarios, particularly in the financial sector.

3. DATA EXPLORATION & MODELLING

PREPROCESSING: The Stock Exchange Data has 9 columns and 104.224 rows, and the dataset has columns with ‘Date’, ‘Open’, ‘High’, ‘Low’, ‘Close’, ‘Adj Close’, ‘Volume’ and ‘CloseUSD’. The first step of preprocessing was changing the type of the Date to datetime. In the second step, we separated the datasets by indexes to be able to call them separately. In the last step of preprocessing we changed the Date column to the index of the HSI DataFrame. Firstly, we trained the models with all the data available but since the first year of information that we have for HSI is 1986 we also decided to remove everything until the

year 2000 and try to improve our models this way, as we believe this data may not be very relevant to make predictions. We chose the year 2000 because we believe it is a time where there were a lot of changes in the financial markets, due to the dot com bubble.

DATA ANALYSIS: We only used one index to make the predictions - the HSI (Hong Kong Stock Exchange), the first step of the data exploration was to conduct a "closed_HIS.info()" in order to confirm the non-null values: 8492. To make the predictions of future prices of the index we used the 'CloseUSD' column, representing the closing price of the HSI Index in USD, a critical indicator of the market performance. Then, we used the describe method to get the quartiles and the minimum and maximum values of this index over the full period.

	Count	Mean	Std	Min	25%	50%	75%	Max
CloseUSD	8492	1976,078	1056,4772	246,337	1203,639	1846,0168	2920,3203	4310,035

Table 1: Describe() on the 'CloseUSD' price

We can see that since 1986 the value of the index was between 246 USD and 4310 USD with a mean above the 50% quartile. This indicates that the price distribution is skewed with a longer tail on higher values. Furthermore, we grouped the price averages by year and found out that the stock evolved from an average of USD 333.88 in 1986 to an average of 3761.57 USD in 2021, which translates to a total return of 1035.56%. The value of the total return is found to be indicating a significant increase in the HSI Index closing price over the period under consideration.

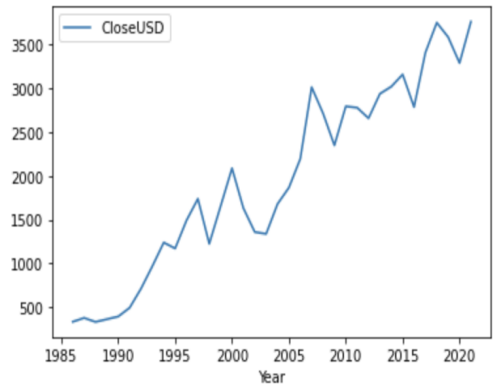

Figure 1: HIS Index closing price over the years.

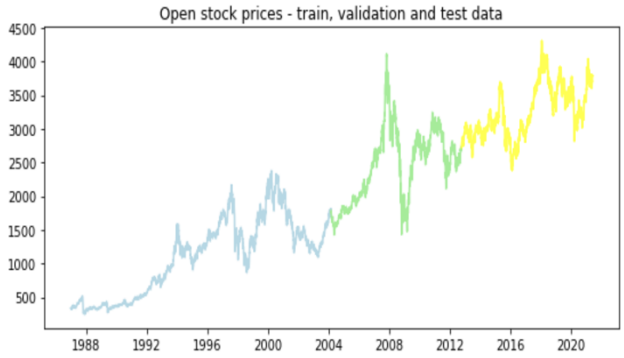

Figure 2: Open stock prices: Train (light blue), Validation (light green), Test Data (yellow)

MODELLING: We split the dataset into train, test, and validation. 50% of the data to train, 25% for test and 25% for validation. Then, we prepared the dataset for modelling, choosing the following parameters: "timesteps"=30, "batch size"=32, our loss function was the "mse", optimizer was "adam", metric was "mae" and we went with 50 epochs. After we performed a reshape of the data with the "MinMaxScaler()".

Train Values	4246	Validation Values	2123	Test Data	2123

Table 2: Values for Train, Validation and Test Data after reshape of the data.

The preparation of the data for training was where we set the number of time steps to look back (30) and the number of samples per batch (32). Then, we prepared the dataset for modelling, choosing the following parameters: our loss function was the “mse”, optimizer was “adam”, metric was “mae” and we went with 50 epochs. With the “timeseriesGenerator()” the train, validation and test were performed. The Model Architecture was first set to the RNN Model, with Sequential() and the Layer type as: SimpleRNN with (128) units with 16640 params and Dense (1) with 129 params. The total params (16769) the same number for the trainable parms. Non-trainable params (0). Then we ran the “fit.generator()” feature. To **evaluate the model** we add a function that prints “Utility function that prints loss and MAE scores of a given model. Returns the …Evaluation of…” when the Evaluation of the model runs for the test, validation, train ‘Loss’ and ‘mae’ if that is the case. After we plotted the predictions. The second model GRU Model was set with Sequential() with 50304 params GRU and dense 129 params. Total params 50433. After we set the fit.generator(). To conclude we also made the evaluation for GRU Model. Then we also performed a second version of the GRU Model with more layers. Our last Model was the LSTM, with 128 units and 1 dense layer. These were our results:

	model_name	test_loss	test_mae	val_loss	val_mae	train_loss	train_mae
0	RNN_MODEL	0.0292	0.1545	0.0119	0.0860	0.0019	0.0286
0	GRU_MODEL	0.0130	0.0982	0.0077	0.0662	0.0018	0.0270
0	GRU_MODEL2	0.0096	0.0814	0.0064	0.0577	0.0017	0.0264
0	LSTM_MODEL	0.0319	0.1629	0.0136	0.0944	0.0020	0.0299

Table 3: Results of the models with the entire dataset

To try and improve our results, we decide to perform these same models, but do a slight change in our data. This time, we were only going to use data starting in the year 2000, instead of 1986. With this, we hope to improve the performance of the model, as we will be using more recent data. As we expected, the results improved, so, using this new dataset, we also decided to apply two new LSTM models: LSTM_MODEL_2 with an extra hidden layer and LSTM_MODEL_3 with an extra dense layer. These were the results:

	model_name	test_loss	test_mae	val_loss	val_mae	train_loss	train_mae
0	RNN_MODEL_New	0.0033	0.0460	0.0020	0.0339	0.0022	0.0315
0	GRU_MODEL_new	0.0039	0.0514	0.0024	0.0382	0.0023	0.0336
0	GRU_MODEL2_New	0.0033	0.0466	0.0021	0.0346	0.0022	0.0316
0	LSTM_MODEL_new	0.0031	0.0443	0.0020	0.0333	0.0022	0.0312
0	LSTM_MODEL_2	0.0022	0.0347	0.0025	0.0383	0.0027	0.0367
0	LSTM_MODEL_3	0.0022	0.0345	0.0022	0.0345	0.0023	0.0320

Table 4: Results of the models with the dataset starting in 2000

4. DISCUSSION

PREDICTIONS: For the prediction we used the function “Predict()” and for each dataset: Train, valuation and test and set a new function to plot these predictions.

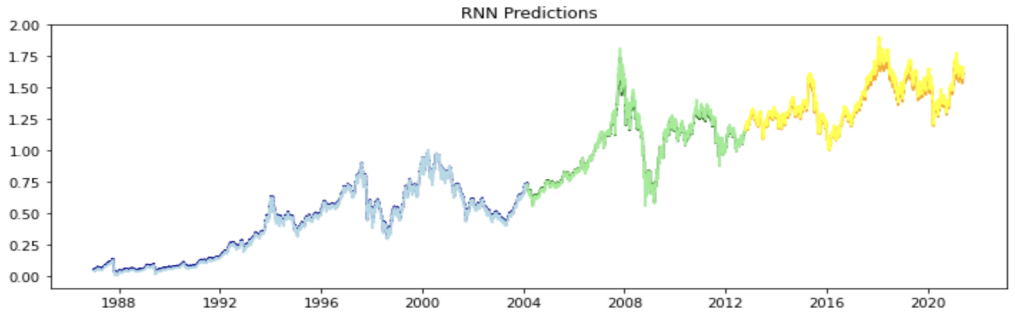

Figure 3: RNN Predictions: Train (Light blue) Valuation (light green), Test (yellow), Train Predictions (dark blue), valuation predictions (dark green), Test predictions (orange)

Our strategy was to perform two rounds with all the models, RNN, GRU, GRU_2 and LSTM, one with the entire dataset, and the other with prices starting in the year 2000. Then to conclude we performed a different version for the LSTM with one more hidden layer marked as LSTM_MODEL_2 and one more dense layer, marked as LSTM_MODEL_3. The final results are presented for LOSS and MAE for all the models. The error of the predictions is measured with LOSS and MAE variables. The best models were: GRU_MODEL2_NEW and LSTM_MODEL_3. In the graphs below we can see the differences between the Training (blue) and Validation (red) values throughout the epochs. We can see that throughout the epochs, in the GRU_MODEL2_NEW, the ups and downs of the validation loss suggest that the model is not generalising well and the performance on unseen data is not good. However, in the LSTM_MODEL_3, the validation loss is getting smaller meaning that the model is learning in each epoch and improving its performance.

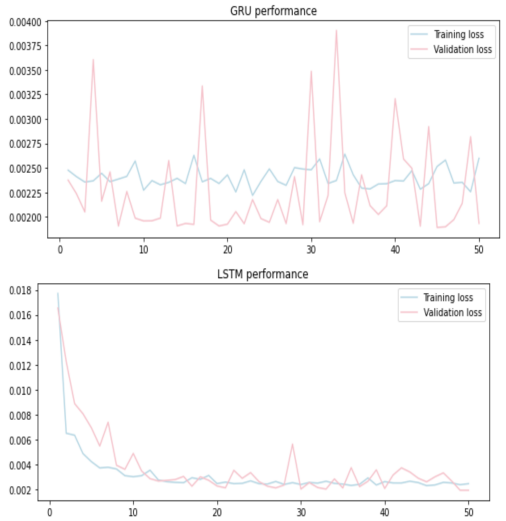

Figure 4: GRU & LSTM Performance (loss)

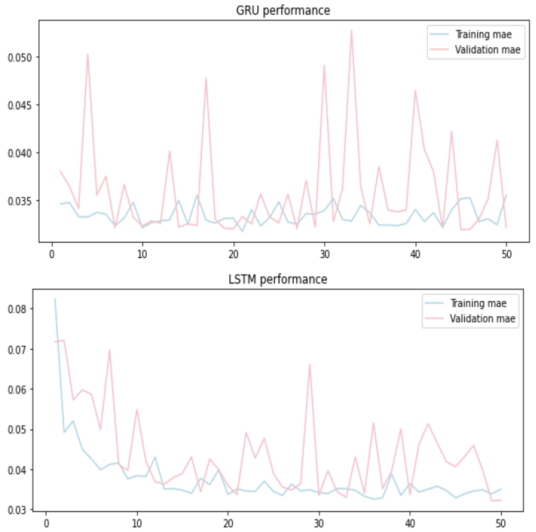

Figure 5: GRU & LSTM Performance (mae)

Regarding the training and validation MAE, the ups and downs on the training MAE of the GRU_MODEL2_NEW, could be explained by the changes in the model error during each epoch. A downward trend is an indicator that the model is improving its ability to fit the data and to make more accurate predictions. Regarding the validation we should also see a downward trend and it should stabilise. However, that does not happen in the validation MAE of the GRU_MODEL2_NEW. The LSTM_MODEL_3 has a downward trend on the training MAE and stabilises and the validation MAE appears to be more stable when compared to the validation MAE of the GRU_MODEL2_NEW. Which leads us to conclude that the LSTM_MODEL_3 is a better model for the problem.

5. CONCLUSION

The python code performed in this study (annex to this report) is trying to solve a problem related to financial data analysis, specifically focused on analysing the 'HSI' Index dataset, which contains the closing price in USD OF THE HSI index. The primary goal was to understand the trends and patterns in the 'CloseUSD' values from 1986 to 2021, and calculate the total return value, which is a key metric in financial analysis. The total return value was 1035.06%, which indicates a significant increase in the 'HIS' Index's closing price over the period under consideration. In addition to this, the code is also trying to predict the future values of the 'CloseUSD' column using various Deep Learning models, including RNN, GRU and LSTM. In terms of data preparation for training the models, we divided the dataset into training, validation, and testing sets. This is crucial for evaluating the performance of the models on unseen data. There was also a need to reshape the data to fit the input requirements of the models. Then, the models are evaluated to determine their performance and effectiveness in predicting the 'CloseUSD' values. Furthermore, we decided to apply these models to two different time frames, one using the entire dataset, and the other reading data that starts only in the year 2000.

6. BIBLIOGRAPHY

Ashofteh, A. (2022) *Big Data for Credit Risk Analysis: Efficient Machine Learning Models Using PySpark*, *easychair*. Afshin Ashofteh. Available at: https://easychair.org/publications/preprint_open/xWFQ