20539425
ANDRES GHERSI SAYAN
PROJECT RESULTS PRESENTATION

Link to Repository: https://github.com/aghersisayan/MachineLearningIIT

INTRODUCTION

For this Fall 23 Term project, we have chosen to develop a machine learning model based on Logistic Regressions, to forecast the movement of the stock value of a top 5 company in the world by market cap: Amazon Inc. (AMZN) (1.5 trillion USD, November 2023).

Why Amazon?

- Publicly traded since 1997. 25 years of data.
- Some ups and downs. Diverse data.
- Part of Nasdaq 100 and S&P 100.

Scope

- Forecasting the stock movement based on historical data: We are not trying to predict the exact value of the stock, as that would introduce too much error and requires more information.
- Preprocessing of data: We are normalizing the data as it is a requirement when working with time series.
- Comparing with a more simple regression that doesn't include data from previous periods: To illustrate cons and pros, we are implementing a logistic regression model to compare with. This will help visualize differences between more simple and more complex models.
- Statistical tests and measurements: For comparison, we are using Accuracy, Precision, Recall, F1 score, Confusion matrix.
- We are Not forecasting the exact value of stock, only the ups and downs
- This is a comparison: We want to highlight the usage of historical data.

OBJECTIVES

We are going to define the objectives of this project and the expected results for each one:

Objective 1: Define the dataset.
    Including the dates to be included in the analysis
    Result 1 for Objective1: Constraints for the data
    Result 2 for Objective 1: Analysis of correlation between variables
Objective 2: Split the data and format it
    Format the data for the time series analysis (preprocessing) and split
    Result 3 for Objective 2: Python code for data preprocessing
Objective 3: Design a Recurrent Neural Network for the data and train it
    Result 4 for Objective 3: Python code for model, data split and train
Objective 4: Implement a state of the art benchmark for comparison
    Result 5 for Objective 4: Python model, data split and train
Objective 5: Compare both models using accuracy estimators
    Result 6 for Objective 5: Metrics for both models and comparison.

Objectives 1 and 2, and their Results will be ready for midterm presentation
Objectives 3 to 5, and their results will be ready for Final presentation

METHODS AND TOOLS

- The programming language is Python, and the ML model will be generated with TensorFlow.
- For measuring accuracy, loss and MSE
- For compute resources, we'll upload the data set and ipython notebook to GoogleColab, which provides enough compute-time and resources for this project

THEORY AND CONCEPTS

How does a RNN differ from a Linear Regression or a Logistic Regression?

RNNs are a special type of machine learning model that allows capturing information from previous states. While a Linear regression or Logistic regression predict values or classes based on the current input, a RNN also considers previous values. This is useful for situations in which temporal dependencies are present (like stock market movements). For this project we will include this advantage into our logistic regression by using the Moving Average to capture previous values.

What are stock market movements?

When trying to forecast the value of a stock in the stock market, we can take the previous values and input them into a linear regression model. This would allow us to predict the future value based on historical data. But for stock movements, we are not trying to forecast the exact value of the stock. We only want to know if the next value will be high or lower than the current. So, we call that a stock market movement.

Which dataset are we using? and how does the data look in the dataset?

We are using the dataset available at Kaggle - Amazon stock 1999-2022 . It presents the daily values from 1999 to 2022, with details of Open value, Close value, High, Low and Volume of transactions.

What is the difference between Time Series Forecasting and Time Series Analysis?

Time series analysis is the study of the data to gain insights like seasonality and trends. Time series forecasting predicts values in the future.

Will there be feature augmentation?

A common indicator used in stock market valuation is the Moving Average (MA). This is a line generated by calculating the average of the values from a specific point in time to the present. Some typical values are MA-200, which is the moving average of

the last 200 periods or the MA-50 for 50 periods. We will be creating two new features with MA-200 and MA-50.

What is the vanishing grading problem and how to solve it with RNN?

When we update weights in a RNN, old values end up vanishing due to a constant multiplication that reduces their impact on the output. This is the vanishing gradient problem. Long memory models losing their old memories. To solve this, a special type of RNN is created: Long Short Term Memory or LSTM RNN. These models have characteristics to maintain old values fresh in the time flow. This is not included in this project, but is worth noting.

What are some improvements that are not included in this solution?

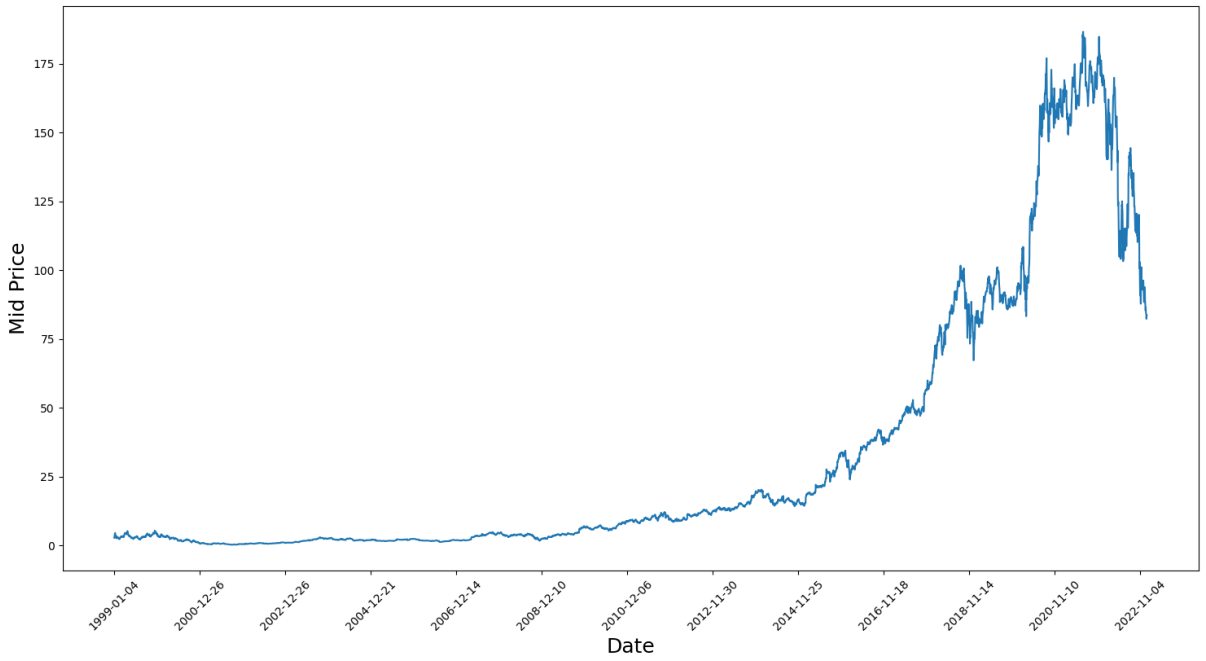
Anomaly detection and seasonal changes will not be covered on this work this time. They are kept as proposed improvements for future work.

RESULTS

The source code for this project is available as two python notebooks on the following repository:
https://github.com/aghersisayan/MachineLearningIIT
These is also a Ppt presentation as pdf at:
Github.com/aghersisayan/MachineLearningIIT/Results Presentation

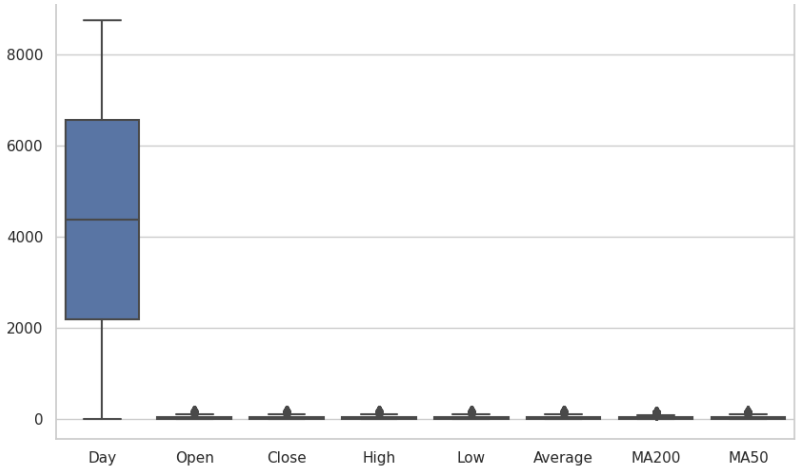
Result 1: Data constraining

We will only use data from 1999 (year in which Amazon.com went public) and 2022 (last date from the dataset).
Limiting the dataset for the RNN machine learning project to the period between 1999 (Amazon.com's IPO) and 2022 offers several advantages. This timeframe encapsulates diverse market conditions, capturing pivotal events shaping stock movements. Additionally, it ensures relevance by excluding outdated data, enhancing the model's capacity to discern contemporary patterns and trends in Amazon's stock behavior for more accurate predictions.



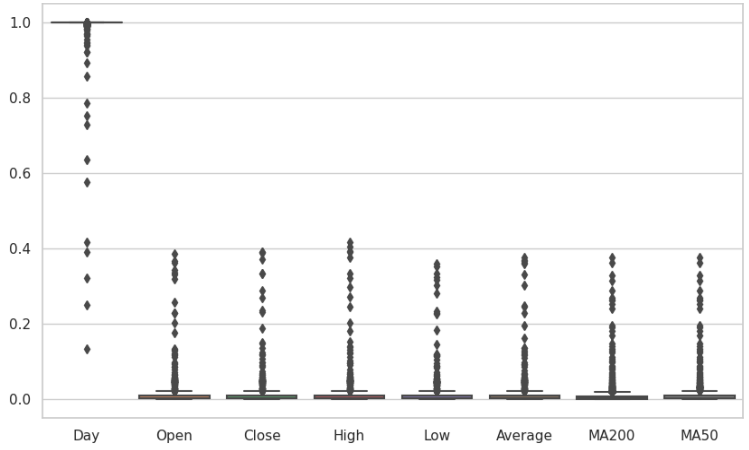
Result 2: Data analysis (original data)

In this stock market movement prediction ML project, the initial focus is on data preprocessing, specifically normalization. Normalizing the data is crucial to ensure that each feature contributes proportionally to the model, preventing certain features from dominating due to differing scales. For instance, stock prices and moving averages can have vastly different ranges. Normalization brings all features to a comparable scale, enhancing the model's ability to discern patterns and relationships across the diverse set of input features.



Result 3: Normalized data and table visualization

Normalization in this stock market prediction project has successfully brought all features, including Day, Open, Close, Low, High, MA200, MA50, and Average value, to a consistent scale of 0 to 1. This ensures equal weighting of each variable, preventing disproportionate influence and promoting a more balanced and effective model.



Result 4 and 5: Logistic regression and LSTM model from literature

In this study, two distinct models were employed for stock market prediction. Drawing from literature, a Long Short-Term Memory (LSTM) neural network was implemented, leveraging its ability to capture temporal dependencies in time-series data.

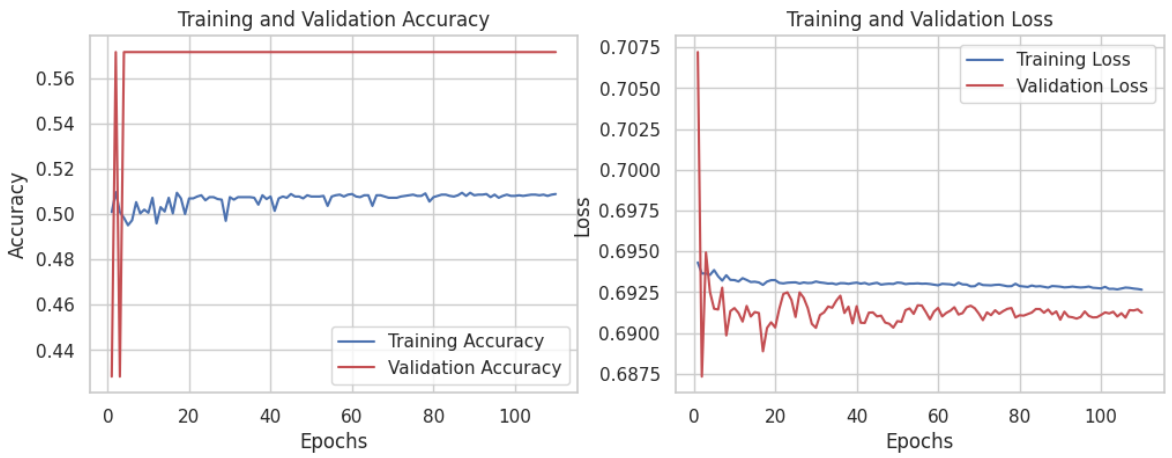
```
model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(train_data.shape[1],)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Complementing this, a logistic regression model was devised, incorporating historical information via the Moving Average (MA) metrics—specifically, the MA50 and MA200. This dual-model approach integrates the sophistication of LSTM with the simplicity and historical context provided by logistic regression, fostering a comprehensive analysis of stock market dynamics and movements.

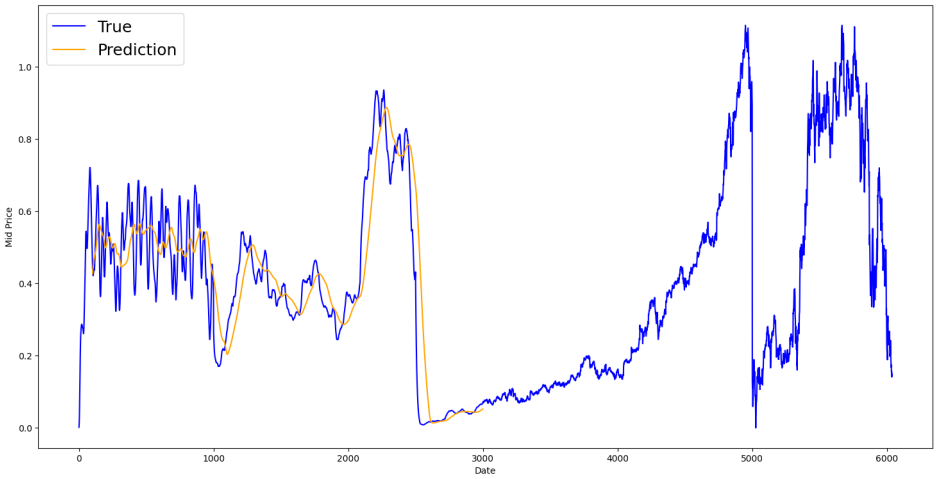
Result 6: Comparison of metrics

Evaluating the accuracy metrics of logistic regression and LSTM RNN provides insights into their respective predictive capacities. Logistic regression, known for its simplicity, is measured by a percentage accuracy score. In contrast, LSTM RNN, with its intricate recurrent architecture, is assessed through metrics like mean squared error (MSE), offering a comprehensive view of its ability to capture complex temporal dependencies in stock market data. Understanding these metrics is crucial for informed model selection and interpretation.

Logistic Regression test accuracy: 0.5078577399253845



LSTM metrics: LSTM MSE error for EMA averaging: 0.00010



Logistic regression achieves a modest test accuracy of 0.5079, reflecting its simplicity, while LSTM outperforms with a low MSE of 0.00010, showcasing its ability to capture complex patterns. Literature supports this, highlighting logistic regression's limitations due to its linear nature compared to the LSTM's recurrent architecture. These results underscore the trade-offs between model simplicity and predictive accuracy in stock market prediction.

CONCLUSIONS

The LSTM model outperforms with a remarkable MSE below 0.01, highlighting its proficiency in capturing intricate patterns for stock movement prediction. In contrast, logistic regression achieves only a 50% accuracy, underscoring its simplicity and potential limitations in handling the complexity of stock market dynamics. This reveals a nuanced insight – while more complex models excel, simpler ones like logistic regression still hold utility. Importantly, these results underscore the pivotal role of datasets, emphasizing that the quality and relevance of input data fundamentally shape the efficacy of machine learning models in financial prediction tasks.

- LSTM behaves better, with a MSE of less than 0.01
- Logistic regression achieves an accuracy of 50%
- While a more complex model is better at predicting movement, simple models can also be used.
- Datasets are the most important part of ML models

REFERENCES

- Google Colab
- How LSTM networks solve the problem of vanishing gradients | by Nir Arbel | DataDrivenInvestor
- Vanishing Gradient Problem in RNNs | by Sagar Patil | Medium
- The Exploding and Vanishing Gradients Problem in Time Series | by Dr Barak Or | Towards Data Science
- A Technical Guide on RNN/LSTM/GRU for Stock Price Prediction | by Chris Kuo/Dr. Dataman | The Startup | Medium
- Amazon (AMZN) - Market capitalization
- Forecasting: theory and practice - ScienceDirect
- When should I use an RNN LSTM and when to use ARIMA for a time series forecasting problem? What is the relation between them? - Quora

- MAE, MSE, RMSE, and F1 score in Time Series Forecasting | by Ottavio Calzone | Medium
- https://www.kaggle.com/code/sriharshaeedala/amazon-stock-price-prediction-with-arima-98
- https://www.kaggle.com/datasets/sriharshaeedala/amazon-stock-price-from-1999-to-2022
- Recurrent neural network - Wikipedia
- How to Develop LSTM Models for Time Series Forecasting - MachineLearningMastery.com
- Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras - MachineLearningMastery.com
- A Quick Deep Learning Recipe: Time Series Forecasting with Keras in Python | by Yang Lyla | Towards Data Science
- Machine Learning to Predict Stock Prices | by Roshan Adusumilli | Towards Data Science
- Understanding LSTM Networks -- colah's blog