DHBW
Duale Hochschule
Baden-Württemberg

Study Paper, T3100: Predicting Stock Prices with Artificial Intelligence using Historical and Real-Time Data

by

Luca Burghard

Matriculation Number:	9209388
Course:	TMT21B2
Study Program:	Mechatronics
University:	Cooperative State University Baden-Wuerttemberg Karlsruhe
Submission Date:	22.12.2023
Auditor:	Steffen Quadt

Blocking Note

The contents of this paper may not be made available, in whole or in part, to persons outside the review and evaluation process unless otherwise authorized by the author.

Used Tools

This document is composed in TEXstudio using LATEX with the language tool provided by LanguageTooler GmbH. Both the text and code are version-controlled on GitHub. The code is scripted in Python using Notepad++. ChatGPT by OpenAI and Bard by Google assist in constructing sentences and identifying synonyms. No content from these chatbots is used. DeepL by Linguee GmbH is employed for translating between German and English and vise versa.

Affidavit

I hereby certify that I have written my study paper: “Predicting Stock Prices with Artificial Intelligence using Historical and Real-Time Data“ independently and that I have not used any sources and aids other than those indicated. I also assure that the submitted electronic version corresponds to the printed version.

_____ _____

Place, Date Signature

Abstract

A project is underway to tackle a common challenge faced by investors: The choice between diversifying their investments or dedicating significant time to researching individual stocks. The objective is to create a program capable of autonomously predicting stock movements and offering insights to users. This allows users to make informed decisions about buying or selling stocks based on these predictions.
The project plan includes the collection of various parameters that affect stock movements, such as news, financial reports, interest rates, global events and economic indicators. The data will be collected from the internet to train an Artificial Intelligence (AI) model. The program will then predict stock movements based on these parameters using real-time inputs.

Building on the fundamentals provided by previous research, the bulk of this paper consists of investigating the methods and programs to be used in the second part. This part of the project aims to collect the necessary data to provide a basis for the second part.
To collect the required data autonomously, a Raspberry Pi (Pi) is set up as a server that automatically executes a program once every day. The data is then stored in folders and backed up on a Solid State Drive (SSD). Using this data, an AI is trained in the following part to predict upcoming stock movements.

This project aims to bridge the gap between informed investing and efficient time management for stock market enthusiasts by harnessing the power of AI and data analytics.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

API Application Programming Interface

CPI Consumer Price Index

CSV Comma-Separated Values

DNN Deep Neural Network

EPS Earnings per Share

ETF Exchange-traded Fund

EUR European Euro

FinBERT Financial Bidirectional Encoder Representations from Transformers

GDP Gross Domestic Product

JSON JavaScript Object Notation

LSTM Long Short-Term Memory

MB Mega Byte

ML Machine Learning

MySQL My Structured Query Language

PCA Principal Component Analysis

P/E ratio Price to Earning Ratio

Pi Raspberry Pi

POMS Profile of Mood States

RNN Recurrent Neural Network

SMTP Simple Mail Transfer Protocol

SOFNN Self Organizing Fuzzy Neural Network

SQL Structured Query Language

SSD Solid State Drive

SSL Secure Sockets Layer

SVM Support Vector Machine

USD United States Dollar

venv Virtual Environment

1 Introduction

Investors face a familiar dilemma: They can choose diversified investment funds or spend significant time managing individual stocks. When investing in a single company, one must track news, review their financial statements, and stay informed about the industry. While trading individual stocks offers greater profit potential, it demands substantial time for research and staying current.

1.1 Motivation and Objectives

This project aims to address the previously mentioned issue by developing a program that can predict stock performance without requiring manual user input. Similar to an investment advisor, the program can provide information such as:

“Considering today’s data, the stock is likely to rise tomorrow. The price will be 5% higher than today“

The user can then decide whether to buy or sell stocks based on the program’s advice. The goal is reached if the investor will get 1% better results trading the stock with the program compared to holding the stock in the long term.

The project is divided into two parts. While the first part mainly focuses on scientific research regarding the state of the art, the second part will be the implementation of the prediction model.

1.2 Delimitation

The program cannot execute trades as it lacks a connection to a broker. Its sole purpose is to provide advice to the user. Additionally, it is designed for a specific stock, which will be chosen at a later point in time. The program won’t have a user interface except the console and won’t be usable by layman. The work is focused on scientific research about the methods and technologies that are possible to use.

2 Main part

2.1 Fundamentals: Stock Market

The fundamental section gives a brief overview about the most important topics mentioned and used in the paper.

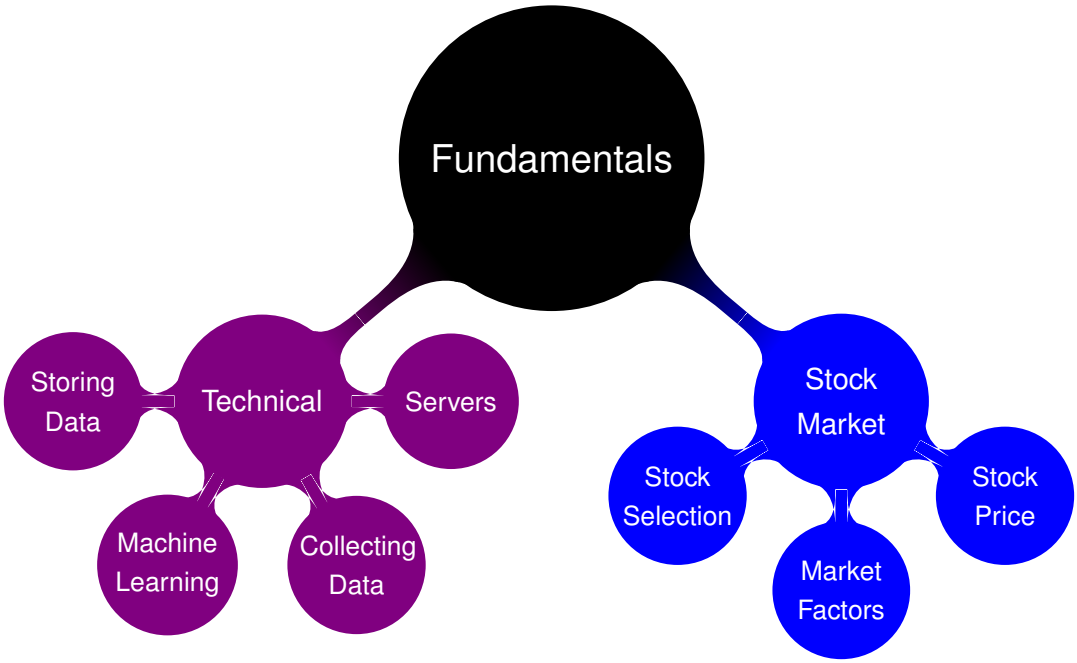

Figure 1: Fundamentals Overview

2.1.1 Stock Price

The price of a stock is determined by supply and demand. If more people want to buy a stock, the price will go up. If more people want to sell a stock, the price will go down. The question is whether people think a companies stock will rise or fall in the future. Sometimes companies perform excellent, but the stock price is falling because investors are thinking the company will fail in the future. It is important to note that stock prices can be volatile and can fluctuate wildly in the short term. However, over the long term, stock prices tend to track the underlying performance of the companies they represent. [1] The difficult part is, that stock prices are not only determined by objective facts and figures, but rather the feelings of traders and buyers about the future of the company. So the project is more of an attempt to predict buyers feelings than a company's success. Or as Philip Fisher said:

"The stock market is filled with individuals who know the price of everything, but the value of nothing" (Philip Fisher 1907-2004)

2.1.2 Stock Selection

Choosing stocks is a complex process that involves a variety of factors. There is no one-size-fits-all approach, the best way to choose stocks will vary depending on the investors individual goals and risk tolerance. However, there are some general principles that you can follow to improve your chances of success.

- **Identify a sector:**
  Firstly the sector needs to be identified. This can be done based on the long-term trends in each sector.
- **Screen for stocks:**
  The next step is to screen for individual stocks. For starters can be done by looking at the biggest companies in that sector.
- **Review the fundamentals:**
  Once a list of potential stocks has been created, the next step is to review the fundamentals of each company in more detail.
- **Select based on Scores:**
  Each share can collect points for the market factors. In each category from 0-10 points. The company with the best score in a category receives 10 points. After adding up all the points from all categories, the company with the highest total can be selected.

2.1.3 Market Factors

There are a number of financial ratios that could influence the stock development. The most important ones are explained here. Some factors can be drawn directly from APIs while others will be calculated. Factors 1-4 are company specific while 5-11 are about the country and the overall market situation.

- **Market Capitalization**: The total value of a company in the stock market, calculated by multiplying the current stock price by the total number of outstanding shares.

$$\text{Market Capitalization} = \text{Stock Price} \cdot \text{Total Outstanding Shares} \tag{1}$$

- **Earnings**: The profits generated by a company over a specific period, typically reported quarterly or annually.

$$\text{Earnings} = \text{Total Revenue} - \text{Total Expenses} \tag{2}$$

- **Earnings Per Share (EPS)**: Calculated as a company's total earnings divided by

its total outstanding shares, indicating profitability on a per-share basis.

$$\text{EPS} = \frac{\text{Net Income}}{\text{Total Outstanding Shares}} \tag{3}$$

- **Price-to-Earnings Ratio (P/E Ratio)**: Calculated by dividing a company's stock price by its earnings per share, used to assess a stock's valuation.

$$\text{P/E Ratio} = \frac{\text{Stock Price}}{\text{EPS}} \tag{4}$$

- **Exchange Rate**: The rate at which one currency can be exchanged for another.

$$\text{Currency Rate} = \frac{\text{Value of Currency A}}{\text{Value of Currency B}} \tag{5}$$

- **Retail Sales**: The total sales of goods and services by retail stores within a specific time frame, often used as an indicator of consumer spending and economic health.

$$\text{Retail Sales} = \sum(\text{Sales of Individual Retail Stores}) \tag{6}$$

- **Consumer Price Index (CPI)**: A measure that examines the weighted average of prices of a basket of consumer goods and services, used to assess inflation's impact on the cost of living.

$$\text{CPI} = \frac{\text{Cost of Market Basket in Current Year}}{\text{Cost of Market Basket in Base Year}} \cdot 100 \tag{7}$$

- **Unemployment Rate**: The percentage of the total labor force that is unemployed and actively seeking employment, serving as an indicator of economic health.

$$\text{Unemployment Rate} = \left(\frac{\text{Number of Unemployed People}}{\text{Total Labor Force}}\right) \cdot 100 \tag{8}$$

- **Federal Funds Rate**: The interest rate at which banks lend reserves to other banks overnight, set by the Federal Reserve to influence economic growth and inflation.

$$\text{Federal Funds Rate is set by the Federal Open Market Committee} \tag{9}$$

- **Gross Domestic Product (GDP)**: GDP adjusted for inflation, providing a measure of a country's economic output.

$$\text{Real GDP} = \frac{\text{Nominal GDP}}{\text{GDP Deflator}} \tag{10}$$

- **Inflation**: The rate at which the general level of prices for goods and services is rising, eroding purchasing power.

$$\text{Inflation Rate} = \left(\frac{\text{CPI in Current Year} - \text{CPI in Previous Year}}{\text{CPI in Previous Year}} \right) \cdot 100 \tag{11}$$

Choosing stocks is a complex process that requires careful research and analysis. However, by following the steps outlined above, investors can increase their chances of success. [2]

2.2 Fundamentals: Technical

2.2.1 Storing Data

Comma-Separated Values (CSV), JavaScript Object Notation (JSON), or Databases are all data formats/ methods that are commonly used to store and exchange data. These formats are all text-based, which makes them easy to read and write and they are also supported by a wide variety of software applications.

- **CSV** is a simple format that stores data in a table-like format. Each row in a CSV file represents a single record, and the columns are separated by commas. CSV files are commonly used to store and exchange tabular data, such as financial records and product catalogs.
- **JSON** is a lightweight data-interchange format that is easy for humans to read and write. JSON stores data in a hierarchical object structure, which makes it ideal for storing complex data structures, such as customer information and product catalogs. JSON is also commonly used in web development and APIs.
- **Database** as storage method is a structured collection of data that is organized so that it can be easily accessed, managed, and updated. Databases are used to store a wide variety of data, including customer information, product catalogs, financial records, and more. Databases are made up of tables, which are collections of rows and columns. Each row represents a single record in the database, and each column represents a different attribute of that record. Common databases that can be used in Python are My Structured Query Language (MySQL). The query language for Databases is Structured Query Language (SQL).[3]

2.2.2 Machine Learning

Machine Learning (ML) is a subtype of AI that allows software applications to become more accurate in predicting outcomes without being explicitly programmed to do so. ML algorithms use historical data as input to predict new output values. There are three main types of ML[4]:

- In **supervised learning**, the algorithm is trained on a set of labeled data, where each input has a known output. The algorithm learns to map the inputs to the outputs, and can then be used to predict the outputs of new inputs.
- In **unsupervised learning**, the algorithm is trained on a set of unlabeled data. The algorithm learns to find patterns and relationships in the data without being told what to look for. This can be used for tasks such as clustering data points into groups or identifying anomalies.
- In **reinforcement learning**, the algorithm learns to take actions in an environment in order to maximize a reward. The algorithm is given feedback on its actions, and it learns to take actions that lead to higher rewards. This can be used for tasks such as training a robot to walk or training a game-playing agent.

Machine learning is used in a wide variety of applications, including:

- **Recommendation systems:** Machine learning is used to recommend products to customers, movies to viewers, and music to listeners. For example, Netflix uses machine learning to recommend movies to its users based on their viewing history.
- **Fraud detection:** Machine learning is used to detect fraudulent transactions and other types of fraud. For example, banks use machine learning to detect fraudulent credit card transactions.
- **Image recognition:** Machine learning is used to identify objects and people in images. For example, self-driving cars use machine learning to identify objects on the road, such as other vehicles and pedestrians.

ML is a powerful tool that can be used to solve a wide range of problems. It is important to note that ML algorithms are only as good as the data they are trained on. If the data is biased or incomplete, the algorithm will learn the biases and produce inaccurate results.

2.2.3 Collecting Data

A set of rules and specifications that define how software components should interact with each other is referred to as API. Different software applications can communicate

with each other and exchange data via APIs. Once a API provider has been identified, an account must be created to obtain a API key. The key is required to make requests and retrieve data from the API.
The following shows an example of an API request:

```
https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&
    symbol=IBM&apikey=demo
```

The violet term called function is the most important part. It describes what type of data will be returned. In this case "TIME_SERIES_DAILY". Separated by "&", other parameters are defined, like the stock symbol.

In Python, the library called "requests" can be used to simplify API requests.

2.2.4 Servers

Servers can serve various purposes, such as hosting websites, storing files, managing emails, or running specific applications accessed by multiple users. In this case the server will be used to run the data collection program every day automatically. The following list describes three methods that can be used to make an server.

- **Laptop**: A server can be created using a laptop by configuring scheduled tasks or "cron jobs" (if using Unix/Linux) to execute the program at specific times. The laptop should be connected to power and kept powered on at the scheduled execution times. The downside is the enormous power consumption even in standby.
- **Raspberry Pi**: A **PI!** (**PI!**) can be utilized as a dedicated server. An operating system (like Debian 11) can be installed and with timers the program can be configured to run at specified intervals. The **PI!** can be left running continuously, serving as a low-power, dedicated server.
- **Cloud Provider**: Services provided by cloud providers like Google Cloud, or Azure allow server instances to be deployed. A server instance can be created, necessary software can be installed, and scheduled tasks can be configured using built-in services. The program can be set up to be run on the cloud server according to scheduling requirements.

2.3 Related Work

The following text with give an overview over the state of the art. Its chronological sorted after the publication year.

Reference	Year	Topic
[5]	2004	Intraday stock price trend forecasting
[6]	2009	Impact of financial news articles on stock market prediction
[7]	2010	Stock prediction through Twitter sentiment analysis
[8]	2013	Use of Twitter feeds to forecast stock closing prices
[9]	2014	Assessing effectiveness of combining market and textual news data
[10]	2018	Sentiment analysis to predict stock price movement
[11]	2018	Comparison of SVM and LSTM models for stock prediction
[12]	2018	Leveraging user-generated content for sentiment analysis
[13]	2019	Application of DNN for stock prediction
[14]	2019	Addressing challenges of financial sentiment analysis
[15]	2021	Comprehensive review on AI in stock market investments
[16]	2023	Stock price prediction using sentiment analysis and deep learning

Table 1: Literature Research Overview

Mittermayer (2004) investigated intraday stock price trend forecasting using text mining techniques. A prior system called "NewsCATS" employed unstructured data, a 392-keyword dictionary, and ML algorithms to categorize press releases into "Good News", "Bad News" or "No Movers." Notably, they selectively included news articles with specific attributes, excluding press releases that lacked a ticker symbol, contained multiple ticker symbols, lacked references to the company's stock exchange, referred to non-NYSE or NASDAQ-AMEX exchanges, or had no subject code. The profit margin was around 0.11% per trade happening 60 minutes after a press publication.[5]

Schumaker and Chen (2009) explored the impact of financial news articles on stock market prediction using three textual representations: Bag of Words, Noun Phrases, and Named Entities. They assessed the ability of these representations to predict discrete stock prices twenty minutes after article release, employing a SVM derivative, which outperformed linear regression. Notably, the Noun Phrase representation scheme proved more effective than the conventional Bag of Words. The resulted in an prediction accuracy of 50.8%.[6]

Mittal (2010) explored stock prediction through Twitter sentiment analysis. The study applied sentiment analysis and machine learning principles to uncover the correlation between "public sentiment" and "market sentiment." Sentiment analysis was performed on publicly available Twitter data, categorizing tweets into four classes - "Calm", "Happy", "Alert", and "Kind". To optimize the analysis process, tweet filtering was applied, considering only tweets containing specific keywords related to expressing feelings ("I feel...", "I am...", "It makes me..."). The paper introduced a unique word list based on the Profile of Mood States (POMS) questionnaire and employed various machine learning techniques, including the Self Organizing Fuzzy Neural Network (SOFNN) for prediction, which was able to get 76% accuracy.[7]

Smailović et al. (2013) investigated the use of Twitter feeds to forecast stock closing prices based on public opinion regarding companies and their products. The study involved categorizing tweets into three sentiment categories (positive, negative, and neutral) and explored the potential influence of emotions on stock market behavior. The research aimed to determine whether sentiment analysis of Twitter data could provide valuable insights into stock price movements. The conlusion was that: "[...]the stock market itself can be considered as a measure of social mood."[8]

Geva and Zahavi (2014) conducted a study to assess the effectiveness of combining numerical market data and textual news data, utilizing data mining techniques to predict intraday stock returns. They emphasized the importance of integrating both data sources to capture joint patterns that might go unnoticed when each source is used independently. They used an new method that was not found in other papers: *"Due to the instability of early-morning trades, we began the analysis at 9:45. We stopped making predictions at 15:00 each day."* The research demonstrated that a trading recommendation system incorporating both market data and news data can leverage the synergy between these two distinct information sources and potentially detect their combined influence on stock prices.[9]

Batra and Daudpota (2018) introduced the concept of utilizing Sentiment Analysis to predict stock price movement based on the sentiment of individuals, demonstrating its influence on stock prices. They conducted sentiment analysis on tweets related to Apple products extracted from StockTwits, a social networking site, from 2010 to 2017. The sentiment score of each tweet was calculated using SVM, categorizing them as bullish or bearish. By combining sentiment scores with market data from Yahoo Finance, they built an SVM model for predicting the next day's stock movement, achieving an accuracy of 76.65% in stock prediction. This research highlighted the

positive relationship between public opinion and market data.[10]

Vignesh (2018) compared SVM and LSTM models for stock price prediction, aiming to predict whether stock prices would increase or decrease. The study's dataset spanned from 2011 to 2015, with a focus on Yahoo and Microsoft data. While the research achieved prediction accuracy for stock price rise or fall, it highlighted the limitation of not being able to predict specific price changes. SVM yielded an accuracy of 65.2%, and LSTM showed a slightly improved accuracy of 66.83%.[11]

Ren, Wu, and Liu (2018) recognized the significance of investor sentiment in the stock market and leveraged user-generated textual content from the internet for sentiment analysis. They integrated sentiment analysis with SVM based machine learning. By considering the day-of-week effect and constructing reliable sentiment indexes, the research achieved remarkable results. The accuracy of forecasting the movement direction of the SSE 50 Index significantly improved, reaching up to 89.93% after introducing sentiment variables, enhancing decision-making for investors. These findings suggest that sentiment analysis can be a leading indicator for stock market behavior, possibly reflecting valuable information about asset fundamental values.[12]

Zhong and Enke (2019) explored the application of DNN and Principal Component Analysis (PCA) for predicting the daily return direction of the SPDR S&P 500 Exchange-traded Fund (ETF) based on financial and economic features. Their study revealed that DNNs with PCA-represented data outperformed other methods in terms of classification accuracy and trading strategy performance. The research emphasized the significance of direction forecasts in trading systems, showing their superiority over level forecasts, especially in a real-world trading environment, as successful direction forecasts are more likely to be profitable.[13]

Araci (2019) addressed the challenges of financial sentiment analysis, primarily the specialized language and limited labeled data in the financial domain. To overcome these challenges, they introduced Financial Bidirectional Encoder Representations from Transformers (FinBERT), a language model based on BERT. FinBERT demonstrated significant improvements over state-of-the-art ML methods in financial sentiment analysis, achieving an accuracy of 97% and surpassing LSTM-based models.[14]

Ferreira, Gandomi, and Cardoso (2021) conducted a comprehensive literature review on the application of AI in stock market investments, analyzing a sample of 2,326 papers published between 1995 and 2019 from the Scopus website. The study provides valuable insights into the advancements and trends in this field over the years, con-

tributing to the understanding of AI's role in stock market trading. The paper was used as an starting point for this papers research.[15]

Kim, Kim, and Choi (2023) explored stock price prediction by combining mathematical-based sentiment analysis and deep learning models. They leveraged the FinBERT Transformer Model, designed for financial language processing, along with LSTM, to investigate the influence of human sentiment on stock movements, particularly in textual data. This specialized language model demonstrated impressive forecasting accuracy, achieving a directional accuracy of 71.18% and a trading return of 8.50%. The study contributes to the understanding of how transformer models like FinBERT can play a crucial role in stock market prediction.[16]

Conclusion

In the realm of stock market prediction, sentiment analysis stands out as a vital tool, utilizing textual data to gauge investor sentiment. ML models, including SVM, LSTM, and transformer models such as FinBERT, have demonstrated their effectiveness in enhancing prediction accuracy. Reliable data sources like Yahoo Finance and The New York Times have played a significant role in model training and evaluation.

Building on the best proven algorithms the program will use FinBERT to analyze sentiment in news because the method has the highest accuracy. Most of the papers used either sentiment analysis or technical factors. To get the best out of both methods the program will use each information source. Furthermore, LSTM seems to be the best method to predict stock prices, so the implementation starts with this method.

2.4 Concept Phase

The overall plan can be summarized as follows: First, select a promising stock based on the criteria and formulas. Next, gather data about the stock through APIs and train an AI model on the collected data using machine learning techniques Subsequently, collect live data and perform analysis to predict stock development. To begin, a specification sheet is created to set our initial direction. Following that, a morphological box will assist in selecting among various options.

2.4.1 Specification Sheet

Some of the requirements are functional and some are non-functional (optional). The following list describes the functional points the final program needs to be able to do.

Essential Requirements:

1. Recommend the user to buy sell or hold a specific stock
2. Collect training data daily
3. Predict short, middle and long term stock development
4. Outperform the "hold strategy" for 1% long term (minimum 1 year)

Optional requirements:

1. Machine learning model is optimized for speed and efficiency
2. Robust security measures to safeguard training data
3. Encryption is used for data transmission and storage
4. Simulate stock investments with virtual portfolios
5. User interface for inputting chosen stocks and viewing predictions and insights
6. System is designed to handle a growing user base and increasing data volume
7. Compliance with relevant financial regulations and data privacy laws
8. Document the system architecture, data sources, and algorithms for internal reference

2.4.2 Timeline

The milestones and dates that are targeted are listet in the table below.

04.10.2023	Project starts
06.10.2023	Project registered
08.10.2023	Paper is set up and formated
13.10.2023	Advisor Steffen Quadt is assigned
20.10.2023	List of Requirements finished
22.10.2023	Overall Projectmanagement and planning finished
23.10.2023	TESTAT 1 is uploaded to Moodle
04.11.2023	Research related Work is done
05.11.2023	All Variations of morphological box are decided
05.11.2023	Stock is chosen –> Adobe
07.11.2023	TESTAT 2 is uploaded to Moodle
12.11.2023	API program is finished and can collect data
18.11.2023	The program runs automatically on a server every day
22.12.2023	Project Part 1 ends
25.03.2024	Project Part 2 starts...

Table 2: Timeline

2.4.3 Morphological Box

Decision/Variant	Variant 1	Variant 2	Variant 3	Variant 4
Programming Language	**Java** +Large ML ecosystem -Verbose syntax	**Python** +User friendly +ML libraries	**C++** +Fast runtime -Complex -Few ML abstractions	
Data Source	**Stock Index** +Common method +Number not emotions	**Combined** +A lot of data -Harder to process -More API = more errors +Much higher accuracy possible	**Sentiment** +Big impact on stocks -Hifficult to analyze +Several sources	**News** -Can be fake -Difficult to analyze
Single or ETF	**Single Stock** +More fluctuations = higher profit margin +Less predictable +Easy to filter relevant articles	**Investment Fund** -Less fluctuation = lower profit margin -Hard to find News about all Stocks		
Data Storage	**Database** +Scalable +Querying +Indexing +Large data -Complex setup	**CSV** +Simple to use +Small memory use -No tools available +Human readable +Hasy to create and share	**JSON** -More complex to use +Structure +Used in API -Larger storage usage	
Data Size Reduction	**PCA FRPCA** +Less data -Complex to set up assumes linearity +Less overfitting -More underfitting +Faster training	**KPCA** +Non linear -More overfitting	**None** +No additional work +Methods can easily implemented later	
Missing Data Filling	**Linear Interpolation** +No tool needed +Easy to use	**Delete** -Model cant find important relations because of missing data	**K-nearest** -Library needed -Complex	
News Analysis Tools	**Term Frequency** +Easy use no library -Words are not overall good or bad	**Inverse Document Frequency** -Complex setup	**FinBert** +Pretrained model for Finance +Accuracy around 97% -complex setup	**Bag of Words** -No context -Just basic understanding
Learning Type	**Supervised** +Labeled data	**Semi-Supervised** +Labeled and unlabeled data	**Unsupervised** +unlabeled Data	**Reinforced** +Reward system
Number or Binary	**Binary (Classification)** -Only 1 or 0 as output	**Numeric (Regression)** +Exact stock price prediction		
Data Source	**Historical and Realtime** -Not possible to find historical data +more data to train	**Realtime** -Less Data +Right format +Easier to validate +Data collection from now on builds historical data	**Historical** -No adjustment to new situations -Can´t find historical news	
Server	**Raspberry Pi** +Low power consumtion +Control over everything -Complex to setup	**Cloud Provider** -Monthly cost +Data safety	**Laptop** -Very high power consumption +Easy to setup -Need old laptop	

Table 3: Morphological Box

The selected variants, indicated by the cells highlighted in yellow, are based on a thorough evaluation of scientific work and a consideration of their respective advantages and disadvantages. These decisions are made after considering what can best be implemented and what is most suitable for the project.

2.5 Implementation

The following sections will use the fundamentals and the methods chosen to implement the program. First the stock will be chosen. Afterwards the API-Provider needs to be chosen. Then the program to collect the data will be written. When finished everything needs to be put on a server to run automatically daily. The last step is to prevent failures and data loss.

2.5.1 Select Stock

The first version of the program will only be able to analyze one specific stock. The right company to invest in is as important as the right program. Firstly the market sector is selected:

10 biggest Market Sectors:

1. Healthcare
2. Financial Services
3. Technology
4. Consumer Discretionary
5. Energy
6. Industrial
7. Materials
8. Consumer Staples
9. Utilities
10. Real Estate

The technology sector is an ideal choice for equity investment as it is home to some of the world's largest companies, all of which operate predominantly in the technology sector, guaranteeing significant growth prospects and innovative advances.

Top 3 Stocks from Technology Sector

The first selection of the top 3 stocks from the technology sector was made on October 12, 2023 at 9 p.m. with the help of "Yahoo Finance" (`finance.yahoo.com`). The website offers search filters for various parameters. The aim of the program, which is developed here, is to predict the long-term development of shares so that a stable share without major fluctuations and yet with high growth opportunities is searched for.

Therefore the selected categories and parameters were:

- Sector: Technology
- Market Capitalization: Mega Cap
- Sort by: Previous year growth

Stocks with no available information were excluded, as were stocks with a currency other than United States Dollar (USD) or European Euro (EUR).

Compare top 3 Stocks
Price, market capitalization, Price to Earning Ratio (P/E ratio), and Earnings per Share (EPS) data were sourced from Yahoo Finance. Yearly growth data was obtained from the Google Stocks Overview (`google.com/finance`). "Revenue Growth" information was provided by Makrotrends (`macrotrends.net`). Oracle information was provided in USD and was converted to EUR at the current exchange rate of 0.95 EUR per USD. Additionally, the Danelfin AI was employed to assign a 1-10 score to each stock, which can be accessed (`danelfin.com`).

Each parameter is normalized to a scale of 1-10 (light brown lines), where the highest score is always 10. The final score for each stock is the sum of these 1-10 ratings. The table is displayed below:

	Oracle Corporation	Intuit Inc.	Adobe Inc.
Yearly Growth	59.07%	33.29%	79.48%
Yearly Growth normalized	7	4	10
Marketcapitalization	285.74 billion €	166.37 billion €	250.40 billion €
Marketcapitalization normalized	10	6	9
P/E Ratio	39.8145	64.95	51.4
P/E Ratio normalized	6	10	8
EPS	2.147	7.98	10.51
EPS normalized	2	8	10
Revenue Growth	8.00%	12.90%	10.00%
Revenue Growth normalized	6	10	8
Danelfin Rating	7	5	6
Danelfin Rating normalized	10	7	9
Danelfin Description	Oracle (ORCL) has an AI Score of 7/10 (Buy) because its overall probability of beating the market (S&P 500) in the next 3 months (41.92%) is +2.97% vs. the average probability (38.95%) of US stocks	Intuit (INTU) has an AI Score of 5/10 (Hold) because its overall probability of beating the market (S&P 500) in the next 3 months (37.30%) is -1.65% vs. the average probability (38.95%) of US stocks	Adobe (ADBE) has an AI Score of 6/10 (Hold) because its overall probability of beating the market (S&P 500) in the next 3 months (40.30%) is +1.35% vs. the average probability (38.95%) of US stocks
Sum	42	45	53

Table 4: Choosing Stock Calculation Sheet

Adobe, with the highest score sum, is the chosen option. Below is a chart displaying its performance over the past year. The vertical axis displays the stock price in EUR and the horizontal axis represents the time.

Figure 2: Adobe Chart (`google.com/finance`, accessed: 13.10.2023)

2.5.2 API Provider

Alpha Vantage was chosen for the stock market prediction project primarily due to its diverse and comprehensive financial data offerings. Despite the limit of 25 free daily requests, an large amount of data is provided for the project's needs including: real-time data, technical indicators, and fundamental data. The platform's ease of integration and clear documentation were crucial factors in the choice, enabling quick access and utilization of the data within predictive models.

The importance of access to various news sources for complete information was decisive for the choice of `newsapi.org`. Recognizing the significance of gathering news from varied outlets, the platform offers a wide spectrum of sources, enabling access to multiple perspectives and diverse content. This variety of sources increases the depth and range of information available for analysis and insight. The site gives unlimited free queries from one month in the past to now.

The following tables shows all the APIs, the provider and the data size. The total folder size from one day of collecting data is around 1.3 Mega Byte (MB).

Name	Source	Update Time	[File Size]=KB
News about Adobe Inc.	Newsapi.org	Daily	4
News about Shantanu Narayen (CEO of Adobe Inc.)	Newsapi.org	Daily	4
Stock Prices	Alphavantage.co	Daily	1
News about ADBE	Alphavantage.co	Daily	18
Currency Rate USD to JPY	Alphavantage.co	Daily	1
Retail Sales	Alphavantage.co	Monthly	17
CPI	Alphavantage.co	Monthly	55
Unemployment Rate	Alphavantage.co	Monthly	36
Earnings	Alphavantage.co	8x yearly	20
Company Overview	Alphavantage.co	Quaterly	2
Income Statement	Alphavantage.co	Quaterly	22
Balance Sheet	Alphavantage.co	Quaterly	34
Federal Funds Rate	Alphavantage.co	Yearly	1033
real GPD USA	Alphavantage.co	Yearly	4
Inflation	Alphavantage.co	Yearly	4
		total	**1255**

Table 5: API Overview

2.5.3 Write Code to Collect Data

The Python script utilizes several libraries, including "requests", "json", "datetime", "re", and "os", along with custom configurations from an extra file. The primary function of this script is to gather financial data and news from various APIs, storing the obtained JSON responses in files organized by date.

The script begins by printing essential information for user reference, such as the date of data collection, stock details, and other relevant parameters as seen in Figure 3.

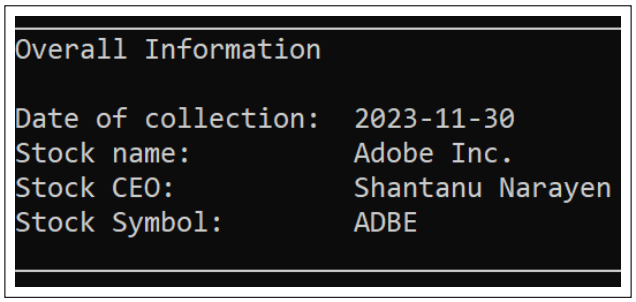

Figure 3: Data Collector Program Output 1

It sets up a directory structure and prepares a list of URLs pointing to different APIs providing financial data and news related to the stock and market. The API-Keys and other Parameters the program needs are stored separately in a configuration file so they can be changed without editing the source code.

Following this, the script iterates through each URL, initiating API requests via "requests.get()" and handles the corresponding JSON responses.

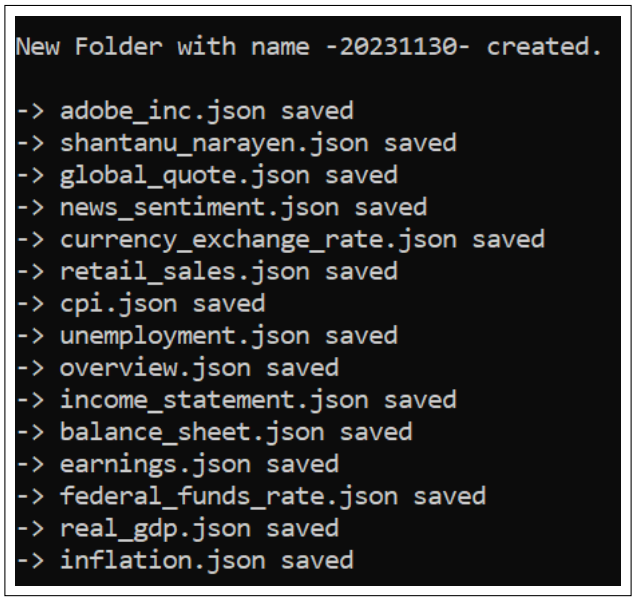

Figure 4: Data Collector Program Output 2

Each day's data is organized into folders named according to the YYYYMMDD syntax, signifying the precise date of data collection. This structuring method optimally maintains data organization and facilitates seamless sorting based on date, ensuring consistent chronological arrangement even after potential modifications or updates.

The significance of displaying the output in the console lies in providing users with real-time feedback about the script's progress, ensuring visibility into the data collection process, and offering insights into any encountered errors or limitations due to API constraints. This transparency helps users monitor the execution and status of the data collection procedure, facilitating informed decision-making and troubleshooting if necessary.

2.5.4 Server setup

The server setup for daily automated execution of the Python script was initiated using a Pi, chosen for its low power consumption. An bundle with all the needed parts including heat sinks and cables is used.

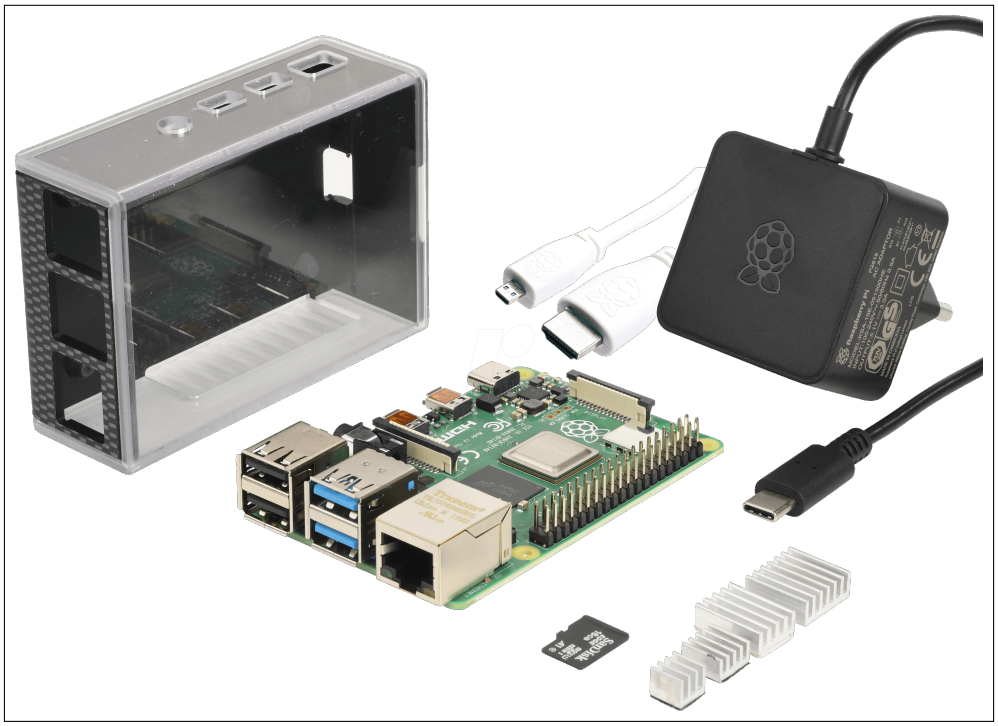

Figure 5: Raspberry Pi 4 Bundle (`reichelt.de`, accessed: 26.11.2023)

The Pi is connected to an external 1 TB SSD. Debian was installed on the Pi, providing a stable operating system suited for the device's architecture. The code was transferred to the device, preparing it for scheduled execution. A ".timer" and ".service" file were created to enable systematic scheduling via systemd.

The coordination of the data collection program was facilitated by the ".service" file.

Upon activation, the service initiated the Python script, enabling the execution of numerous API requests and the meticulous collection and storage of data on the local drive of the Pi.

2.5.5 Redundancy and Checkup

After developing the initial program, several measures were implemented to enhance its robustness and data safety. To address potential API errors, an error handling mechanism was integrated into the program, ensuring graceful handling of any encountered errors during API requests. Additionally, the program was augmented to retrieve and display the cumulative file size of the day's data, providing a comprehensive overview of the data collected.

Recognizing the significance of data redundancy, an auxiliary program was designed to copy all the collected data to an external SSD, mounted to the Pi. Leveraging a the "shutil" libary, this program verified existing data on the SSD and performed a complete data tree copy, effectively establishing redundancy for all collected information.

Furthermore, to maintain a transparent and informative process, an email notification system was established. The data collection program gives information about the API requests and the data size and writes it into an .txt file, the backup program confirms the backup and adds the information to the same text file. A Python script utilizes the "EmailMessage" module to compose an email. It reads the content from the text file mentioned before and sets up an email message, defining sender, recipient, subject, and content. Using Simple Mail Transfer Protocol (SMTP) and Secure Sockets Layer (SSL) from "smtplib" libary, it establishes a secure connection with Gmail's SMTP server over port 465, logs in using the sender's email credentials, and sends the email to the specified recipient.

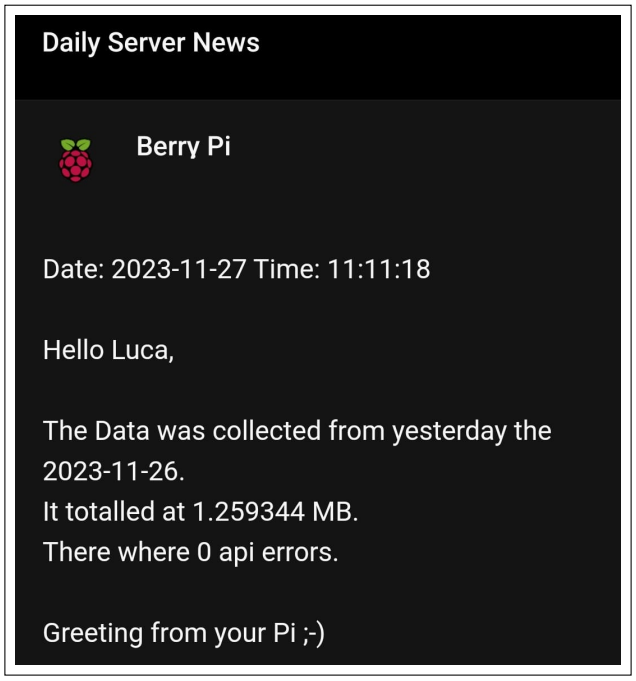

Figure 6: Example Server Mail

To run all the programs one after another in the right order a main file is made, that starts the program inside of a Virtual Environment (venv).

2.5.6 Overall Program Plan

The table shows all the programs and files implemented on the server to collect and save the data properly.

Program names	my_main.timer	my_main.service	my_main.py	my_dc.py	my_backup_to_ssd.py	my_ms.py
Descriptions	-Linked to calendar -Triggered daily at 11:11 p.m	-Triggered by my_main.timer -then runs my_main.py	-Runs all scripts inside venv	-Collects data -Saves JSON -Writes mail body file	-Makes backup on ssd -Copies whole tree	-Reads mail body file -Sends mail to user
Dependencies	OnCalender	my_main.timer	dc.py my_backup_to_ssd.py my_ms.py	my_config.py my_email_body.txt	Collected_Data Local Collected_Data SSD	my_config.py my_email_body.txt
Order	1-->	2-->	3-->	4-->	5-->	6!

Table 6: Program order, Dependencies and Descriptions

The timer is run at 11:11 p.m. and runs the service file. This file starts the main program that runs one script after another inside the venv. When data collection is finished a mail is sent as seen in Figure 6.

3 Conclusion and Outlook

The stored data will be collected till the second part of the project starts in five months. Training data will then be available for almost six months(=180 data-points). An analysis will be conducted on all folders, gathering data to compile into a comprehensive CSV file as specified in the morphological box. To address the intermittent data gaps, a linear interpolation will be implemented between consecutive points to mitigate abrupt data variations. Subsequently, an evaluation of the daily news using FinBERT will generate news scores across various topics. The overarching goal encompasses creating a substantial dataset wherein each day constitutes a distinct data point inclusive of all samples. Financial ratios will be derived from this dataset, followed by training an LSTM Recurrent Neural Network (RNN) model to forecast stock prices at varying future intervals, such as one day, one week, and one month ahead.

In the realm of AI, having quality training data stands out as the most crucial step, and currently, that cornerstone is under construction having the Pi running every day to collect data. Alongside this, meticulous planning of methodologies and gaining a comprehensive overview of the literature has been pivotal. This process sets the stage for a deeper exploration of the subject and provides a clear direction for steering the project towards its intended goals.

References

[1] Benjamin Graham. *The intelligent investor: The definitive book on value investing*. HarperCollins, 1934.

[2] CFI Team. *Financial Ratios - Complete List and Guide to All Financial Ratios*. https://corporatefinanceinstitute.com/resources/accounting/financial-ratios/. Accessed: 2023-11-28. 2023.

[3] Nico Litzel. *Big Data, SQL und NoSQL – eine kurze Übersicht — bigdata-insider.de*. https://www.bigdata-insider.de/big-data-sql-und-nosql-eine-kurze-uebersicht-a-602249/. [Accessed 07-12-2023].

[4] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN: 9781999579517. URL: https://books.google.de/books?id=0jbxwQEACAAJ.

[5] M.-A. Mittermayer. "Forecasting Intraday stock price trends with text mining techniques". In: (2004). DOI: 10.1109/HICSS.2004.1265201.

[6] Robert P Schumaker and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system". In: *ACM Transactions on Information Systems (TOIS)* (2009).

[7] Anshul Mittal. "Stock Prediction Using Twitter Sentiment Analysis". In: (2011). URL: https://api.semanticscholar.org/CorpusID:9097723.

[8] Jasmina Smailović et al. "Predictive sentiment analysis of tweets: A stock market application". In: (2013).

[9] Tomer Geva and Jacob Zahavi. "Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news". In: *Decision support systems* (2014).

[10] Rakhi Batra and Sher Muhammad Daudpota. "Integrating StockTwits with sentiment analysis for better prediction of stock price movement". In: (2018).

[11] CK Vignesh. "Applying machine learning models in stock market prediction". In: *IRJET, Journal* (2018).

[12] Rui Ren, Desheng Dash Wu, and Tianxiang Liu. "Forecasting stock market movement direction using sentiment analysis and support vector machine". In: *IEEE Systems Journal* (2018).

[13] Xiao Zhong and David Enke. "Predicting the daily return direction of the stock market using hybrid machine learning algorithms". In: *Financial Innovation* (2019).

[14] Dogu Araci. *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. 2019. arXiv: 1908.10063 [cs.CL].

[15] Fernando GDC Ferreira, Amir H Gandomi, and Rodrigo TN Cardoso. “Artificial intelligence applied to stock market trading: a review”. In: *IEEE Access* (2021).

[16] Jihwan Kim, Hui-Sang Kim, and Sun-Yong Choi. “Forecasting the S and P 500 Index Using Mathematical-Based Sentiment Analysis and Deep Learning Models: A FinBERT Transformer Model and LSTM”. In: *Axioms* (2023). ISSN: 2075-1680. URL: https://www.mdpi.com/2075-1680/12/9/835.