EWI3615TU

Design and Validation of Stock Market Prediction Tool for Oil Companies

Group 02

Delft University of Technology

EWI3615TU

Design and Validation of Stock Market Prediction Tool for Oil Companies

by

name (0000000),
name (0000000),
name (0000000),
name (0000000),
name (0000000)

TU Delft

Contents

1 Overview

1.1. Purpose

The purpose of our project is to predict future closing values for the stock market of oil companies. The daily stock market data that was analysed and processed consists of six variables, namely: the opening value, peak value, minimum value, closing value, adjusted close value and volume. In addition to this, data on oil price fluctuations, as well as data on the EUR to USD conversion was used. The developed predicting tool delivers as output closing values of different oil companies stocks for up to 30 days into the future.

1.2. Process and challenges

The design process began with the generation of requirements and constraints to guide our project. Furthermore, risk factors were identified, the primary ones including the availability of sufficient data to produce a valid prediction and adapting the webscraping algorithm to be compatible with different websites.

Along the way, it was observed that a large range of websites would introduce too many variations of data organization and formatting, which then would make using a single algorithm to scrape from all of them difficult. To add to this challenge, in the beginning it was planned that future closing values would be predicted for a wide range of companies. However, it was shortly concluded that this goal would be too complex to achieve within the allocated timeframe for this project. In order to surpass this challenge, it was agreed upon that the final product would only deliver predictions for oil companies. This led to a simplification of the webscraping process. The webscraping was performed on only three websites, as there were three types of data that needed to be formatted and consequently fed into the machine learning algorithm, namely:

- Historical data on stock market values for the desired oil companies, retrieved from Yahoo! Finance (`https://finance.yahoo.com/`)
- Data on the fluctuation of oil prices, retrieved from Nasdaq (`https://www.nasdaq.com/`)
- Data on the conversion of EUR to USD, retrieved from the Deutsche Bundesbank (`https://www.bundesbank.de/de`)

The machine learning algorithm was then used to train and test the webscraped data, ultimately predicting the closing values of different oil company stocks a few days into the future. Supervised regressive machine learning was used. 80% of the scraped data was used to train the machine learning algorithm and the remaining 20% was used to test the validity and reliability of the predictions of the machine learning algorithm.

The final step of the process consisted of implementing a GUI which would let the user choose an oil company and then would showcase both the historical stock market data for said company and the predictions for the next few closing values provided by the machine learning algorithm.

1.3. Tools

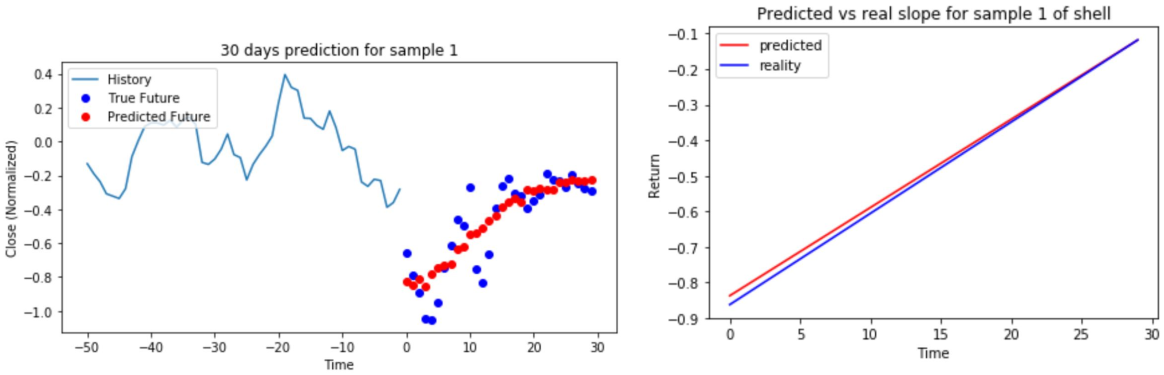
As the team were new to webscraping, it therefore was logical to use tools and packages with plenty of online support and examples to guide both understanding and code implementation. It was decided to use the Selenium package in order to automatically navigate to the correct web page and click the relevant buttons to scrape the site for data. Selenium Webdriver also supports many different browsers and can run multiple scripts. Once the data has been found, Beautifulsoup4 was used to pull the data from the site into an Excel file. Beautifulsoup4 was used because it enables a neat way of finding and formatting data using table headers. A demonstration of our webscraping process can be found here: `https://www.youtube.com/watch?v=zdnqtUsUirg`.

The developed machine learning algorithm lays in the category of time-series forecasting. Thus, it was decided that a tuned Recurrent Neural Network was to be implemented. The RNN used in this case is part of a newly released TensorFlow package.

The Python micro web application framework, Flask, was used in order to implement the GUI. The separate webpages have been created using HTML5 and JavaScript. A visualisation of the GUI may be found in Figure 4.4 and Figure 4.5.

1.4. Results and Evaluation

The results obtained after training the machine learning algorithm can be found in Figure 1.1a and Figure 1.1b. Validation samples, extracted from the 20% of the total scraped data, which represented the testing batch, were used in order to test the accuracy of the predictions. As can be seen, the predictions follow the general trend of the true data. More results can be found in Figure A.3 and Figure A.5. Furthermore, the accuracy of the prediction in the case of the Royal Dutch Shell company may be observed in Figure A.4, where the average error distribution of the closing value over four different validation samples is displayed.



(a) Predicted versus true closing values for the company Shell for 30 days.

(b) Slope and intercept obtained from applying linear regression to predicted and real values, over one validation sample extracted from the scraped data on Shell.

Figure 1.1: Obtained results.

Another method of evaluating and validating the results consisted of a test that was conducted in order to decide whether the set of parameters used for training, presented in Table A.1, truly determined the algorithm to predict the future closing values of oil companies more accurately than those of other types of companies. Thus, data from three oil companies, as well as The Walt Disney company, were fed into the algorithm. The validation loss factor in the case of the oil companies was up to 9.5 times smaller than in the case of Disney. This confirms that the algorithm is more responsive and indeed delivers more accurate predictions for oil companies. A more extensive explanation of the test may be found in Chapter 5.

2

Introduction

Data has become a valuable commodity in current society. It has become increasingly important to be able to extract pertinent information from reliable sources. The acquired raw data needs to be manipulated and analysed in order to provide useful patterns from which to draw conclusions about the research topic. This project will focus on using web-scraping principles and simple machine learning algorithms to analyse oil companies on the stock market.

The stock market is a public market that is used to buy, sell and issue shares of publicly listed companies. The growth and decrease of the stock market can have a profound impact on companies, employees and investors. Being able to predict the trend of the stock market may save companies and investors many millions of dollars, and save employees from potential job losses [5], [4].

This report represents the documentation set of our project, built during its entire development phase. Regarding the structure of this document, it will first present the development plan created by the team. After this, the design and implementation phase will be described. Next, the testing, validation and the evaluation phase will be discussed. Finally, access to the code documentation will be made available via the team's code repository.

3

Development Plan

3.1. Project Objectives and Context

The primary objective is to predict future closing values for the stock market of companies. This is important as fluctuations in the stock value of large companies have a significant impact on the local economy. Furthermore this would aid in predicting job security and financial losses for investors.

In order to achieve this, data has to be extracted from the web and then sorted into data sets and fed into a machine learning algorithm. The daily stock market data that will be analysed will consist of six variables, namely: the opening value, peak value, minimum value, closing value, adjusted close value and volume.

3.2. Requirements and Constraints

The requirements for this project have been formulated and prioritized using the MoSCoW method. Thus, the software built by the end of this project:

MUST

1. Predict stock value of a company for one day into the future
2. Provide some form of visual representation of the stock value fluctuations in the past, followed by any predictions made
3. Perform predictions of a usable, reasonably high fidelity

SHOULD

1. Enable users to view predictions for multiple different companies
2. Incorporate factors beyond the core variables listed in Section 3.1 in the prediction procedure
3. Compare the fidelity of predictions made including and excluding the aforementioned additional variables.

COULD

1. Predict stock value of a company for up to a week into the future
2. Let the users add their own data sets for existing companies, concerning new variables

WON'T

1. Send notifications/e-mails to the users to announce when it would be the most profitable to invest in a certain company

2. Provide information regarding the prediction accuracy rates for the past few days

Furthermore, the main constraints and limitations that can occur are:

- Difficulty in accounting for political environment and its impact of stock value functions
- Predicting global financial crashes
- Predicting internal company dynamics
- Lack of knowledge regarding internal company initiatives and reaction of the public market to said initiatives.

3.2.1. Risk factors, 'to-avoids'

There are many factors that can influence the functionality and usability of the data that is collected and the validity of any predictions that will be made. It is therefore important to be able to identify and avoid any factors that can negatively affect the process.

The following risk factors have been specifically chosen as factors that will be avoided during the development process:

1. Scraping too many websites for data. This is because scraping too many websites can mean having to manipulate data of different sizes and formats and doing too many can be too time consuming and make the data too complex.
2. Using unrelated variables. The data collected will eventually be used to make predictions. For the predictions to be effective the variables should be strongly related.
3. Not using enough data points. In order to train machine-learning algorithms a lot of data points are needed to get reliable results.
4. Not manipulating the data before drawing conclusions. Raw data alone does not provide enough information. It must first be made into a format that can be visually understood and used in the machine-learning algorithms.

3.3. Project Approach, Resources, Priorities

The approach that will be taken in this project is to first develop an algorithm that will be used to scrape relevant data from the web (mainly using the Yahoo Finance website: `https://finance.yahoo.com/`).

In parallel, the architecture of an algorithm intended to be trained, using machine-learning, is developed; the algorithm will be trained by feeding it batches of data, formatted into .csv files, with the intention of teaching it to predict stock value fluctuations. The main priority is to produce a prediction for a given company's stock value with a reasonably high fidelity.

3.4. Design objectives, design strategy and validation

During the design phase of our project the tools that we will be using in order to perform web scraping, as well as the main approach of collecting and storing the data into useful batches will be decided upon. Furthermore, the main architecture of the machine-learning algorithm will be created and documented. In addition to this, the main features of the user interface, as well as approaches on how to achieve delivering these features will be discussed. Before starting the implementation, a validation phase will be carried out. The main goal of this phase is to evaluate whether all previously formulated requirements will be satisfied by the proposed design. This will take place in week 2.3.

3.5. Implementation planning, managed

After the design phase has been concluded, the implementation may begin. The implementation phase will consist of 3 main activities: building the web scraping algorithm, building the machine-learning algorithm and building the user interface.

Web scraping is going to be executed by using a combination of the Selenium and BeautifulSoup tools. The web scraping process will consist of building an algorithm which will extract the relevant data from the web (see Section 3.1), organising it into batches and storing them as csv files. This process will start in week 2.3, but it is aimed to be finished in week 2.5.

Next, the data batches will be fed into the machine-learning algorithm whose skeleton was settled upon in the previous phase and the output of the algorithm will be analysed. The implemented machine-learning architecture will follow that of a Supervised Learning algorithm which is motivated by a regressive task; this means that the outcome of the learning process is a float value, obtained based on the regression constructed based on historical data. The research on how to build such an algorithm, as well as the development of its main architecture will start in week 2.3. However, reliable data will be fed into it after the data batches have been extracted, sorted and formatted, so in week 2.5. Furthermore, it is expected that the algorithm will produce the desired output, which may be used for testing, in the beginning of week 2.7.

The user interface should provide a way for the user to visualise the fluctuations in the stock market, as well as the prediction provided by the machine-learning algorithm for the following day, by means of a graph. The implementation of the user interface will take place parallel to the testing phase, in week 2.7.

For a better visualisation of the implementation process, an activity diagram is displayed in Figure 3.1 .

3.6. Testing approach, test planning, validation reporting

Regarding the testing, 80% of the web scraped data will be used to train the machine-learning algorithm. The rest of the data will be split into two sets, each representing 10% of the data. The first 10% will be used to test the prediction model generated, while the other half will be used for validation purposes. The performance of the algorithm will be measured using the Root Mean Square Error (RMSE) which will measure the standard deviation of the errors that the system makes when calculating the predictions [3].

The validation stage also concerns comparing our product with the requirements generated at the beginning of the project, and thus analyzing the success of our product.

During the testing and validation phases, the model can still be subject to changes. The testing and validation activities are planned for weeks 2.7 and 2.8.

3.7. Evaluation

The final phase of the present project will be evaluation. During this phase, an analysis of the goals achieved during the project, as well as the completed objectives and activities will be performed, in order to decide whether all the requirements have been met and whether our project has produced the planned results. In order to do this, a compliance matrix will be created, in which all the requirements will be displayed, together with a statement which describes if and how they have been satisfied during the development process. By analysing this matrix, conclusions about the success of the project may be drawn.

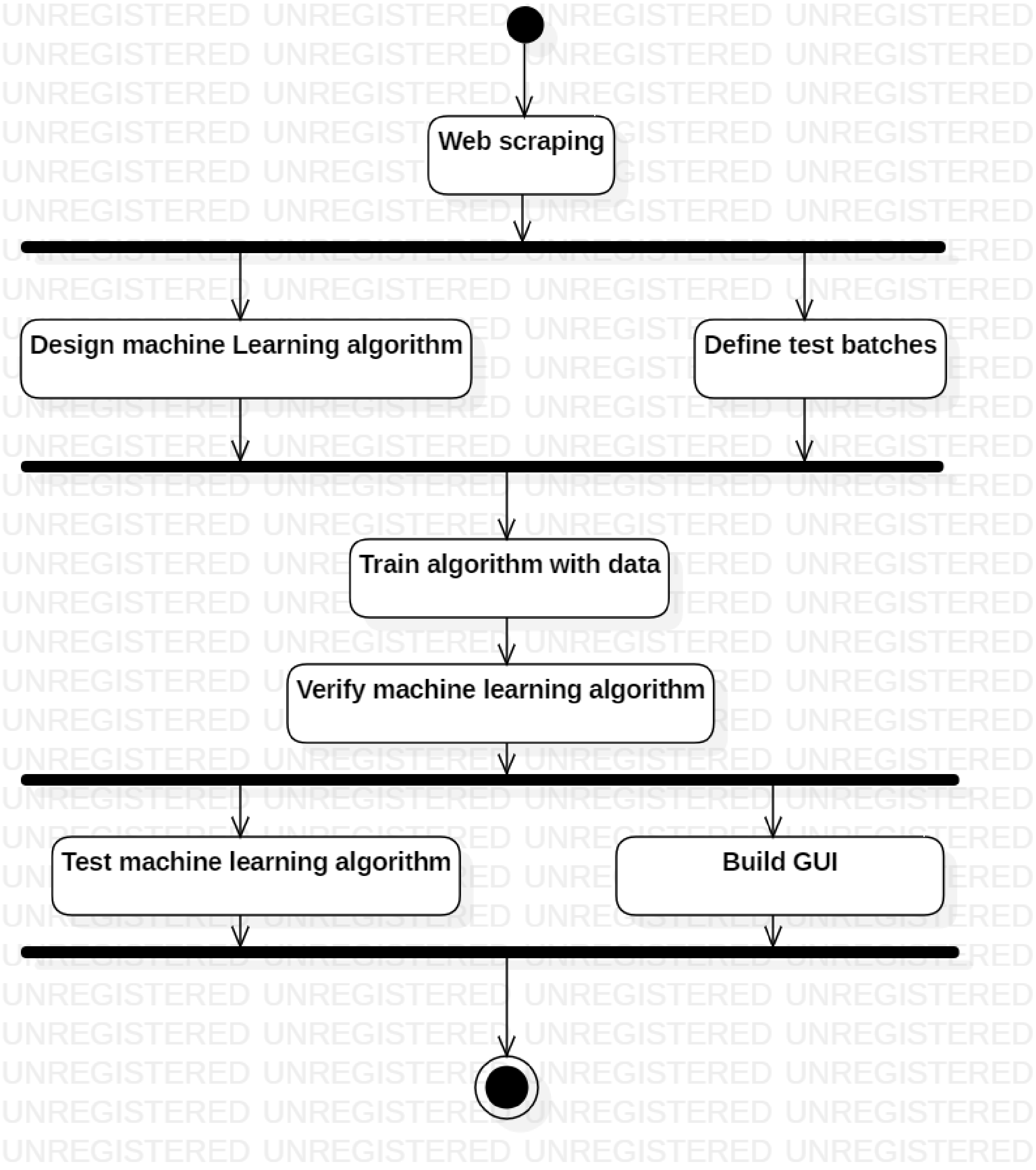

Web scraping
Design machine Learning algorithm
Define test batches
Train algorithm with data
Verify machine learning algorithm
Test machine learning algorithm
Build GUI

Figure 3.1: Activity diagram displaying the main flow during the implementation process.

4
Design and Implementation

The aim of this chapter is to present the decisions made during the design phase of our current project. First, the design process of the web scraping algorithm will be described. Then, the main steps taken towards the development of the machine learning algorithm will be discussed. Finally, the design validation phase will be presented.

4.1. Web Scraping Algorithm

The main objective of the web scraping algorithm is to extract data from three websites:

- Yahoo! Finance (`https://finance.yahoo.com/`), concerning the daily stock market fluctuations
- Nasdaq (`https://www.nasdaq.com/`), concerning the fluctuation in oil prices
- Deutsche Bundesbank (`https://www.bundesbank.de/de`), concerning the fluctuation in the conversion of EUR to USD

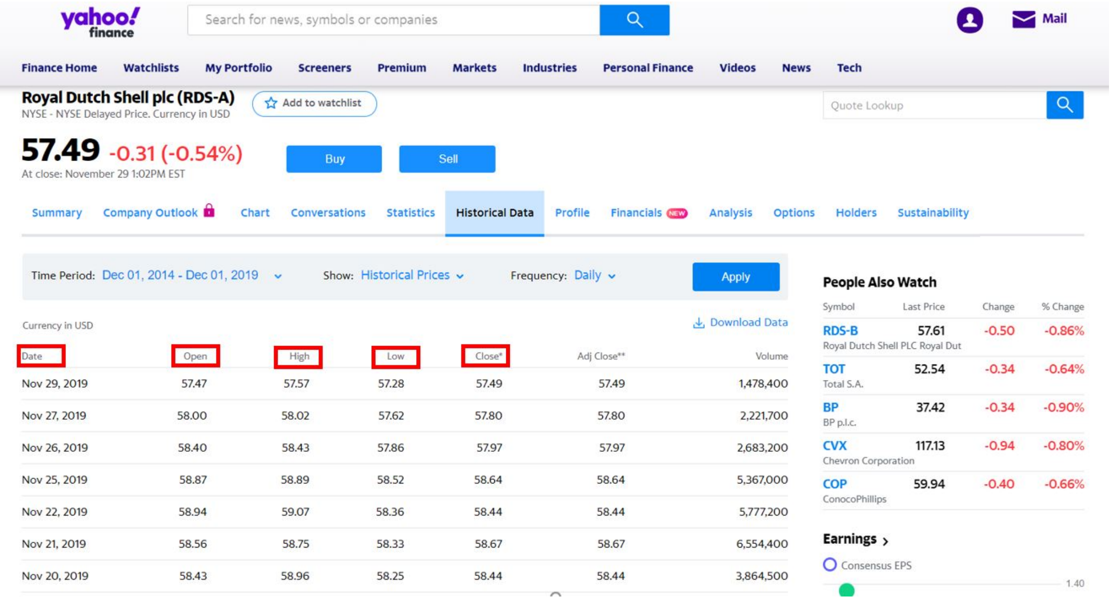
After the data was acquired, it will be formatted into an accessible csv file and stored.

In order to reach the aforementioned objective, it was decided that two main tools will be used in the implementation process:

- **BeautifulSoup** - a Python package that is useful for parsing HTML documents. It works by creating a parse tree for parsed pages, which can be then used to extract data from a certain website
- **Selenium** - a framework used to automate a browser

In order to obtain the relevant data, the algorithm will first make use of the BeautifulSoup library. The design of the web scraping algorithm implies that the name of the company that we wish to collect and store data for will be given as an input by the user. Having this information, the algorithm will first open the Yahoo! Finance main page and search for the specific company. Next, the algorithm will navigate to the historical stock market data, which is displayed on the company's overview page. After this, a function will be used to update the time period of the data displayed on the web page, selecting the last 5 years as the desired time period. All of the navigation of the website will be done using the selenium library. Once the correct time frame has been chosen, the algorithm will extract the data displayed in the columns "Date", "Open", "High", "Low", "Close", as highlighted in Figure 4.1. All these data points will be saved into a xlsx file for further processing. The websites of Nasdaq and Deutsche Bundesbank will be scraped in a similar manner.

After the data is extracted, a function that will convert it to a csv file will be used. This function will also save the csv file, which represents the output of the web scraping algorithm, in order for it to be further processed. A visualization of the complete flow of the web scraping algorithm may be observed in Figure 4.2.



Royal Dutch Shell plc (RDS-A)
NYSE - NYSE Delayed Price. Currency in USD

57.49 -0.31 (-0.54%)
At close: November 29 1:02PM EST

Time Period: Dec 01, 2014 - Dec 01, 2019 Show: Historical Prices Frequency: Daily

Currency in USD

Date	Open	High	Low	Close*	Adj Close**	Volume
Nov 29, 2019	57.47	57.57	57.28	57.49	57.49	1,478,400
Nov 27, 2019	58.00	58.02	57.62	57.80	57.80	2,221,700
Nov 26, 2019	58.40	58.43	57.86	57.97	57.97	2,683,200
Nov 25, 2019	58.87	58.89	58.52	58.64	58.64	5,367,000
Nov 22, 2019	58.94	59.07	58.36	58.44	58.44	5,777,200
Nov 21, 2019	58.56	58.75	58.33	58.67	58.67	6,554,400
Nov 20, 2019	58.43	58.96	58.25	58.44	58.44	3,864,500

People Also Watch

Symbol	Last Price	Change	% Change
RDS-B Royal Dutch Shell PLC Royal Dut	57.61	-0.50	-0.86%
TOT Total S.A.	52.54	-0.34	-0.64%
BP BP p.l.c.	37.42	-0.34	-0.90%
CVX Chevron Corporation	117.13	-0.94	-0.80%
COP ConocoPhillips	59.94	-0.40	-0.66%

Earnings ›
○ Consensus EPS
1.40

Figure 4.1: Screenshot of the Yahoo! Finance website, from Shell's web page, showing which columns of data will be extracted in the web scraping process.

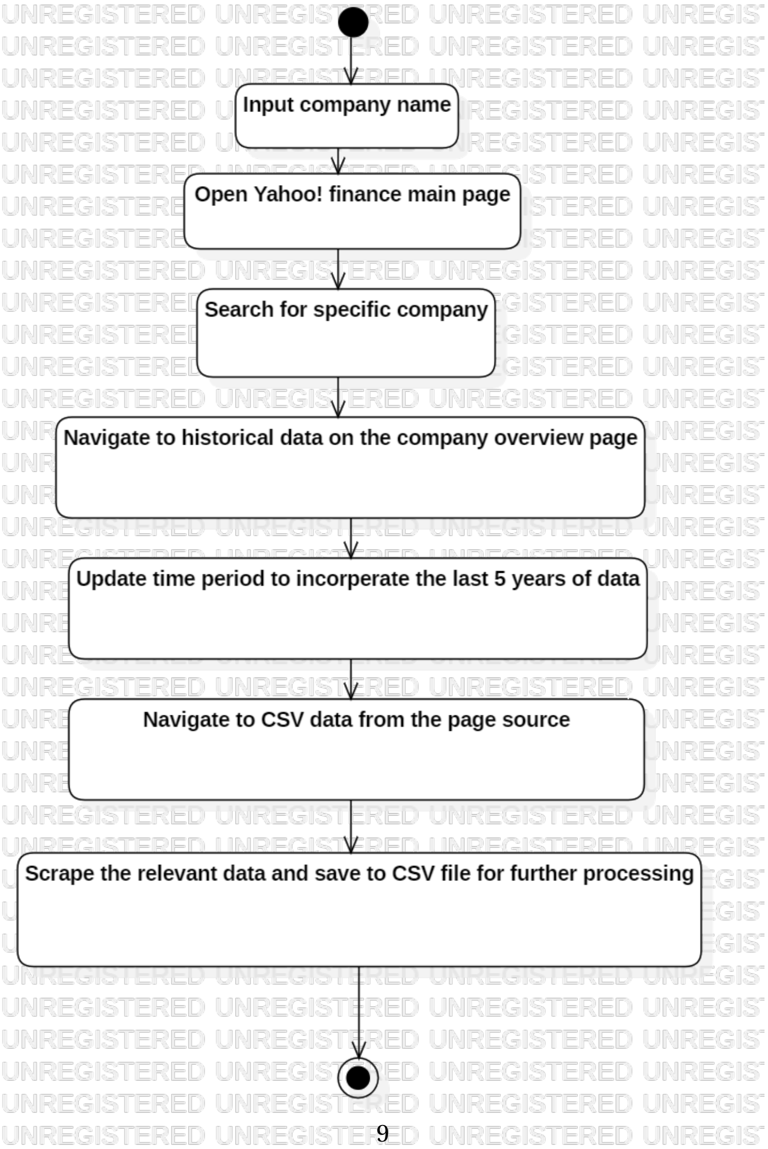

Input company name
Open Yahoo! finance main page
Search for specific company
Navigate to historical data on the company overview page
Update time period to incorperate the last 5 years of data
Navigate to CSV data from the page source
Scrape the relevant data and save to CSV file for further processing

Figure 4.2: Activity Diagram showcasing the flow of the web scraping algorithm.

4.2. Machine Learning Algorithm

The objective of implementing machine learning in this project is as follows:

- **Produce a prediction of the relevant company's stock closing value for the upcoming few days**
- **Produce a measure of the error of the delivered prediction**

4.2.1. An overview

Following from the web scraping component of the project, there would be a csv file with all required data to train the machine learning algorithm to produce the aforementioned prediction. The task at hand involves processing historical data and predicting future values.

An overview of the architecture of the machine learning algorithm is portrayed in the Class diagram found in Figure 4.3.

4.2.2. Implementation

The very first thing to consider when a good prediction is to be made is data exploration. A substantial part of the process must indeed be dedicated to the seeking of correlations between the event to be predicted and external factors. For this project, the team focused the study around European oil, gas and energy companies. The value to predict is the return (see Equation 4.1) of a single company for a specified amount of days in the future.

$$\text{Return}(t) = \frac{\text{Close}(t) - \text{Open}(t)}{\text{EuroValue}(t)} \tag{4.1}$$

It is agreeable that past return values would be the most useful factor to influence the algorithm, but it is to be added that it cannot be the only one, as the outcome we wish to predict is, in the real world, affected by a multitude of factors.

For this work, a research has been conducted on what can substantially affect the return trend of oil companies. Bianconi et al., in their study on non-renewable energy sectors [2], consider the price of crude oil and the change from USD to EURO as the two main causes of oil company return trends shape. The information came out to be true as a correlation has been found between these three factors (see Figure A.1 and Figure A.2).

The second step to follow is therefore choice of the algorithm that best suits the challenge. Since the tool lays in the category of time-series forecasting, the best possible choice would be to implement a tuned Recurrent Neural Network, or RNN [1] [6]. RNN's look at large collections of past values for an event, and based on that, they predict a defined number of information in the future. The neural network used for this purpose is part of a newly released TensorFlow package and it has been tuned with the parameters shown in Table A.1.

The final step before being able to use the tool consists in training it. Depending on the time available, multiple parameters can be tuned to choose the expected training time and the accuracy. Since the predicting tool developed is expected to give immediate result to the user, parameters have been set up to deliver a fairly accurate result for a reasonable amount of time, as 30 seconds are expected for the training of one company.

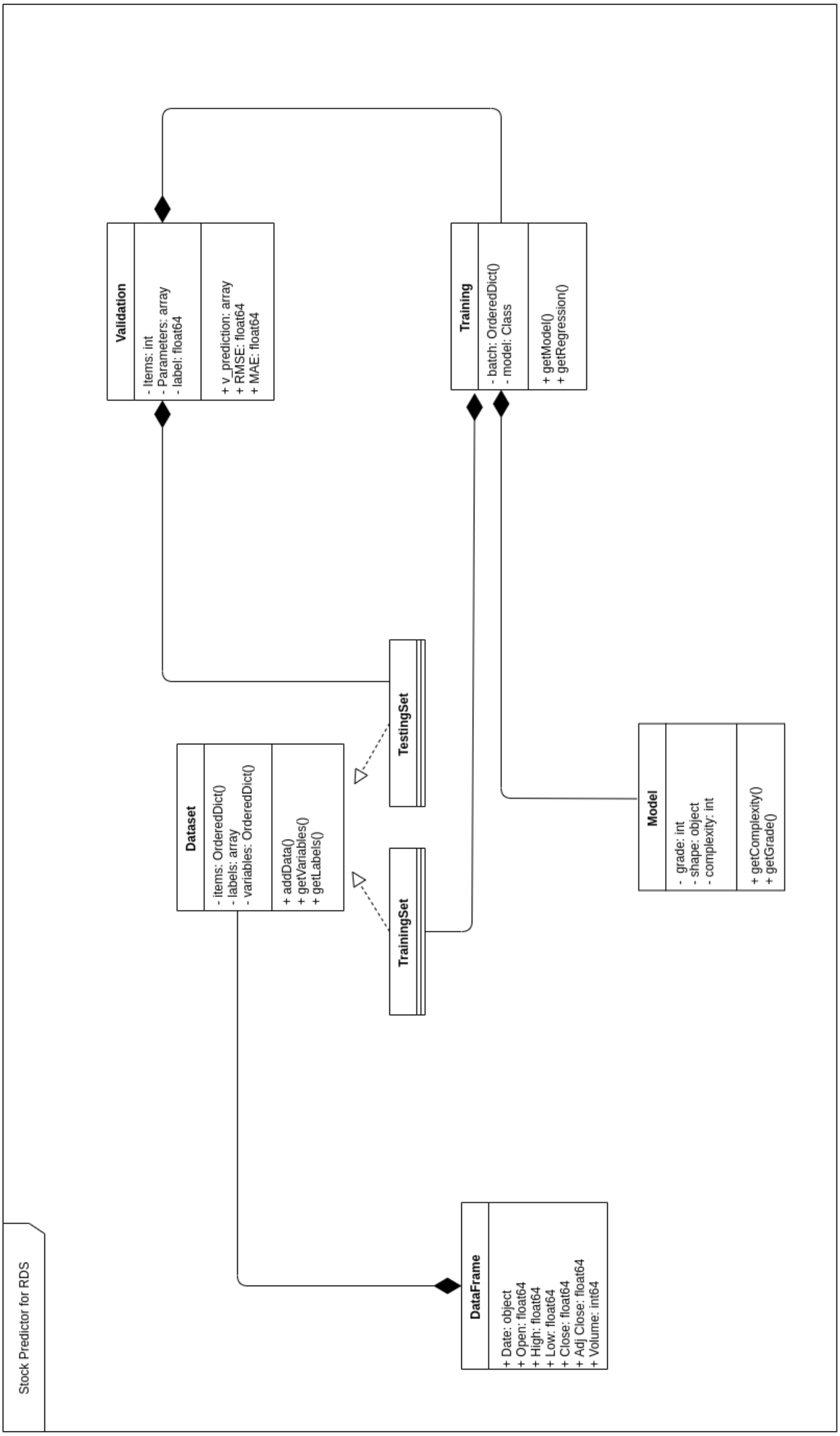
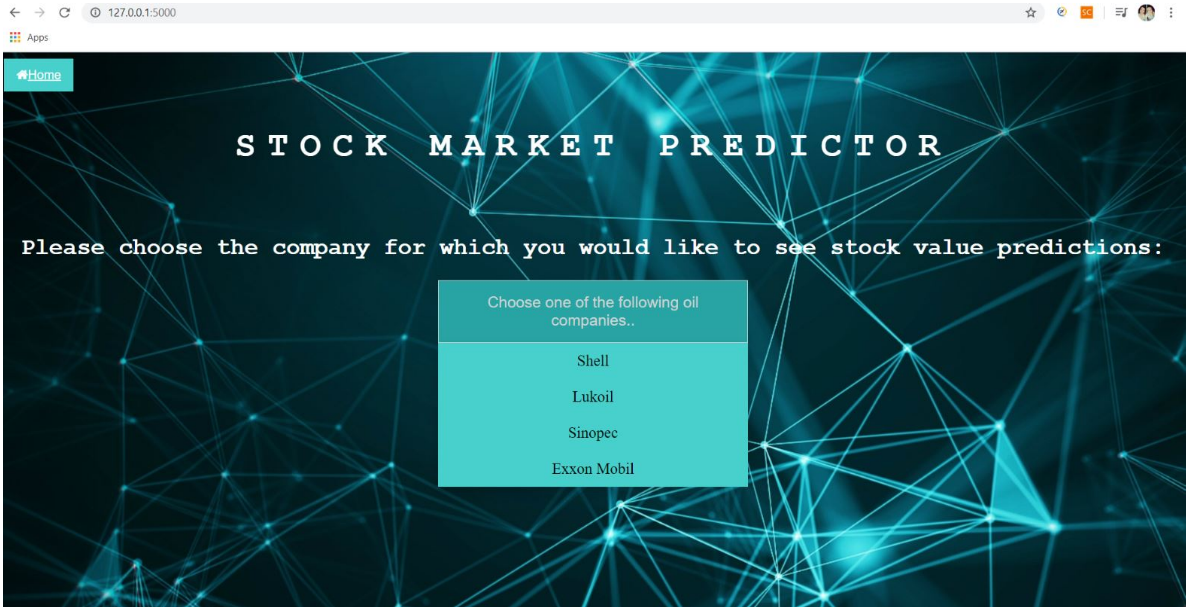


Figure 4.3: Class Diagram of expected prediction algorithm

4.3. Graphical User Interface

After the data was scraped and the machine learning algorithm used it to produce predictions, the results needed be made available to the user. It was decided that this will be done through a simple Graphical User Interface, which would allow the user to choose one of the available oil companies and then would display a line chart containing the historical stock market data for said company, as well as the future closing values, as predicted by the machine learning algorithm. The GUI was implemented using the Python micro web application framework, Flask. A visualisation of the GUI may be found in Figure 4.4 and Figure 4.5.

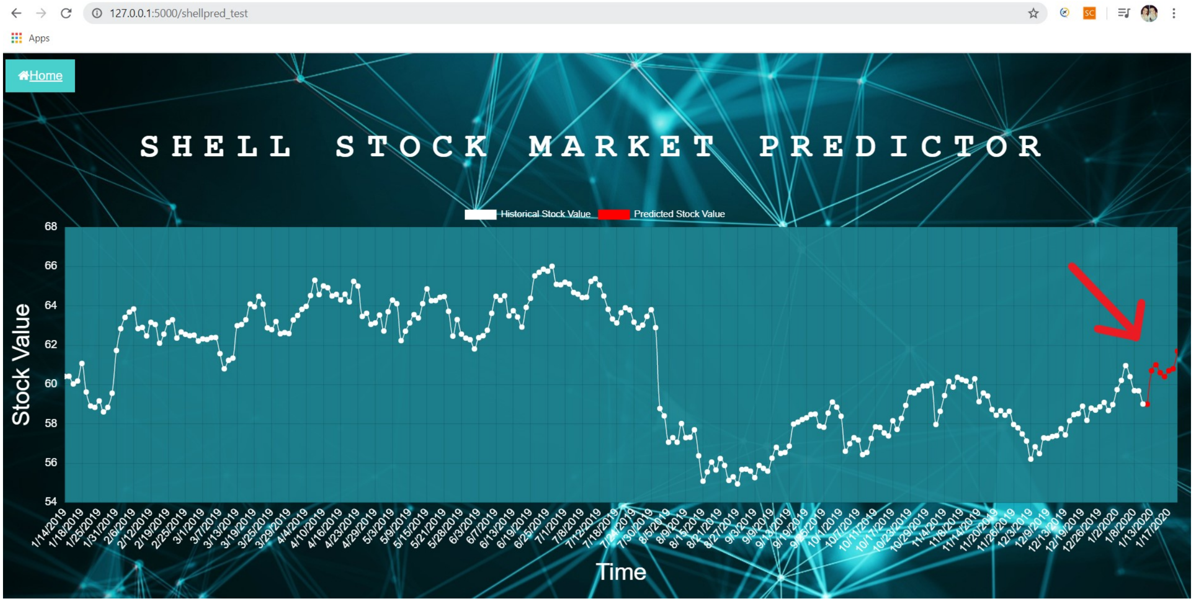


127.0.0.1:5000

STOCK MARKET PREDICTOR

Please choose the company for which you would like to see stock value predictions:

Figure 4.4: The Home page of the GUI, which contains a dropdown menu with different oil companies from which the user may choose.



127.0.0.1:5000/shellpred_test

SHELL STOCK MARKET PREDICTOR

Figure 4.5: The page for Shell, displaying a chart with the historical stock market data (white) and the predictions for the closing value (red).

5
Testing, Validation and Evaluation

The choice of factors for the study around oil companies turned out to be pertinent. The set of parameters used for training, presented in Table A.1, turned up in an expected training time of 30 seconds per company selected, a reasonable amount of time. The expected validation loss after the process is 0.45, while the training loss reaches 0.10. Such gap has been restrained as much as possible to avoid over-fitting.

Visually, the prediction accuracy can be appreciated in Figure A.3, where it was estimated over four validation samples, as well as in Figure A.4, where the absolute error distribution is shown.

However, the error distribution alone does not give a real overview of the accuracy of the tool. It is known that the most important information that a user would like to obtain concerns not only the daily fluctuation of the closing or opening stock value, but their long term trend, therefore if the value at day x is guessed correct but the slope obtained through an extended time frame differs greatly, the user cannot gain advantage from it. Luckily, the algorithm performed excellently in this task, as Figure A.5 shows.

In addition, to check the consistency of the tool and the validity of the factors, a "fake" oil company has been fed to the tool, and its outputs have been compared with a real oil company. The company used for this purpose is The Walt Disney Company. The dramatic increase in loss validation factor is showed in Table 5.1. It can be therefore said that the factors used for a more accurate training did actually help the RNN to build a taylor-made tool for energy companies.

Company Name	*Category*	*Validation Loss Factor*
Royal Dutch Shell	Oil and Gas	0.4431
Eni S.p.A.	Oil and Gas	0.6591
Total SA	Oil and Gas	0.6743
The Walt Disney Company	Enteirtainment	**4.2236**

Table 5.1: Increase in validation loss factor for companies not directly linked to factors selected for tool training

In addition to the evaluation of the success of the machine learning algorithm, an evaluation of the overall success of the entire project was conducted. In order to showcase this, a requirements compliance matrix is included in Table 5.2, as previously discussed in section 3.7 of the Development Plan. As it can be seen, the most important requirements initially imposed by the team have been met. Thus, we may consider this project a successful one.

Table 5.2: Compliance matrix for the initially formulated requirements.

Requirement	**Requirement met?**
Predict stock value of a company for one day into the future	✓
Provide some form of visual representation of the stock value fluctuations in the past, followed by any predictions made	✓
Perform predictions of a usable, reasonably high fidelity	✓
Enable users to view predictions for multiple different companies	✓
Incorporate factors beyond the core variables listed in Section 3.1 in the prediction procedure	✓
Compare the fidelity of predictions made including and excluding the aforementioned additional variables.	✓
Predict stock value of a company for up to a week into the future	✓

6

Code Documentation

GitLab link: `https://gitlab.ewi.tudelft.nl/ewi3615tu/2019-2020/data/ewi3615tu-ds2/ewi3615tu-ds2`
Webscraping screencast link: `https://youtu.be/zdnqtUsUirg`

Appendices

A

Supporting Material

A.1. Data Exploration

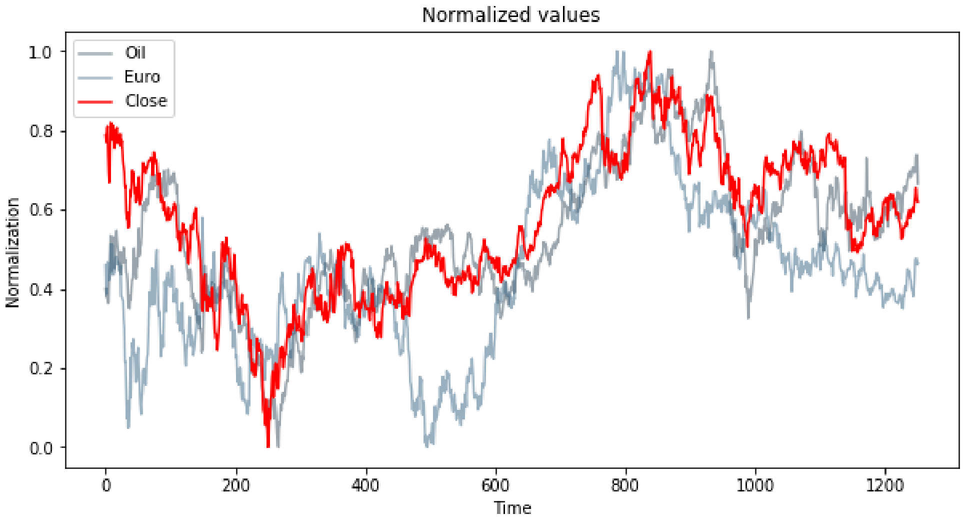
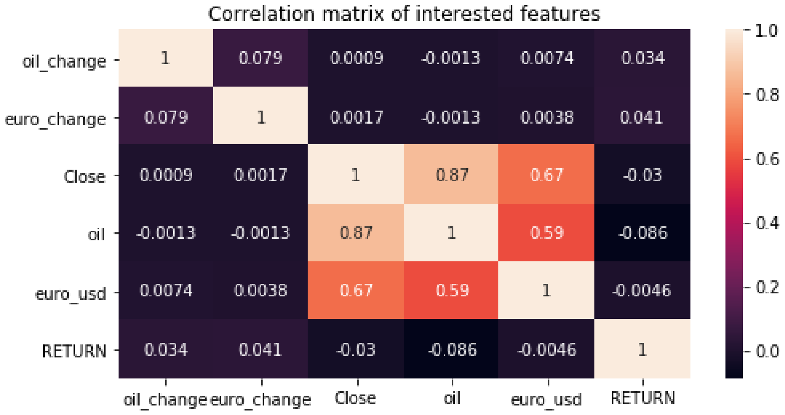


Figure A.1: Visual correlation between daily oil and euro change to dollar values and daily close value for Royal Dutch Shell, normalized



	oil_change	euro_change	Close	oil	euro_usd	RETURN
oil_change	1	0.079	0.0009	-0.0013	0.0074	0.034
euro_change	0.079	1	0.0017	-0.0013	0.0038	0.041
Close	0.0009	0.0017	1	0.87	0.67	-0.03
oil	-0.0013	-0.0013	0.87	1	0.59	-0.086
euro_usd	0.0074	0.0038	0.67	0.59	1	-0.0046
RETURN	0.034	0.041	-0.03	-0.086	-0.0046	1

Figure A.2: Correlation matrix shows a relation between daily oil and euro change to dollar values and daily close value for Royal Dutch Shell

A.2. Prediction Tool Design

Parameter	Setting
features considered	oil price, euro to us dollar change, closing price
dataset normalization	mean normalization
past time stamps considered for iteration	50
time stamps to be predicted, for iteration	30
batch size	100
buffer size	1000
RNN layer	Long Short Term Memory (LSTM)
optimizer	MAE
training epochs	10

Table A.1: Parameters settings for RNN used in the prediction

A.3. Testing and Validation

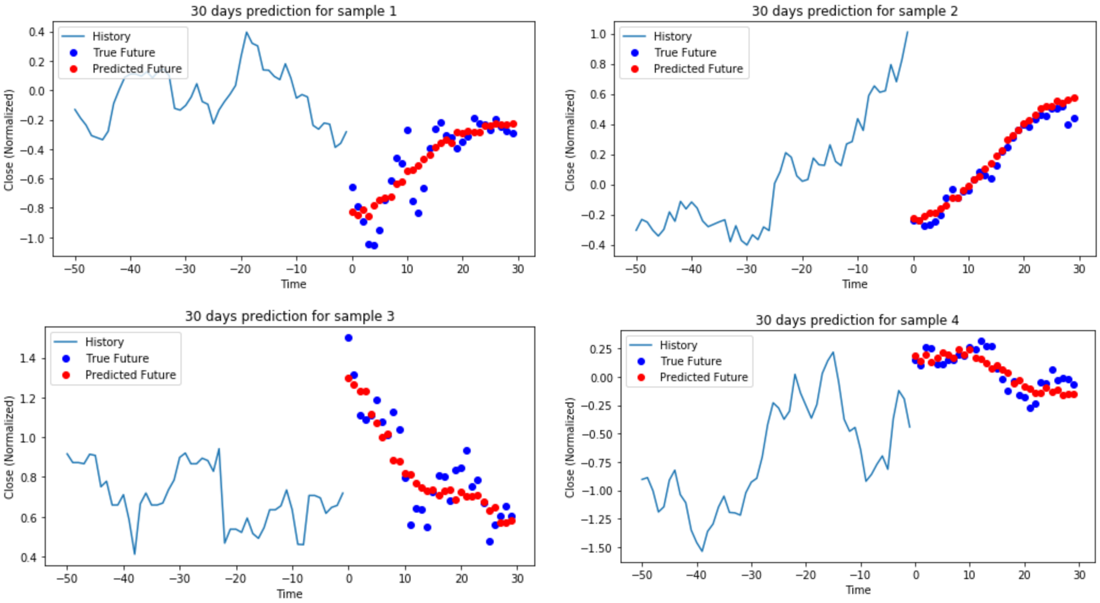


Figure A.3: Close value prediction for four validation samples

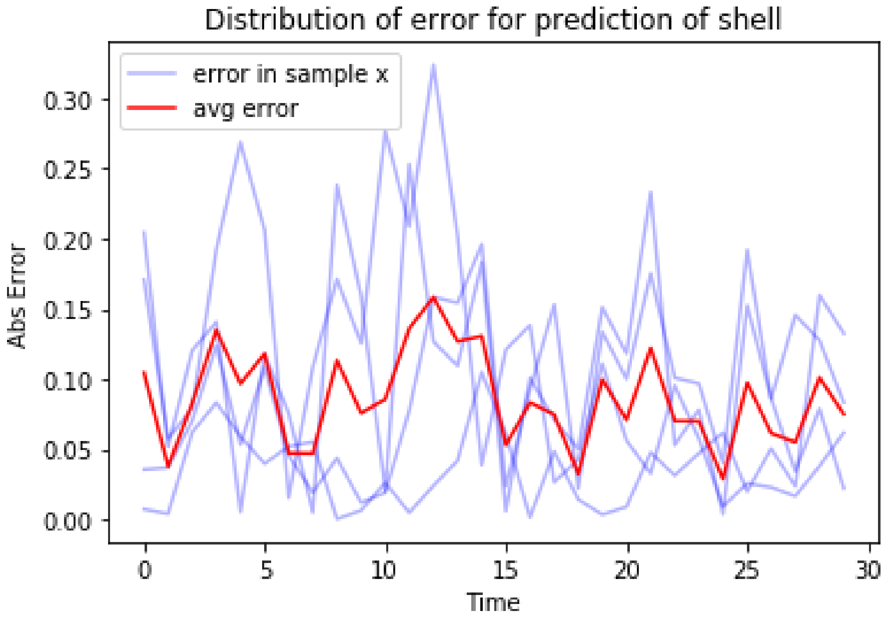


Figure A.4: Average error distribution of close value over four validation samples

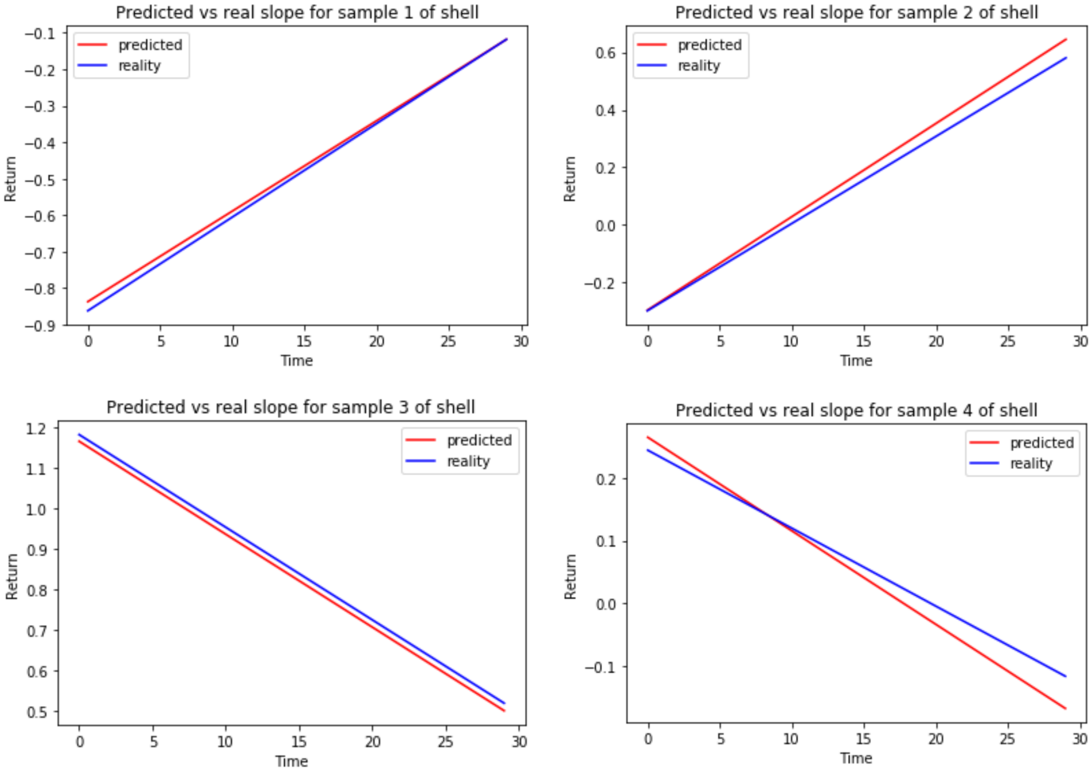


Figure A.5: Slope of and intercept obtained from applying linear regression to predicted and real values, over four validation samples

Bibliography

[1]

[2] Marcelo Bianconi, Risk Factors Yoshino, Joe Akira, and Vol. 45 2014. Available at SSRN: https://ssrn.com/abstract=2200526 or http://dx.doi.org/10.2139/ssrn.2200526 Value at Risk in Publicly Traded Companies of the Nonrenewable Energy Sector (January 13, 2013). Energy Economics.

[3] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Inc., 1st edition, 2017. ISBN 1491962291, 9781491962299.

[4] James Chen (Investopedia). Definition of 'Stock Market', 2019. URL `https://www.investopedia.com/terms/s/stockmarket.asp`.

[5] The Economic Times. Definition of 'Stock Market', 2019. URL `https://economictimes.indiatimes.com/definition/stock-market`.

[6] Hands-On Machine Learning with Scikit-Learn and Inc. Release Date: March 2017 ISBN: 9781491962282 TensorFlow by Aurélien Géron Publisher: O'Reilly Media.