

FINANCIAL DATA FORECASTER

Supervisor: Dr. Beta Yip

Shubhankar Agrawal (3035345306)

COMP4801: Final Year Project

FINAL REPORT

3 MAY 2020

**ABSTRACT**

Forecasting financial instruments has been a topic of interest in the world of research for a considerable number of years. The uncertainties in risk along with the potential for significant returns have lured people into financial data prediction. Over time, a wide range of methodologies and approaches have been implemented to analyse and forecast market movement. The ever-changing world of technology and finance constantly demands for newer strategies to obtain successful results for forecasting the market. This project aims to create a solution that will incorporate time series data of financial markets on a cross domain and international scale, along with news and public sentiment to obtain a better comprehension of the factors driving market movement and to predict them. Data extracted from sources online along with analysed sentiment of social media posts are used to obtain forecasts for market indices in a future time period. A wide range of machine learning models are employed and further optimized to obtain the most relevant parameters to improve the learning approach. The forecasted predictions are backtested against the market to generate significant positive equity returns thereby reinforcing the forecasting models. A project website has also been set up to describe the different aspects of the project. Following a proposed schedule, this project worked to deliver a forecasting methodology that can predict movement of cross-domain market indices using historical data with a high accuracy. This can additionally provide insight into the underlying factors influencing economies and help professionals in several domains.

**ACKNOWLEDGEMENT**

I would like to extend my gratitude to my project supervisor Dr. C L Yip for giving my colleague and myself the opportunity to work on this topic as our final year project. His assistance in navigating the complex field of machine learning while dealing with financial instruments was invaluable in enhancing the quality of our research and guiding us throughout the workflow of the project. I would like to thank my colleague Karan Mahajan for collaborating on this project with me.

**TABLE OF CONTENTS**

**ACKNOWLEDGEMENT II 5.**

**LIST OF FIGURES and tables V**

**. LIST OF ABBREVIATIONS VII**

**1. INTRODUCTION 1**

1.1 Background 1

1.2 Focus 1

1.3 Deliverables 2

1.4 Report outline 2

**2. LITERATURE REVIEW 2**

2.1 Previous Findings 2

2.1.1 Naïve Methods 2

2.1.2 Sophisticated Methods 3

**3. METHODOLOGY 5**

3.1 Software development practices 5

3.2 Project framework 5

3.3 Data collection 6

3.3.1 Numeric Data 6

3.3.2 Textual Data 7

3.4 Data cleaning and Pre-processing 7

3.4.1 Numeric Data 7

3.4.2 Textual Data 7

3.5 Exploratory data analysis and Feature Engineering 8

3.5.1 Exploratory Data Analysis 8

3.5.2 Feature Engineering 9

3.6 Algorithm implementation 9

3.6.1 Train Test Split 10

3.6.2 Parameter Selection 10

3.6.3 Train and Test Model 11

3.6.4 Evaluation Metrics 11

**4. PHASE 1 DELiverables 12**

4.1 Progress 12

**5. Data Collection and Pre-Processing 13**

5.1 Data gathering 13

5.1.1 Numeric Data 13

5.1.2 Textual Data 13

5.2 Sentiment analysis 15

5.3 Data Pre-processing and Feature Engineering 16

5.4 Challenges Faced 18

5.4.1 Data Collection 18

5.4.2 Data Pre-processing 19

**6. Experimentation 19**

6.1 Experimentation Approach 20

6.1.1 Regression Approach 20

6.1.2 Classification Approach - bins 24

6.1.3 Classification Approach - binary 25

6.2 Challenged faced 26

6.3 Experimentation Results 26

**7. Optimization 30**

7.1 Hyperparameter Selection 30

7.2 Walk Forward 31

7.3 Ensemble Voting 32

**8. BACKTESTING 33**

8. 1 Simple Moving Average Strategy 34

8. 2 Market Intraday Strategy 35

**9. Future Research 37**

9.1 Testing 37

9.2 Scalability 37

**10. Conclusion 37**

10.1 Work distribution 38

**APPENDIX 39**

**8. REFERENCES 41**

**LIST OF FIGURES**

Figure 1: Term Frequency - Inverse Document Frequency Algorithm. Extracted from [14] 3

Figure 2: Cross-Domain aspect of Forecasting indicating correlation within a domain, between domains and across a time series. Extracted from [15] 4

Figure 3: Seasonal Decomposition of data indicating observed data, trend, seasonal and residual components. 8

Figure 4: Iterative Machine Learning Process 9

Figure 5: Train test split and Cross validation 10

Figure 6: Evaluation Metrics - MAE, MSE, RMSE, R^2 11

Figure 7: Evaluation metrics - F1 Score 12

Figure 8: How Scrapy works, from sending a request, getting a response to parsing data and saving it in a database. Extracted from [11] 14

Figure 9: BERT Accuracy; increase in accuracy owing to the Masked LM feature of the model 15

Figure 10: Seasonal Decomposition of Hang Seng Index Returns Over 10 Years 16

Figure 11: Results for Augmented Dickey-Fuller Test Highlighting the Stationarity of Calculated Daily Returns 17

Figure 12: Correlation Matrix for Currencies Highlighting Correlations Discovered Over the 10 Year Period 18

Figure 13: Raw values regression results (BDT) 20

Figure 14: Regression Overfitting (BDT) 20

Figure 15: Returns regression results (BDT) 21

Figure 16: ARIMA Raw values regression results (BDT) 22

Figure 17: ARIMA Raw values diagnostics (BDT) 22

Figure 18: ARIMA Returns regression results (BDT) 22

Figure 19: ARIMA Returns diagnostics (BDT) 22

Figure 20: Prophet Raw values regression results (BDT) 23

Figure 21: Prophet Returns regression results (BDT) 23

Figure 22: Bins Classification Confusion Matrix (BDT) 24

Figure 23: Binary Classification Confusion Matrix (BDT) 25

Figure 24: Indices selected for optimization. 2 primary indices in each category followed by their highest correlated international markets. 27

Figure 25: Principal Component Analysis (BDT) 28

Figure 26: Sentiment Scores Compared with Close Returns (BDT) 30

Figure 27: Optimizing hyperparameters (BDT Ridge Classifier) 31

Figure 28: Walk forward test results 32

Figure 29: Ensemble Voting Performance 33

Figure 30: SMA Trading Strategy 34

Figure 31: Market Intraday Trading Strategy 36

Figure 32: Backtesting Results (BDT) 36

Figure 33: Backtesting Results (MNT) 36

**LIST OF TABLES**

Table 1: ARIMA vs Prophet Comparison 23

Table 2: Bins Classification Report (BDT) 24

Table 3: Binary Classification Report (BDT) 25

Table 4: Work Distribution 38

**LIST OF ABBREVIATIONS**

**APAC** Asia Pacific

**OHLCV** Open High Low Close Volume

**ARIMA** Auto-Regressive Integrated Moving Average

**MVP** Minimum Viable Product

**BERT** Bidirectional Encoder Representations from Transformers

**API** Application Program Interface

**NoSQL** No Structured Query Language

**NLP** Natural Language Processing

**TF-IDF** Term Frequency – Inverse Document Frequency

**TDD** Test Driven Development

**MAE** Mean Absolute Error

**MSE** Mean Squared Error

**RMSE** Root Mean Squared Error

**JSON** JavaScript Object Notation

**CSV** Comma Separated Values

**ADF** Augmented Dickey Fuller Test

**PCA** Principal Component Analysis

**SMA** Simple Moving Average

**Often referenced Market Indices:**

**BDT** Bangladeshi Taka Forex

**MNT** Mongolian Togrog Forex

**CSE** Colombo Stock Exchange

**1. INTRODUCTION**

The following chapter entails the overview for the project. To begin with, the background and motivation of the project are presented, followed by a section about the primary focus of the project. Further described in this outline are the key deliverables for this project along with an outline for the remaining content of this report.

**1.1 Background**

The analysis of financial data is an important tool in the business world. The use of computational power as a catalyst in such analyses has created a new space for technology in finance. As emerging technologies are constantly bringing in disruptions to the financial industry, there is a constant need for the development of new and innovative ways of exploiting technology in the domain of finance. Machine learning, a powerful tool is being increasingly deployed to understand financial market movements and to forecast future performances of financial instruments, according to Zhang [1].

In addition, the global economy and the access to information has possibly led to an increased relevance in the interdependence of international economies. The open nature of world markets can prompt discussions as to how and to what extent they affect each other. Such economic possibilities lead to interesting studies in the field of machine learning in order to investigate if such cross-market relationships can be determined using algorithms and if they can be exploited to help forecast future financial data.

Market movement is also affected by other non-financial factors. Ludhiyani et Al say that some of these may include political agendas, environmental aspects and/or public sentiment about these issues [2]. The ability of human psychology to drive investor decisions can be a crucial factor in certain cases where a technological analysis of historic data may not suffice. Capturing these records may also provide valuable insight into some of the non-numeric data that causes market indices to rise or fall.

**1.2 Focus**

The primary focus of this project is to obtain successful forecasts for some of the prominent market indices and use analysis to discover the important factors which influence these markets. To narrow down the focus, the Asia Pacific (APAC) sector has been chosen along with the US markets. Data for Open High Low Close Volume (OHLCV) will be collected for the market indices in these countries over a historical time period daily dating to the existence of these markets. In addition, the forex rates for these countries will also be collected and investigated. The final element will be social media sentiment from Twitter and Reddit using keywords pertaining to the countries such as their capital cities and their presidents. These factors together will be used to predict the market movement by splitting the time series obtained into training and validation data to backtest our algorithms.

**1.3 Deliverables**

Encompassing the deliverables of the project are a project website, a script to obtain forecasts for the next time period and research documentation. Creation of a script to generate forecasts will additionally require scrapers that can obtain both numeric and non-numeric data that are periodically stored in a maintained database. The submission of these above-mentioned deliverables will result in the completion of work on our final year project.

**1.4 Report Outline**

The report has been divided into 10 chapters. Chapter 1 has given a background to the topic chosen and a brief introduction. Chapter 2 provides a brief description of the literature reviewed by the team in order to narrow down to the topic. Chapter 3 explains the methodology of the approach that will be followed to complete this project as per the requirement of the deliverables. Chapter 4 contains details on the work done as part of the initial deliverables of the project. Chapter 5 discusses the following phase involving data collection and pre-processing. Chapter 6 contains the essence of the experimentation with machine learning models while Chapter 7 discusses the optimization approaches used. Chapter 8 encapsulates the backtesting approach. Chapter 9 has ideas for future research while Chapter 10 acts as a conclusion summarizing the information discussed in this report.

**2. LITERATURE REVIEW**

This section entails a brief summary of the literature review conducted by the team for this project. Past research has provided us with information on the different machine learning approaches that have been employed in the financial industry. This acted as a guiding tool to formulate our methodology.

**2.1 Previous Research**

There have been several studies with the aims of predicting future prices of financial instruments such as derivates, securities and cash instruments. Through a rigorous process of literature review, our team moved from research discussing simplistic methods to ones using more complicated statistical models for analysis.

**2.1.1 Naïve Methods**

Many of these studies have primarily focused with one particular entity to attempt a forecast for future periods [3]. For example, the studies performed by Boldt, and Chong and Pu focus on only the sales of Nike and the paying rate index respectively [4][5]. By limiting research to one individual instrument, these research attempts may lack several potentially impactful insights in the data. Historical data taken over a long period of time is sometimes a sufficiently good indicator of the future values in a time series owing to the seasonal and repetitive nature of a financial instrument over a quarter, a year or other financial periods. However, quite often external events often disrupt this periodic nature of the industry thereby creating unexpected movements in the prices of financial instruments. Thus, it is useful to take into consideration external variables to capture such external influences which lead to unprecedented volatilities in data.

Some alternative studies have focused on a range of products albeit having only tested them on a few specific machine learning methods [6][7]. There are two different kinds of studies found, of this type. The first consisted of research that used learning models which could be implemented across a range of financial products. This can be seen in the paper by Wang and Nie where the methodology implemented by them can be reproduced across several different market indices thus enabling greater flexibility in the learning model [8]. This added configurability also permits the audience to review and compare performance across different markets and evaluate the research based on the results obtained. The second and other type of study uses a range of products in a different way. In this case, several different financial products are used as predictive variables (features) to forecast the value of one entity. Research by McCluskey and Liu uses one such approach to forecast the NASDAQ index based on financial data from 12 other technical indicators which comprise of global stock market indices [9]. The usage of external variables can potentially improve the learning process but there is often data in other formats that can also act as an important feature in prediction.

**2.1.1 Sophisticated Methods**

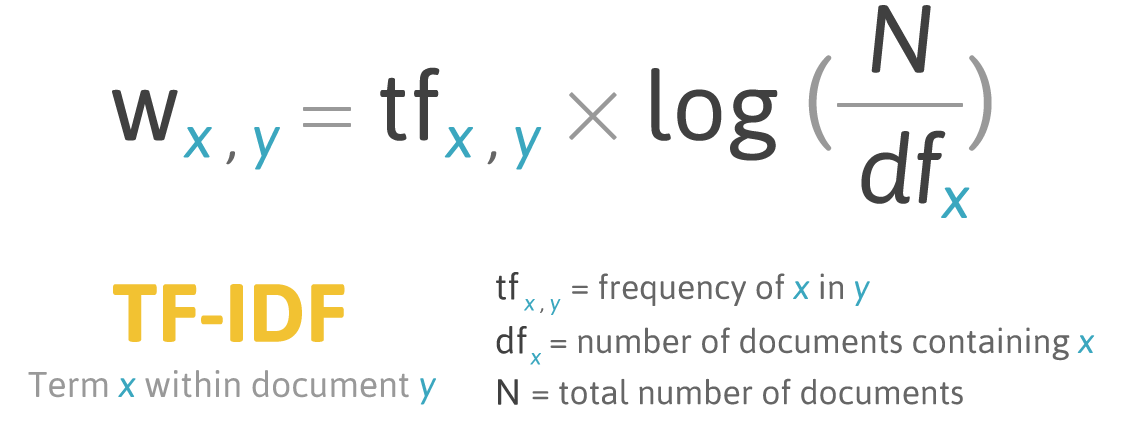
Research has also been conducted on using sentiment obtained from social media websites, but they are confined to the domain of forecasting the direction of the market movement (although with a relatively higher accuracy compared to the methods formerly mentioned) and not the exact values themselves [10][11][12]. These pieces of literature describe using methods of Natural Language Processing (NLP) to extract keywords from the data. As seen in research by Yıldırım et. al. one common method used in the text-processing approaches is using the Bag-of-ngrams approach, which helps in text representation to count the frequency of words in a document [13].

Figure 1: Term Frequency - Inverse Document Frequency Algorithm. Extracted from [14]

A more popular approach considered is that of Term Frequency – Inverse Document Frequency (TF-IDF) which is used to understand the importance of words in a document by calculating their weights with respect to the remainder of the document. As seen in Figure 1 above, the TF-IDF algorithm calculates the weight of each word within a document to get a measure of the overall importance of phrases, sentences and/or paragraphs. This helps convert textual data of various forms into numeric probabilities which are then used in machine learning algorithms. Similar to the approach mentioned before with the range of financial products, there has been some research in the cross-domain aspect which utilized statistical analysis models such as the Auto-Regressive Integrated Moving Average (ARIMA) to perform an analysis for correlation between time series data using automatic regression and moving average. In addition, more sophisticated methods such as the neural networks were also implemented to compare performance with statistical models to check for improvement in performance thereby producing better results. One interesting approach was carried out by Pan et. Al. where a cross-domain study was carried out considering a few major worldwide markets along with their currency pairs over a time series, as illustrated in Figure 2 [15].

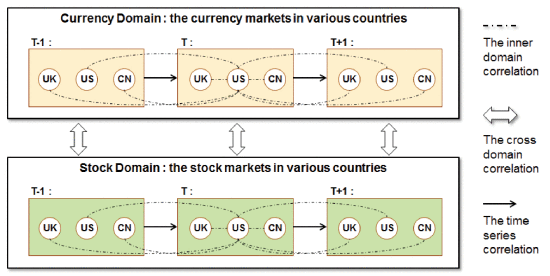


Figure 2: Cross-Domain aspect of Forecasting indicating correlation within a domain, between domains and across a time series. Extracted from [15]

This approach involved using the correlations between countries along with the correlations amongst different markets within the same country as important variables of prediction. In addition, the time series aspect ensured historical values were not ignored and used as a predictor of future values as well. This served as our primary inspiration for carrying out the project. By combining certain aspects of these previous works and adding in some new ideas, our solution will leverage existing research and build on it to forecast financial data. The detailed plan for the approach to be carried out as part of this project is discussed in the following sections.

**3. METHODOLOGY**

A plan of action has been set up to ensure continuous progress for the project through the academic year. The project has been split up into phases for easier development and to pursue a systematic approach of work.

**3.1 Software Development Practices**

The adoption of a software development lifecycle enables us to pursue an organized method of work. The agile system of development, an iterative and incremental model is chosen to allow extended flexibility while working on the project. As per the agile development cycle, the project is split into its requirements as Minimum Viable Products (MVP), each of which are developed iteratively within their own respective phases of work. Bi-weekly sprints act as these individual phases of development where deadlines to complete every 15 days are set within the team. A complete schedule of the proposed plan of work can be found in the Future Plans section along with the status on the individual parts.

The team aims to follow several agile programming practices while moving through the phases of the implementation. A system of simple design is ensured while programming for the project by avoiding complicated programming style. For example, modularity is ensured by separating tasks into different functions. At the same time, redundancy is avoided by not creating impractical functions for menial tasks. The standard of coding is maintained across the members of the team in terms of readability and functionality. For example, the structure of naming entities, indenting code and file formats are kept constant throughout the project. A system of regular refactoring is done to avoid any errors in variable and function naming across the programming environment. The codebase is shared through the team using an effective version control system that ensures there are no leakages in code and that commits can be traced back in a chronological fashion. The process of continuous integration is followed to ensure changes made are added to the pipeline as when they are completed. Test Driven Development (TDD) helps to keep code in check by writing tests to suit our needs which initially fail and then following it up by programs that work to pass those tests. In addition, a pair programming approach is used often in the team to help debug important parts of the code and to discuss efficient ways of writing parts of a program.

**3.2 Project Framework**

The project work is carried out primarily using Python as the development language. Python is chosen by the team for multiple reasons. The presence of large-scale standard libraries enables to use the language in a variety of applications. In addition, the simplicity of coding in Python along with the ability to integrate lambda functions and other aspects of functional programming in the codebase. The extensive usage of Python also enables the use of open source libraries (such as those written by Google and Facebook), some of which are used regularly in this project. Libraries such as **pandas** and **scikit-learn** facilitate working with big data by providing tools to use configurable data structures (like data frames) and provide functionality for machine learning algorithms usable with a wide range of parameters.

In addition, we also aim to utilize several frameworks available on Python for the different requirements of our project. The **Scrapy** framework on Python is used to construct crawlers for websites that enable quick and efficient access to web data in the required format. The alternative to Scrapy, being Selenium, is not considered for this project owing to the ineffective nature of Selenium where it requires the opening of the webpage on a browser window in order to scrape data from it. The use the **Django** framework in Python, permitted us to also deployed a web server to **Heroku** to handle data queries. The **Anaconda** platform provides the functionalities of **Jupyter**, a service useful for experimentation while working on the data with the machine learning models.

For the purpose of the development of the Final Year Project website, HTML, CSS and JavaScript are used to create a static webpage to demonstrate the progress of the project.

**GitHub** is used as the primary version control system to ensure code is maintained in a secure and organised allowing the team members to trace back work at any point of time. The services of **Trello** are also used to create a virtual scrum board for the team to keep a track of progress through the different iterations of the project.

**3.3 Data Collection**

The initial steps required for our project of financial data analysis required data to be collected from the various sources. The process of data collection first required the identification of the different types of data required for the project. The data needed was divided into numeric financial data and non-numeric textual data.

**3.3.1 Numeric Data**

The numeric financial data comprised of 2 different types of data. The former is the values of the market indices for the prominent stock markets in the APAC zone. For this purpose, 18 different countries were selected, and the primary indices were picked from these limiting to 2 indices from each country. In addition, 3 indices were selected the United States to provide a common ground while cross analysing these indices. The values of the indices were obtained daily containing the OHLCV data for each day the individual markets were open and operating in their respective countries.

The latter part of the numeric data incorporated the currency exchange (i.e. Forex) rates amongst these countries. Forex rates were not calculated and obtained for each of the pairs permutable among these 18 countries. Instead, the US Dollar (USD) was kept as a baseline and the Forex rates were obtained for each country’s currency in the form XYZ/USD (where XYZ represented the currency). The currency data is also obtained daily in the OHLC format.

**3.3.1 Textual Data**

The non-numeric data to be obtained consisted of social media posts from some popular social media such as Reddit and Twitter. Keywords relating to the countries selected, their primary capital cities and their heads of state were used to identify relevant tweets for further analysis. A similar methodology is adopted with Reddit to gather posts from certain subreddits. These subreddits were identified by picking the countries and their capital cities.

**3.4 Data Cleaning and Pre-processing**

Similar to the data collection steps, different pre-processing methods must be employed to the numeric and textual data owing to their individual natures. The numeric data can be directly cleaned as it exists on a day to day basis whereas the text data must be processed first before other computations can be performed on it.

**3.4.1 Numeric Data**

The numeric data is cleaned by getting rid of missing values and ensuring that time periods are consistent between the different countries selected, taking into consideration the different time zones. The records containing missing values will be removed if they are in a minority. A better workaround is to replace the missing values with an average or median of the prices for the days immediately preceding and following it to remove any potential discrepancies. Owing to the different countries for each market index picked, the dates for the market being open are to be taken into consideration while cleaning. The common dates across which the different markets are open are taken as they are keeping the US market as the standard, whereas the unique dates for certain countries must be dealt with differently. One potential solution for this is to aggregate the market movement over the extra open days and reflect it in the first common date of both the markets.

**3.4.2 Textual Data**

In order to utilize the obtained text data in machine learning models, they will have to be transformed to some numeric format to be used for the purpose of optimization. One method we aim to employ for this is the use of sentiment analysis on the stored text. Sentiment analysis analyses a piece of text and provides a mathematical value on a scale for the general sentiment of the phrases. Each post from social media is evaluated individually to get a score that is then to be aggregated on a daily basis. If the post is in a different language, the evaluation is done in the language of the post using a multi-lingual encoding program. This aggregation of sentiment will be done by calculating a geometric mean rather than an arithmetic mean to ensure that the compounded effect of sentiment is not ignored. In addition to the post itself, the social media data also consists of values such as number of replies and favourites which will be aggregated in a similar manner.

**3.5 Exploratory Data Analysis and Feature Engineering**

The Exploratory Data Analysis (EDA) and the Feature Engineering step of the process are extremely crucial to the project. This phase holds greater significance to the project than the implementation of the machine learning algorithms. It is only when the data has been understood and transformed to the relevant form can the machine learning algorithms make the substantial predictions for the future time periods with greater accuracy.

**3.5.1 Exploratory Data Analysis**

Exploratory Data Analysis is the part of the project pipeline which investigates the data to identify existing patterns or trends that can ease the process of machine learning. The exploration phase of the project can help us obtain initial analyses of the raw data to draw our own relevant conclusions as to what some of the potential parameters can be in the learning models. Common techniques of exploration include charting out the data using graphs such as histograms, scatterplots and candlesticks.

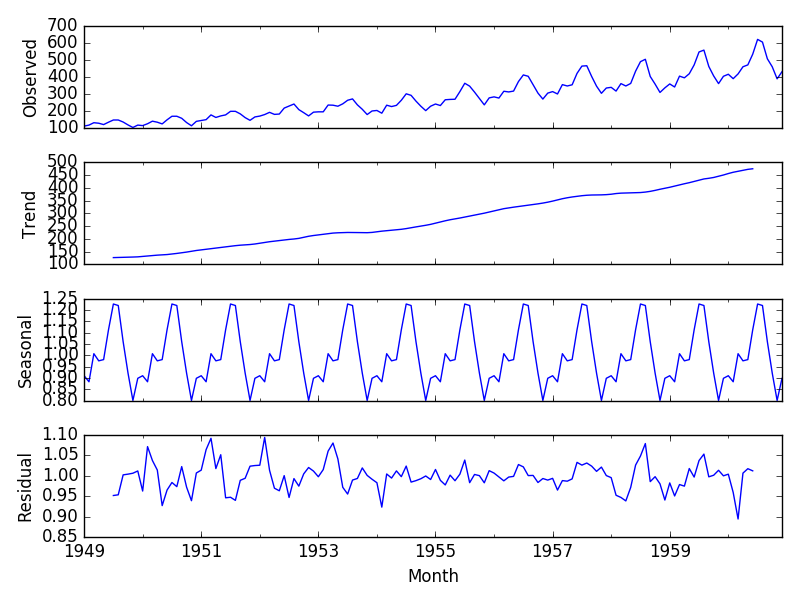


Figure 3: Seasonal Decomposition of data indicating observed data, trend, seasonal and residual components.

As seen in Figure 3 above, applying a method like seasonal decomposition can identify trend and seasonality and allow us to compare it with the actual data to obtain relevant insights. A yearly seasonality can be clearly noticed in the data. This can be exploited in our machine learning models where setting a seasonal component of 12 months can achieve better results than naïve computations.

**3.5.2 Feature Engineering**

Based on the results obtained from the data analysis, we can perform feature engineering to obtain the relevant features and manipulate them. Common methods involve resampling the data to obtain a non-biased sample. However, dealing with several time series, resampling of data points in our dataset cannot be performed for this purpose without an. The data once plotted can give an indication of outliers in the data. Such values of acute skewness can be removed by transforming the data by performing some mathematical functions. Some of the common ways to address this are to use normalization, standardization and even simple mathematical functions such as a logarithm to remove negative values and scale all values to a smaller and easier computable range. Feature engineering will increase the relevancy of the data to be incorporated in the machine learning algorithms to achieve better results.

In addition to the above-mentioned ways of transforming the data, there are also steps in feature engineering based on the machine learning models chosen. For naïve machine learning models which do not implicitly evaluate the time series nature of the data, it is important to provide it with a few values from the past to allow previous records to be used as features too. This will be done by either directly including the raw values of time period **(t-1)** in row **(t)** or by calculating the rates of return over the periods thereby further simplifying the data. For such machine learning methods, inbuilt functions will also help us generate moving averages over a weekly/monthly time period to be used as features.

**3.6 Algorithm Implementation**

Our methodology incorporates the use different algorithms iteratively over our data through the phases of implementation. This serves the purpose of comparing the performance over different algorithms as well as identify the algorithm with the best results.

Figure 4: Iterative Machine Learning Process

As mentioned in Figure 4, the iterative machine learning process starts with the train test split of the data. Next, we pick the best parameters for the models that are experimented with. Following this, the model is trained on the train data and then test it using the test data. The model’s performance is evaluated using the metrics selected.

**3.6.1 Train Test Split**

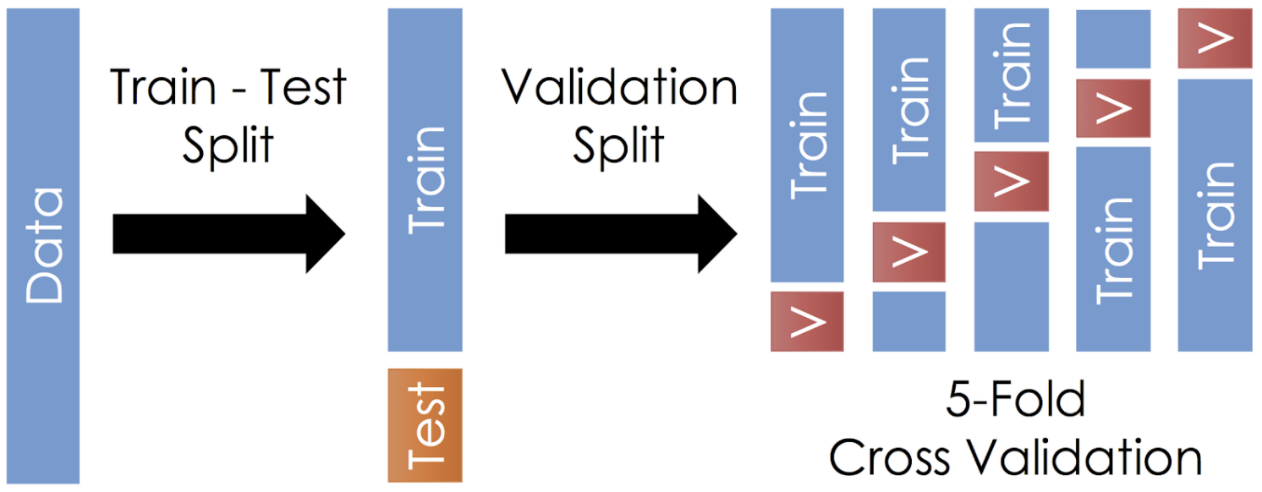
Once the data has been collected and cleaned, the next step is to split it into a training and testing data to pass into the algorithms. For the naïve methods, we can randomly sample the data as the past values have already been taken into consideration for each individual record. However, for methods that take the time series factor into consideration (implicitly such as ARIMA) the train test split if performed by considering the older values as part of the training set and the newer values as part of the testing set. Potential values for the train test split range from anywhere between 50% to 80% which will be maintained throughout the models. To improve the performance of the training model, cross-validation is deployed on the training set as shown in Figure 5 below. By splitting the training data into several validation sets, the model learns during the process of training by penalising itself for its errors and improving its parameters.

Figure 5: Train test split and Cross validation

**3.6.2 Parameter Selection**

Once the data has been split into the necessary train, test and validation sets, the best parameters for each of the individual models are selected. By using the cross-validation approach as mentioned before, the most optimal parameters for each of the models are picked by simultaneously building models with the different parameters and picking the ones with the ideal learning curve. For example, in the Random Forest approach, the number of trees per cluster is decided after running the model training over the cross validated subsets of data and choosing the parameter with the most accurate fitting. For more sophisticated models like neural networks, the parameters such as the number of nodes per layer and activation function will be manually varied and tested to arrive at the conclusion for the best set of parameters.

**3.6.3 Train and test model**

The machine learning algorithms will take the features as inputs and aim to predict the target variables as outputs. The features used in the algorithms will be the time series data of the indices of the other markets, the currency exchange rates between their countries, the social media and news posts. Concerning the target variables, the machine learning models will not only try and forecast raw values of the indices but also consider the accuracy of prediction in terms of only forecasting the direction and attempting to calculate the bins of the price range into which the values fall into. The modelling of algorithms will start with classical regression and classification approaches, along with ensemble methods. The supervised learning methods the team aims to implement here include Ordinary Least Squares, Ridge, Lasso, ElasticNet, KNN, Naïve Bayes, Support Vector Machine (SVM) and RandomForest and AdaBoost. The next phase consists of using time series modelling using statistical models. Models such as ARIMA, SARIMAX, and Facebook Prophet are examples of some of the models as part of this. Finally, we will implement neural networks to aim to forecast market behaviour as has been performed before [16].

**3.6.4 Evaluate Metrics**

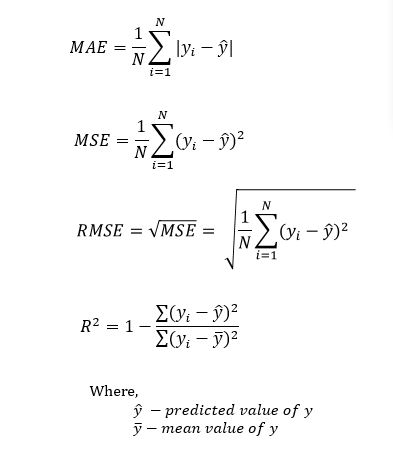
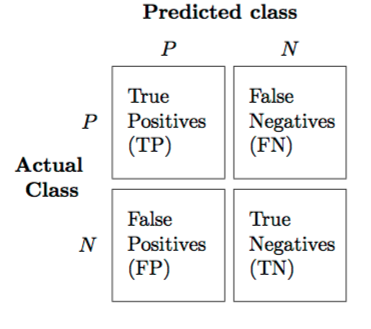
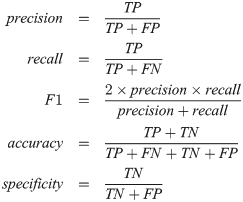
The results across models will be consolidated in an organised way to help clearly identify the best model that is able to learn from the data provided. Owing to the different nature of the target variables (raw values-continuous, bins-discrete), there are different metrics to be considered for each of those. Metrics such as the Mean Absolute Error (MAE), Mean Squared Error (MSE), the Root Mean Squared Error (RMSE) can be maintained across the different types to judge the performance of the models. The regression methods will also have an additional metric of the R-Squared Score (R^2) which is an indicator of the extent to which the feature values are correlated with the target values. The formulae for these metrics can be seen in Figure 6.

Figure 6: Evaluation Metrics - MAE, MSE, RMSE, R^2

The classification methods will have the additional metric of the F1 Score (as in Figure 7) which is a good measure of the accuracy. The F1 score is calculated by first creating a confusion matrix that represents the various values predicted for each category across the actual values. This also calculates the Precision and Recall scores which indicate the ratio of true and false positives with the data. The metrics mentioned are to be compared only across similar target variables to ensure there is no bias in judgement.

Figure 7: Evaluation metrics - F1 Score



**4. PHASE 1 DELIVERABLES**

Continuous work was carried out to meet the deadlines of our proposed schedule. After having crossed the milestone of the first deliverable, the initial arrangement was confirmed, and work is being carried on based on those decisions.

**4.1 Progress**

Due at the end of September, the phase 1 deliverables of the project primarily aided us in deciding the areas of work for the project and how to work about them. The two submissions for phase 1 included a project plan for the project and an initial website for the same. The project plan was a detailed introduction to our ideas for the project and how it was intended to be completed. This included the literature review, the proposed methodology to be carried out and the schedule for the project. The latter of the deliverables, the project website was created using an existing template found on the Creative Commons. This website provided a comprehensive outline to our Final Year Project containing an overview of the project, links to all the documentation, the schedule and proposed approach for the solution. During the implementation of our project, this website was continuously updated having a summary of our approach and the documents prepared by the team. This static webpage created is deployed on the HKU CS server for public access at the URL <https://i.cs.hku.hk/fyp/2019/fyp19020/>

**5. DATA COLLECTION AND PREPROCESSING**

**5.1 Data Gathering**

The requirement for collecting data of two different natures needed two different processes for their procurement.

**5.1.1 Numeric Data**

The two types of numeric data are obtained by using Application Program Interfaces (APIs) provided by open source projects online which contain historical data from websites that offer free real time quotes of financial data. The **InvestPy** package was identified to procure the numeric finance data. This data collection was achieved by installing the relevant packages in Python and utilizing these APIs to scrape data between the required time periods. The flexible configurability of the APIs allowed us to select between different time periods from the inception of the preselected markets until the current date to obtain the records. By storing the countries, their currencies and indices in a text file which can be easily parsed over the Python, the Pandas package was used to iterate over the pairs of currencies stored in the text file and save relevant data frames with the information that was required. These data frames were stored in a locally managed spreadsheet for quick and easy access. In addition, owing to the small size of the data, the need to maintain a server to store it was made redundant as it would lead to overhead costs.

**5.1.2 Textual Data**

A different process had to be employed for the non-numeric data required for this project. For this part of the information retrieval, APIs could not be used and hence a web crawler was constructed to collect the desired data. Tools such as Scrapy and Beautiful Soup enabled the construction a social media crawler for Twitter which will provide tweets between any two historical dates. The working of Scrapy can be seen in Figure 8 below which indicates the process our scraping mechanism goes through to obtain the tweets. A request is sent to the web server from which information is to be extracted. Once a positive response is received, Scrapy extracts the relevant parts of it and saves it in the database of choice.

The query can further be narrowed down to searching for a keyword or limiting it to tweets of a user. Posts from users were scraped from these specific subreddits providing public information about the happenings in these countries. In addition, news articles were obtained using APIs online as well. News articles were collected by searching for the keywords pertaining to the countries and obtaining the articles.

All the textual data collected is stored in JavaScript Object Notation (JSON) files on a server online. As there is a large amount of data from social media and news sources, the decision to store the data on a server online was taken in order to not occupy local storage affecting processing power. This will also provide decentralized access to the data allowing to use external computing power to perform analysis on the data.

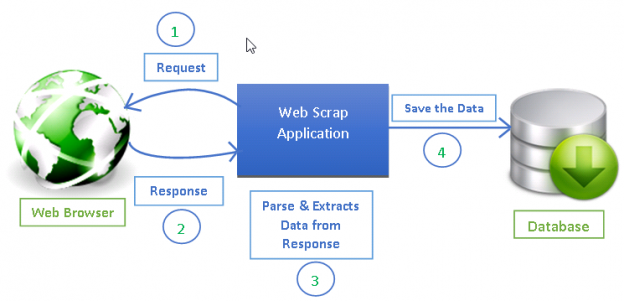


Figure 8: How Scrapy works, from sending a request, getting a response to parsing data and saving it in a database. Extracted from [11]

The Python scripts involving the use of the APIs and the scrapers are set up to run on a periodic basis. This will ensure that data is constantly collected and stored regularly. Moreover, this will quicken the process of financial data forecasting when needed as new data will only have to be scraped from the last checkpoint of the scraping process.

To periodically run these scrapers and for storage online, a server with virtual machine access was obtained from the HKU Computer Science department. Remote access to this server allowed data collection to be efficiently completed by the team. This enabled us to set up scripts that would run on a frequent basis to query data, transform it, update relevant queries and move them to a storage directory.

Several scripts were written to accomplish this goal. A file containing the queries required is maintained. The first of our scripts starts a background process using one of the incomplete queries from the file. Another script helps verify the completion of a query and if so, updates the file with the status of the query. A third script is used to check if there are missing dates in the data collected and helps create duplicate queries for the missing dates. Following this, a script converts the data obtained to the required JSON file removing the unnecessary fields, and creates a Comma Separated Values (CSV) file copy with only the post for use in sentiment analysis. The results of these scripts are stored in the appropriate location along with log files containing notifications of whether the script crashed due to an error or caused any issues.

Automating this part of the process enables a more seamless transition to the server-based prediction model we aim to deliver. In addition, this would also eliminate the redundancy of the team having to manually run scripts to collect the data for the project.

**5.2 Sentiment Analysis**

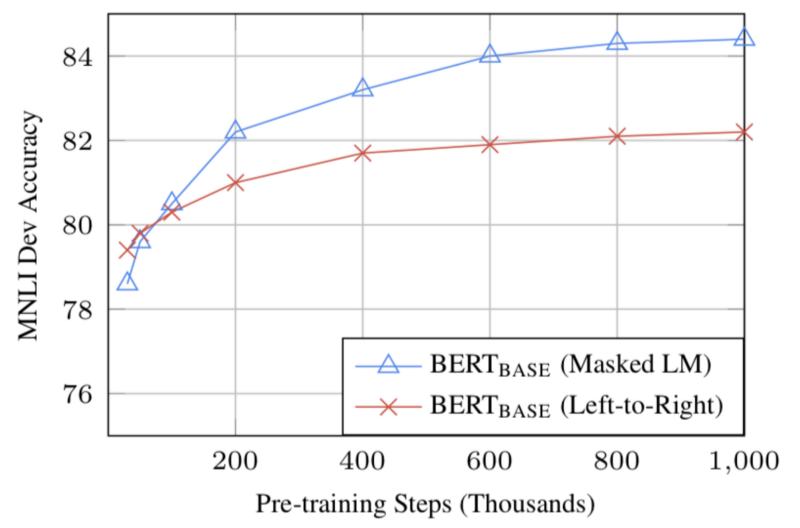
The sentiment model used in this project is built upon a publicly available model titled BERT which stands for a Bidirectional Encoder Representations from Transformers.

Figure 9: BERT Accuracy; increase in accuracy owing to the Masked LM feature of the model

BERT works by using a mechanism to initially encode the words in a sentence. Contrary to other methods which parse a sentence in one specific direction (either left to right or right to left), BERT reconstructs the sentence from both directions thus building contextual relations for each of the words present. As shown in Figure 9 above, by proceeding in a bidirectional format the BERT model is able to perform better than models which read through a sentence in only one direction. The data used in the statistical comparison is of the MNLI data set, which is a corpus of nearly 400,000 sentences spread across several genres.

Next, two strategies are employed in the learning process of our BERT model. The first is one called Masked Learning Method (Masked LM). As part of this, certain words from the sentence are masked out and the model aims to predict the scores of these words while analysing the remaining words. The second is called Next Sentence Prediction (NSP) where using one sentence, the next sentence is predicted. Utilizing these two to minimize the losses caused due to errors, BERT can arrive at the general sentiment for a sentence.

The BERT model is initially trained using a Multilingual Cased model provided by Google which encompasses the training corpus for the project. The results of BERT give a matrix of analysed sentiment. This is further fine-tuned by adding a layer of classification to get results on a binary scale. The fine- tuned version of BERT provides us with two probabilities indicating whether the post is positive or negative. These probabilities are converted to absolute values on a 0/1 scale to aid in our machine learning approach.

**5.3 Data Pre-processing and Feature Engineering**

The data collected spanning different countries across the world had several inconsistencies. The dates on which the time series start for the different markets differed as several of the national stock market indices did not exist from before 2010. In addition, due to each country having a different set of national holidays, the dates for which data was missing also was not uniform across the different markets. For this purpose, interpolation was performed on the dataset. Each time series was interpolated for the missing weekdays using the component of the time as the factor to calculate the magnitude by which the values were to be changed for the missing time periods.

An initial plot of the values generated indicated that the raw values were not suitable for use as the target variables on their own. As discussed in our Section 4.4.1 of our approach, this was because as more than often, indices would oscillate around a certain point and predictions around the point would be considered to have high accuracy despite having predicted in a wrong direction. In order to ensure the target variable would not allow falsely accurate results, the data was to be made stationary.

A screenshot of a cell phone

Description automatically generatedIn order to ensure that our data was stationary, the Augmented Dickey-Fuller (ADF) test was used to inspect the data. The test revealed that the raw values were not stationary. Thus, to make the raw values stationary, the daily returns were calculated for each of the indices. A seasonal decomposition of the returns calculated in Figure 10 below show that the calculated returns in fact do not possess any observable trend or residuals.

Figure 10: Seasonal Decomposition of Hang Seng Index Returns Over 10 Years

To facilitate the classification prediction methods, the returns calculated were also transformed into classes using two methods. The first method was to classify them on a binary scale by converting the values of the returns into 0/1s on whether they were positive or negative. Apart from this, a bins classifier was also used to classify the returns in 5 bins depending on the proportion by which the returns increase or decrease. This was done using a uniform distribution of the returns.

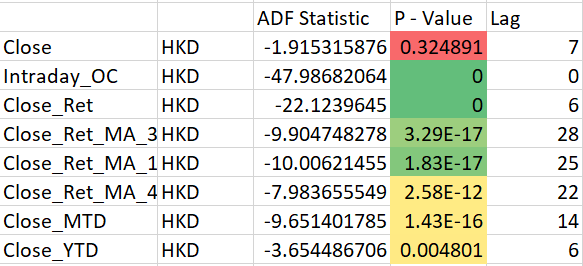
In addition, the use of raw values as feature variables was also not enough for the prediction. This was because each market index was on a different scale and one could not be suitably used as a feature for another in its raw format. In order to ensure comparable features across the different markets, several additional features were to be engineered which would have to be stationary like the target variable. The intraday return between the Opening and Closing price, along with the Intraday return between the High and the Low were calculated. Moving averages over periods of 3, 15 and 45 days were also obtained.

Figure 11: Results for Augmented Dickey-Fuller Test Highlighting the Stationarity of Calculated Daily Returns

The ADF test was once again performed to check the stationarity of these additional generated features as seen in Figure 11. The highly significant results of the test with a P – Value of less than 0.05 indicate that barring the raw values, all other engineered features are stationary and are thus used in our learning approach.

The results obtained from the sentiment analysis were to be aggregated as well. These were aggregated daily by calculating different metrics. After the collection step, the sentiment scores along with the number of likes and replies are collected along with the relevant timestamps in a dictionary. From these metrics such as the arithmetic mean, and the geometric mean are calculated along with those such as the number of posts and the rate of posts in a day. These metrics are lagged to create the relevant features for our time series modelling approach. As sentiment results are also collected over the weekend, the sentiment scores from Saturday and Sunday were aggregated along with Friday to ensure consistency with the time series data.

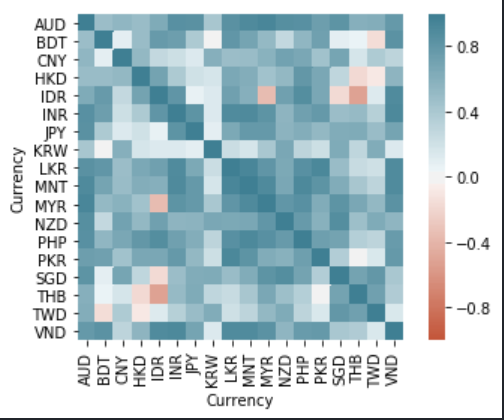
Further analysis of our prepared time series data helped identify some interesting patterns on the cross-market domain which could be exploited. Strong correlations were seen between several national currencies indicating that the currencies in pairs with the highest correlations could be used as additional features while predicting the values for the others. One such example can be seen in Figure 12 where BDT can be seen to be highly correlated with VND indicating the possibility of VND being a good predictor for BDT.

Figure 12: Correlation matrix for Currencies highlighting Correlations discovered over the 10-year period

**5.4 Challenges Faced**

The task of building a financial data forecaster comes with several challenges throughout the process. Our team faced some issues while working with the project and creating the necessary codebase required for its implementation. These challenges and their mitigations are discussed in this section.

### **5.4.1 Data Collection**

Financial data forecasting is usually performed using tick-by-tick data collected from financial data firms which collect such information. We were unable to find APIs online providing information at such high frequency available online for projects. As an alternative, the team resorted to using daily data for the project. The lack of quantity of data owing to reduced frequency was compensated by collecting data over a large time period of nearly a decade.

Social media like Twitter and Reddit provided APIs to access tweets, although they were limited to obtaining the most recent tweets with a limitation to the number possible to obtain. To obtain posts from a historical time period, our own scraper was constructed using Scrapy as mentioned before.

The files created from the crawlers are large (~1 GB) each and were unable to parse through normal Python scripts. To move past this issue, the team resorted to different languages for writing the script. First a Shell script was attempted as a lower level language would bypass memory restrictions. Although it showed better performance than Python, some of the issues were still prevalent. Some research about programming languages pointed us to Perl, which differs from other languages in the way that it uses a buffer to process files in small parts contrary to reading the whole file at once, as Python does. Recreating the script in Perl helped mitigate this challenge and run the scripts successfully.

To maintain a fast querying system for data, the team looked to deploy the database to a server with a computing engine that would enable us to work with the data in a scalable and organised way. However, owing to financial restrictions on the project, we resorted to utilizing services on Heroku, which provides a free minimal server for application development that will be suitable for our intermediary phases. The provision of additional storage space on the server by the Department of CS at HKU led to the shifting of our data from the free online cloud platform to the university’s server.

### **5.4.2 Data Pre-processing**

One of the primary data pre-processing challenge involved training and evaluating the sentiment analysis framework BERT. Owing to the large size of the training corpus, an average server was unfavourable to the team. Thus, the services of Google Colab were utilized to train the BERT model using the hosted GPU provided. This reduced the training time for the model. However, since there was a large amount of social media data to be evaluated against the trained model, Google Colab was not picked by the team for the approach as Google Colab disconnects hosted runtimes after extended periods of inactivity. Instead, the sentiment analysis was performed on the HKU CS server provided to the team. This led to a severe decrease in speed and extended the time taken for this phase by a significant margin.

**6. EXPERIMENTATION**

Once all the data was prepared, the next phase involved experimenting with the available data with our modelling methods. As proposed in the methodology, 3 different types of predictor variables were to be used in the modelling approach. Discussed in this section is the experimentation phase of our project which broadly encompassed 3 different types of methods.

**6.1 Experimentation Approach**

For most of the approaches carried out, scripts were written which would iterate over the sets of the currency and market pairs for the provided regression/classification methods, and for scaling applied to the variables before fitting. This enabled us to examine results over a range of different parameters in our approach (machine learning model, scaler, and dataset) and compare results to see if one would be more favourable than the others.

Approaches were also carried by considering the inclusion and exclusion of methods such as Principal Components Analysis to reduce the number of features and to transform the data to a lower dimensional space. In addition, the results of the sentiment analysis were also verified with an inclusion/exclusion method to understand their effects in the forecasting process.

As part of the machine learning models, all models from Scikit Learn were experimented with in the regression and classification space, respectively. While performing these approaches, experiments were carried out using endogenous lagged variables of the currency/market pair first. Methods which provided adequate performance were then probed further with the cross-domain approach.

A train-test split of 80:20 was followed consistently throughout the approach without shuffling the data as the primary technique. The experiments were also carried out with and without generating polynomial features.

To maintain consistency, all graphs represented in this section will be involving the Bangladesh Taka (BDT) currency to get a better understanding of the difference in performance between models.

**6.1.1 Regression Approach**

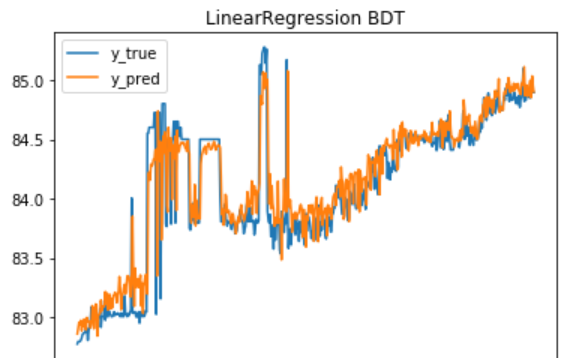
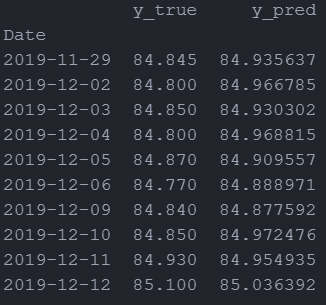
Aiming to predict the raw values directly, the preliminary approach was to apply the regression models directly to the time series using the lagged features generated in the previous step. The results for most of the currencies for several of the regression models were similar which led to incredibly high R-Squared scores. To avoid redundancy, results from one model are shown here in Figure 13 below.

Figure 13: Raw values regression results (BDT)

Figure 14: Regression Overfitting (BDT)

However, closer examination from Figure 14 showed us that the models were unfavourable to raw values as they moved according to the returns in the previous time period. A quick glance at the data frame shows that the changes from one period to the next for the true values is reflected in the following time period in the predicted values. This led to the shift of using the returns as target variables for the machine learning models.

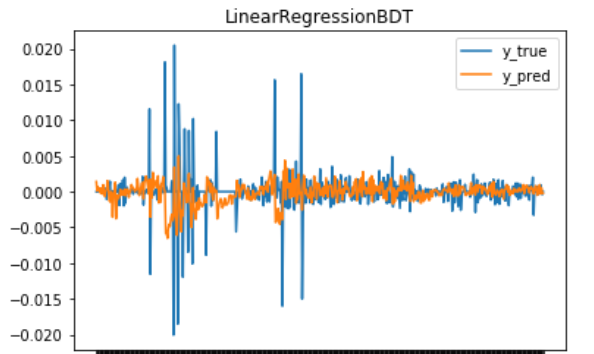
Using the returns as the values to predict, several different regression methods were tried across the currency and market pairs available to us. The accuracy scores obtained were extremely small and almost negligible. Our team noticed that such regression methods were not providing us with good performance as some of the larger trends were not anticipated and picked up by the model seen in Figure 15. Using returns did not favour ****the models and resulted in some better albeit very low R-Squared scores. The Mean Squared Error values decreased as the approach was shifted to using returns, but the team attributed that to the decrease in the scale of the values.

Figure 15: Returns regression results (BDT)

As several of the machine learning models did not generate considerable results, the next approach was to move to classical time series models rather than the machine learning algorithms. As part of this, two models were experimented with, the ARIMA model and Prophet, developed by Facebook.

To experiment with the ARIMA model, an existing framework called AutoArima was used. Using this framework, we were able to perform a stepwise selection of the best parameters along with the results from the ADF test that was performed. To ensure a wider scope was experimented on, the AutoArima was tested with varying configurations of parameters that included seasonality and exogenous variables. In this way, AutoArima was able to experiment ARIMA and SARIMAX models.

First, ARIMA was tested with the raw values itself as ARIMA accounts for the trend in a time series. ARIMA seemed to perform better than the machine learning methods by predicting the values in the general direction of the market as shown in Figure 17. However, it was unable to factor in the magnitude of changes despite accounting for the seasonality.

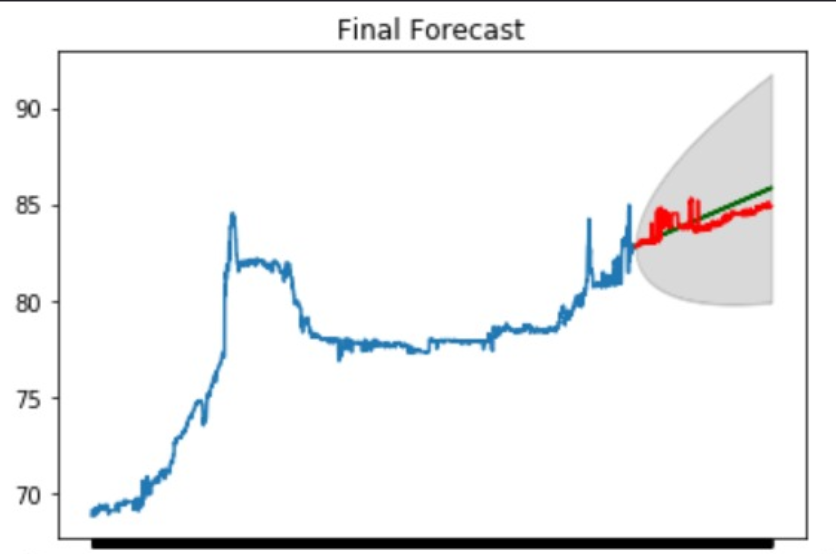
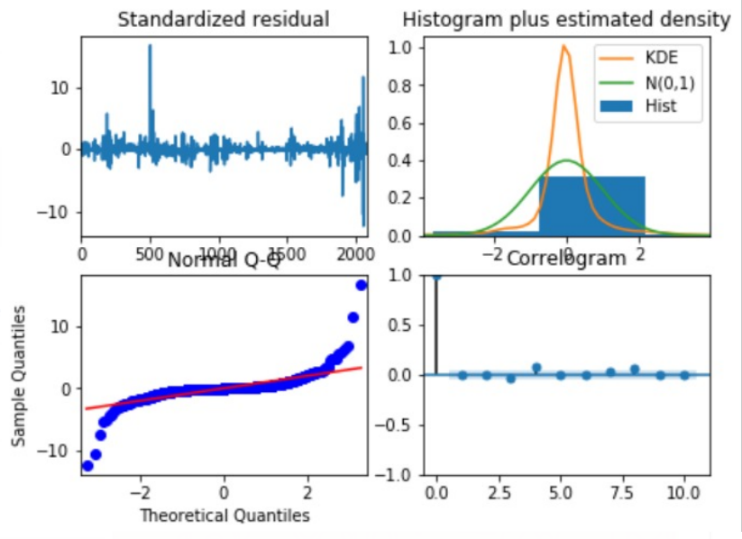
****

Figure 16: ARIMA Raw values regression results (BDT)

Red – True. Green - Predicted

Figure 17: ARIMA Raw values Diagnostics (BDT)

Taking a deeper analysis into the diagnostics of the fit model, interesting observations were made from the residual plots in Figure 17 above. The KDE curve indicate that the data fit is not optimal for the model as it does not fit the normal distribution. In addition, the Q-Q plot curves off in the extremities pointing out few extreme values in our data, which can be attributed to using raw values as there is a visible trend in the data.

Like the machine learning methods, attempts were made to combat this issue by utilizing the returns instead of the raw values as the time series. The ARIMA model was run on the returns in a similar fashion as the machine learning models. Seeing relatively better performance, the model was augmented with feature variables which were the lagged features generated. The lagged features were then used additionally while fitting as well as predicting the ARIMA model.

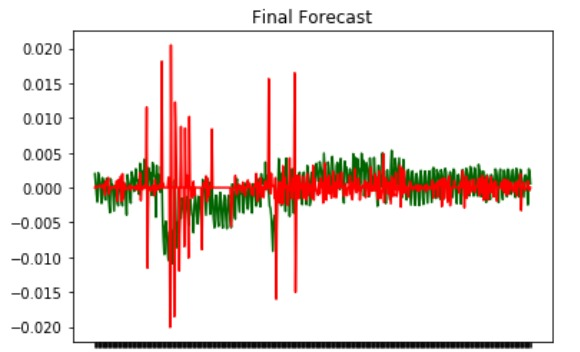
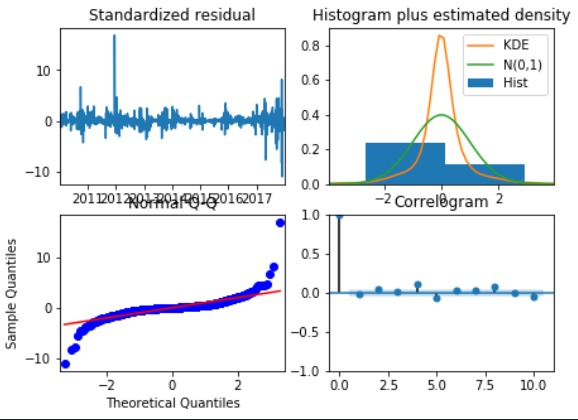
Figure 19 shows us the predicted values for the evaluation period using the returns. Although the results are inaccurate albeit performing better than for the raw values, ARIMA was more successful in understanding the volatility of the returns and provided predictions that varied to a larger extent when the returns were fluctuating. In addition, the subplots in figure 18 show us that the returns are more favourable towards a normal distribution and thus have higher predictability with the ARIMA model.

Figure 18: ARIMA Returns Regression (BDT)

Red – True, Green - Predicted

Figure 19: ARIMA Returns Diagnostics (BDT)

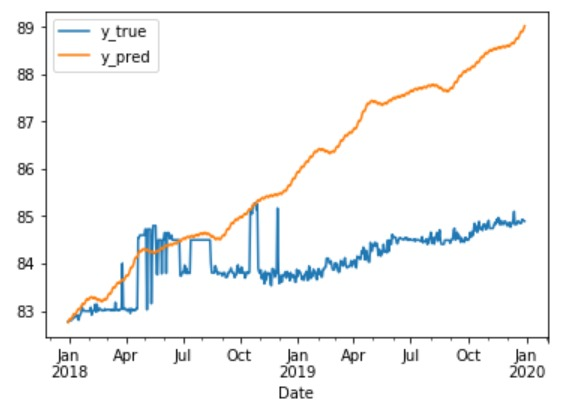
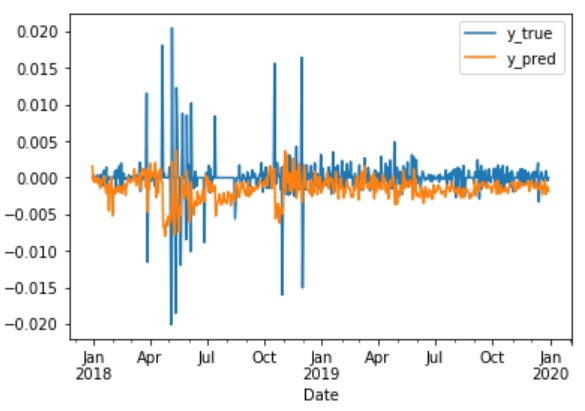
Figure 19 shows us the predicted values for the evaluation period using the returns. Although the results are inaccurate albeit performing better than for the raw values, ARIMA was more successful in understanding the volatility of the returns and provided predictions that varied to a larger extent when the returns were fluctuating. In addition, the subplots in figure 18 show us that the returns are more favourable towards a normal distribution and thus have higher predictability with the ARIMA model.

Figure 20: Prophet returns regression results (BDT)

Figure 21: Prophet raw values regression results (BDT)

Similar experiments were carried out with Facebook’s Prophet forecasting procedure to test if it provided better results. Being built using a combination of models, Prophet was likely to have different results when compared to ARIMA on the same data. Looking at the results from Prophet in Figure 20 above, we can see that Prophet shows a more comprehensive approach to understanding the seasonality of the data although the general trend is exaggerated. While forecasting the returns Prophet provides a good measure of the seasonality of the data as in Figure 21.

The results of the ARIMA and Prophet model are compared using the Mean squared error across the different types of variables used (i.e. raw values against the returns) and whether exogenous variables were used in the fitting process.

|  |  |  |  |
| --- | --- | --- | --- |
| MSE Values | Exogenous features | ARIMA | Prophet |
| Raw values | True | 0.55 | 2.59 |
| False | 0.55 | 5.40 |
| Returns | True | 1.50e-05 | 1.05e-05 |
| False | 1.50e-05 | 8.37e-06 |

Table 1: ARIMA vs Prophet Comparison

Table 1 above indicates how ARIMA has better results than Prophet while using the raw values. Although the results are unsatisfactory for both, ARIMA works better in this area as the moving average is used to predict the future trend. Prophet on the other hand calculates a trend and applies the trend to the evaluation data set which accounts for the inflated trend while forecasting. However, Prophet performs better than ARIMA when the returns are used as the variables. Prophet appears to have a better grasp of the seasonality owing to which it can forecast the market prices better.

Thus, both ARIMA and Prophet have their own advantages and disadvantages when it comes to forecasting market prices. However, none of the regression methods provide good results for forecasting. The R Squared score for none of the models attempted was higher than 10% thus this approach was not carried forward with in our following steps.

**6.1.2 Classification Approach - Bins**

As the regression approaches did not yield any significant result, the next steps were to experiment with the classifying the market values. As most of the markets face long periods of continuously increasing indices, raw values were not used for the classification approach. Instead the returns of the forex and market indices were used to classify.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| -1 | 0.40 | 0.32 | 0.35 | 136 |
| 1 | 0.25 | 0.06 | 0.09 | 53 |
| 2 | 0.53 | 0.34 | 0.42 | 203 |
| 3 | 0.33 | 0.68 | 0.44 | 130 |
| Accuracy |  |  | 0.39 | 522 |
| Macro Avg | 0.38 | 0.35 | 0.33 | 522 |
| Weight Avg | 0.42 | 0.39 | 0.37 | 522 |

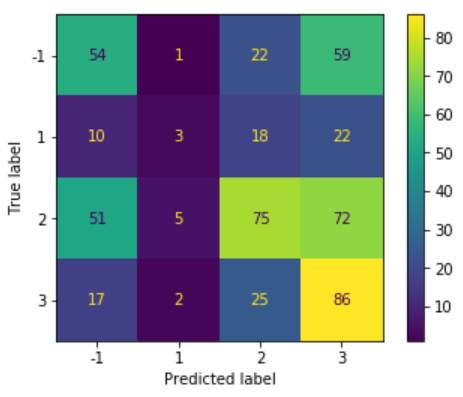
Experiments were performed with different numbers of bins for the market data. Too few bins did not want to be picked as it would reflect a Binarizer. On the contrary, too many bins would be unsuitable as several of our values for the returns were <<1 and would thus create almost indifferentiable splits. As a result, 5 splits were decided as the optimal number after testing. Thus, the data would be split into 5 or lesser splits if the data wasn’t spread across a wide enough range. In addition, the first attempt was discretizing into bins was based on a uniform split looking at the values. However, since many of the values were clustered around the same space and there were only a few outliers, this led to biased discrete bins. Thus, the quantile method was picked as it processed the bins according to the quantiles of the data providing us with an even split. For this approach, only the target variables were transformed to the discrete bins and the features were either scaled or kept as they were.

Figure 22: Bins Classification Confusion Matrix (BDT)

Table 2: Bins Classification Report (BDT)

The first method used as part of this approach was to iterate over the different Scikit-Learn classification algorithms. Several of the algorithms were tried and one of the better results obtained consistently across the different currency pairs was using Random Forest Classification.

As seen in Figure 22 above, the confusion matrix provides us with better results than the regression methods. It is observable that for the classes towards the more positive extent, more correct predictions have been made by the classifier. As a note to point out, the difference in the classes being positive and negative does not pertain to the returns being positive or negative themselves. This was done to avoid the usage of 0 as a class and it does not affect the performance of the classification models themselves as the multiclass classification models do not consider the classes in a numeric sense but as individual distinguishable classes.

From table 2, it is evident that the low values of precision and recall can largely be attributed to the misclassification of the class 1 in our test data. It was interesting to note how one single class was primarily misclassified. Further investigation into this indicated the margin for this class being relatively narrower compared to the other classes explaining the low rate of recall for it.

Classification into bins appeared to be a better approach to our method while making a smaller trade off for the scale to which we predict (Bins classification helps predict a range into which the return for the following day will occur in comparison to vanilla regression which aims to target the specific value).

As none of the variants of the bins classification gave us further improved results to the ones shown above, the next method of classifying on a binary scale was pursued.

**6.1.3 Classification Approach – Binary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| -1 | 0.70 | 0.63 | 0.66 | 316 |
| 1 | 0.51 | 0.59 | 0.55 | 206 |
| Accuracy |  |  | 0.61 | 522 |
| Macro Avg | 0.61 | 0.61 | 0.61 | 522 |
| Weight Avg | 0.63 | 0.61 | 0.62 | 522 |

Table 3: Binary Classification Report (BDT)

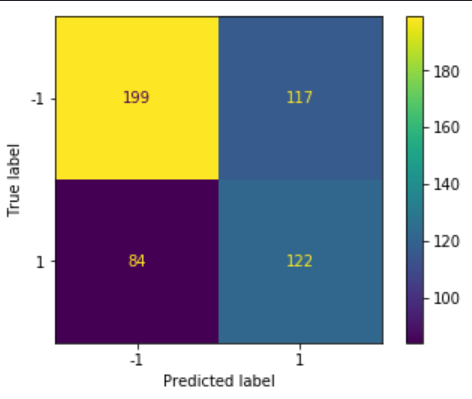
****The third and final approach as part of the experimentation phase was to broaden the scope of classification into one that involved binary classification. For this approach, the returns were used as the target variables and were classified into a positive (1) or a negative (-1) scale. For the purpose of maintaining two distinct classes, the values of 0 returns were converted to the negative class. Iterative experiments were performed as before across the different variables involved in the machine learning approach. There were a few models that provided better performance amongst others.

Figure 23: Binary Classification Confusion Matrix (BDT)

The Confusion matrix in Figure 23 above shows the results for running binary classification on BDT. The confusion matrix is biased towards the negative class as several of the values during the testing period were negative. However, the classification report in Table 3values point out that results are better than average results for precision and recall for both the classes.

These results led the team to see that the binary classification approach thereby lead to a relatively better performance despite the trade-off for the scale of prediction when compared to the bins classification approach. Since a higher accuracy is seen overall with binary classification, the team continued to use this approach in the further steps of optimization to aim for better performance. The other methods were not considered further as part of the project.

**6.2 Challenges Faced**

Several hurdles were faced by the team while experimenting with the different available modelling approaches. To begin with, the initial approaches of running regression models did not yield any comparable results after having tested over many different transformations, modelling approaches and features sets. In addition, the time series models did not provide any significant improvements in performance either. This led to us having to resort to a change in direction of our approach from one of regression to that of classifying the variables.

The experimentation having to iterate over many variables to consider while modelling could not be run on the team’s local machines. Moreover, the statistical based time series models created several bottlenecks in our project. Being computation heavy, these would use up a considerable amount of the processing power while running which would not allow other models to run simultaneously. Thus, all the models to be run as part of this phase were set up in automated scripts and run as daemon processes on the server.

**6.3 Experimentation Results**

The experimentation phase led to several significant results were to be followed up on the continuing phases of the project. These results included identification of the better approaches to use in further progress of the project. The decisions taken from the experiments are consolidated below.

To begin with, the binary classification method was the method chosen to be used for the forecasting of financial data. As seen previously in this section, the binary classification method provided the best results compared to all the different approaches and was selected to be suitable for the continuation of the project. Thus, the returns of the raw values were all transformed on a binary scale for prediction.

As time series modelling approaches did not have significant results, our team resorted to only aiming to forecast for the following day given data up till the current data. Although this narrowed the approach of how far into the future forecasting could be applied, it led to more significant results in the process of prediction.

The experiments also consisted of utilizing cross-validation on the training samples. Contrary to normal cross-validation where random splits are made, this was based on a time series split method which would incrementally add validation sets to improve the accuracy of the model. However, this did not seem to have significant effects on the test data set and was thus not considered to be an important factor in forecasting.

In order to better facilitate the next steps, the target currencies and markets were shortlisted to focus the team’s effort on a few better performing indices. Thus, from the forex and market indices of the 17 countries, two forex indices and two market indices were selected which showed better performance. It was interesting to note that the countries picked were mostly developing countries as they were more influenced by some of the more international economies they were dependent on.

As seen in the Figure 24 above, the middle column shows the primary indices, which were two forex indices and the two market indices selected after the experimentation phase. The column on the right reflect the secondary indices, which are two additional indices (one forex and one market index) chosen for each index which were the highest correlated with it. Although, several of the markets were correlated and moved in the same direction for a predominantly larger duration of the time, these were the indices which had the strongest correlation values across several of the features when compared with the original index. The correlation values were calculated by comparing the lagged values of the secondary indices with the actual values of the primary indices. In this way, the lagged values of the secondary indices will act as additional features along with the lagged values of the primary index too. These would serve to be part of our cross-domain approach in the Asia Pacific region.

Figure 24: Indices selected for optimization. 2 primary indices in each category followed by their highest correlated international markets.

Regarding the feature variables, it was decided that the returns would be used as features rather than any of the raw values. The inclusion of the raw values as features did not provide any improvement to our results when returns were considered to be the target variables. Our experiments revealed no significant changes with or without them. This was as per the team’s hypothesis as the returns were already doing justice by providing the rate of change of the values and doing justice to the classification problem. Moreover, the raw values often showing trends were not suitable as the returns did not show trends and were theorized to be disruptive to the feature space.

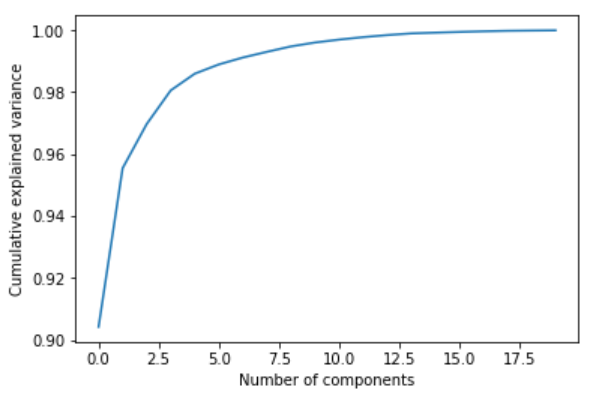
Several different feature scaling methods were implemented to test whether they would act as a more representative version of the features. The transformations experimented with were primarily the MinMaxScaler, the StandardScaler as well as the log1p (log(1+x)) function as a transformer. In addition to not showing any considerable improvements with the regression approaches, none of them aided in the classification approach either. As the feature values were already contained within a margin of (-1,1) and often oscillated around 0, none of the scalers provided any additional information that would be useful to the learning models. As a result, none of the feature scaling methods were chosen to be applied for the forecasting process.

Figure 25: Principal Component Analysis (BDT)

The addition of the cross-domain markets led to a large feature space with nearly 10 lagged variables from each of the indices, including the index being predicted. To reduce the dimensionality, the Principal Component Analysis (PCA) method was used with the dataset. Figure 25 shows the PCA explained variance plot for the feature space containing lagged values of BDT as well as for the secondary indices. Similar converging curves for variance were notices were most of the indices. As a result, a value of 6 was picked for PCA to reduce variance in the model. However, it was noticed that experiments carried out after applying PCA to the dataset only led to a marginal increase in the accuracy. Yet it was favoured for the purpose of the forecasting approach.

The classification models to be used for the optimization purpose were also shortlisted after the experimentation phase. After having iterated through several different models, two of the models that were identified to have provided better performance for the results were the Ridge Classifier and the Support Vector Classifier models. As per our hypothesis, the Ridge Classifier performs well due to the aspect of regularization. The regularization aided by adding constraints to the co-efficients thereby preventing the model from overfitting. In addition, Ridge helps get rid of any possible issues of multi-collinearity, which may arise as our features are susceptible owing to several of them being lagged and calculated from the raw values. For the Support Vector Machine Classifier, we believe the higher accuracy can be accounted to the complexity of the SVM which utilizes its kernels to solve intricate problems. SVM too avoids the risk of over-fitting owing to the ability to maximize its margin and provide more generalisation.

The experiments were also carried out with the scores calculated from the process of sentiment analysis. These did not provide any new additional insight into the data. To ensure that all domains of experimentation were explored using the sentiment scores, models were built with scaled values of the sentiment scores as features too, in efforts to get more insight into the data. However, none of the scaling provided any additional effort

The investigation of the effects of using the results of sentiment analysis was also meticulously performed. No significant difference was noted on comparing accuracy scores with and without sentiment analysis results, keeping the other variables constant. Further investigating was performed to understand the reasons behind the non-favourable behaviour of the aggregated sentiment scores. The sentiment scores were not correlated with the market returns across the indices considered. Although it was interesting to note that most of the sentiment metrics (the list including Average, Variance, Count and a few others) were negatively correlated with the returns, none of the values were considerably significant to appear to have a significant effect on the machine learning process.

In addition, the sentiment scores did not appear to have any additional abilities of forecasting the sudden peaks and falls in the data. Moreover, the sentiment scores do not appear to follow any scores were plotted with the target variables (Close returns), there was no visible signs of the sentiment having a significant trend either. The sentiment scores appeared to vary uniformly after the initial period where it displayed extremely high variance and continue to oscillate around the median as seen in Figure 26 below.

From the figure in the following page, it was also noticed that that the sudden rises in the values of the returns of the index were often followed by sudden rises in average sentiment scores. This led the team to believe that social media sentiment did not provide any new information into the forecasting process.

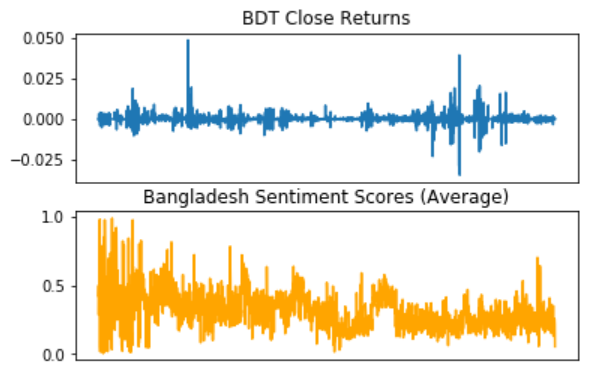
Our hypothesis for the same involved the fact that sentiment generated by the public regarding any important economic/financial decisions that would have led to the rapid changes in the index prices was more than often already reflected in the market before the news reaching the people as a result of the Efficient Market Hypothesis. Thus, our experiments did not lead to any new information than the general perception of sentiment analysis as present in the community.

Figure 26: Sentiment Scores Compared with Close Returns (BDT)

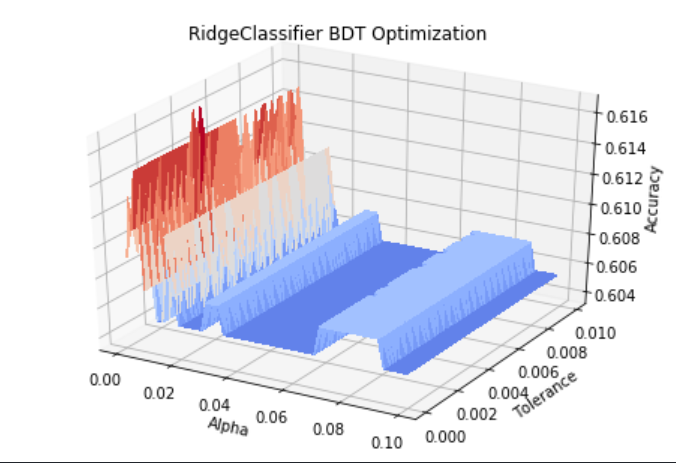
The observations from the experimentation phase were then carried on to the optimization phase aiming to achieve better results.

**6. OPTIMIZATION**

Following the experimentation phase, the team worked on optimizing the learning approaches for better results. Having fixed the indices and the pre-processing steps to be performed, the hyperparameters of the models selected were investigated and the evaluation method improved to consolidate the machine learning methodologies of this project.

**6.1 Hyperparameter Selection**

For the two models selected, a range of values for the hyperparameters were tested. The hyperparameters primarily tested were those for the tolerance (which represents the precision of the solution) and the regularization parameters for both models. This step enabled us to select the optimal hyperparameters that would be proceeded with for the respective market pairs in the project.

Having run the selected models for the selected pairs of market indices, the appropriate hyperparameters and fixed. Considering the example of BDT, as the Ridge Classifier was varied with different values for tolerance and alpha (a parameter indicating regularization strength and is the inverse of C for the equation) and extremely small values for both were found to be optimal. As shown in Figure 27 below, the cascading fall in the accuracy as values increased indicated the highest points around the lowest values of tolerance and alpha.

Thus, the most suitable hyperparameters were identified for each of the models for this step of the project. In pursuit of improving performance, the optimization segment of our approach drove us to work with some additional methods.

Figure 27: Optimizing hyperparameters (BDT Ridge Classifier)

**6.2 Walk Forward**

The current methodology only adapted a simple train test split using the testing data set as the evaluation set over which the accuracy was calculated. Although having a significant amount of data as the training set, the team decided to optimize this approach by utilizing a walk-forward method of validation to obtain better results.

The walk forward approach selected for several reasons. The first, was in order to provide a wider range of predictions for evaluation which had potential for better results. In addition, the nature of the time series made it more suitable to use this approach in comparison to the simple split, as the general perception of time series forecasting indicates that older values do not hold much significance and go through exponential decay in importance as the time series walks forward. Moreover, the results from our ADF test indicated a small number of lags significant for prediction thereby making it redundant to utilize values from the start of the time period.

Several research papers read for the project also carried similar approaches for the prediction of financial data considered as a time series. As per Cao and Tay, the walk forward routine is more suitable than the normal train-test split when working with adaptive parameters [18]. Working with classification models, past research has shown enhanced results using walk forward approaches with financial time series data.

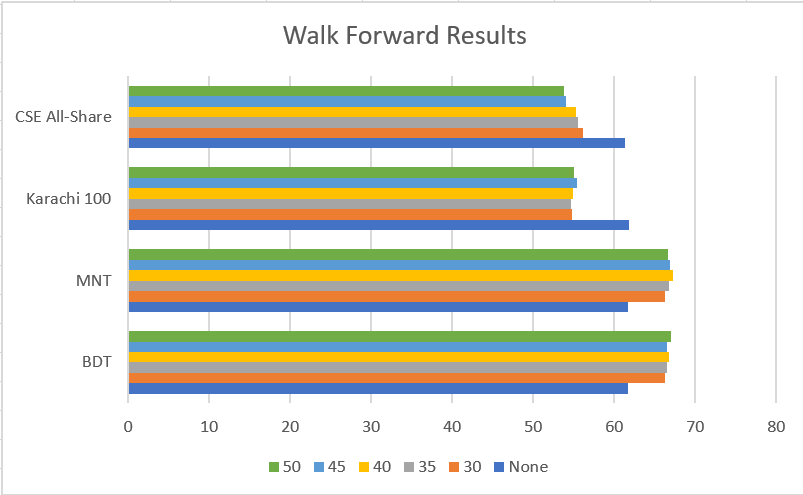
The walk forward validation approach was implemented testing for windows of different sizes. Using a rolling window approach, the data was tested by retraining the model for every window and using that fit to evaluate the prediction for the following time period. Figure 28 below demonstrates the performance of the walk forward validation approach for different window sizes represented by the numbers where None represents the classic train test split approach. As seen below, the walk forward approach was seen to consistently perform better than the normal approach for the forex pairs across a varied set of window sizes.

Figure 28: Walk Forward Test Results

Across all the forex pairs shortlisted there was a considerable improvement in performance seen of around 4%. However, a similar decrease was noted for the index pairs thus proving it was not more optimal than the walk forward approach. Thus, the walk-forward approach was taken as the optimal method of forecasting the forex prices whereas the normal approach was considered optimal for the market index prices. The lower accuracy for the market indices attributed to the fact that the market indices required a longer training period to provide improved results compared to the forex prices.

**6.2 Ensemble Voting**

The final part of the optimization involved applying an ensemble voting approach. From the previous phases, the most optimal models were selected for each of the 4 indices chosen where the walk forward models were the more appropriate ones for the forex indices and the normal train-test split based models were chosen for the market indices.

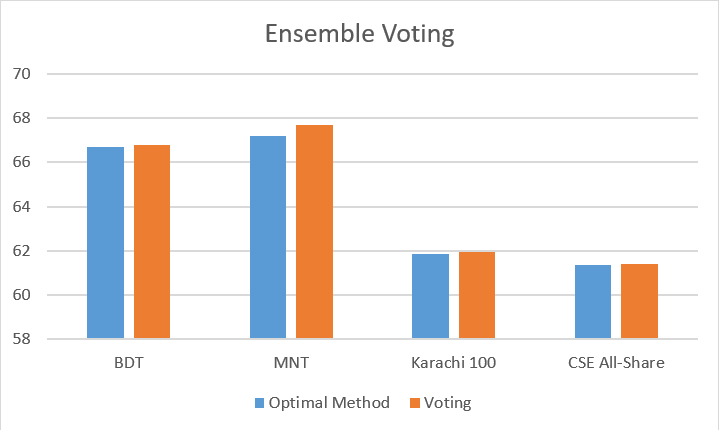
The voting approach was a simple layer over the current machine learning models where the value predicted would be the mode of the predictions from the various models. The way chosen by us to implement this was to essentially take the sum over the predictions from the different models and appropriate that as the final prediction. As the predictions were either a +1/-1 this does not lead to any biases in the voting calculation. In order to maintain the predictions of the model with the highest accuracy as the most important, its results were amplified in comparison to those of the other learning models. This enabled the team to create a form of weighted voting where the model with the highest accuracy would be considered to have a higher weight compared to the others.

Figure 29: Ensemble Voting Performance

Figure 29 above consolidates the results of the Ensemble Voting Approach. As seen in the graph, this method improved the prediction performance for each of the market indices picked, albeit by a marginal amount. Picking the mode of the predictions has certainly provided more accurate results by leveraging the power of the different models combined. The voting approach was thus seen as the final optimized performance for each of the currencies picked and the end of the machine learning phase.

**7. BACKTESTING**

The final step of the project involved backtesting the optimized learning techniques against the market to verify if this would be successful given the history of the market. The backtesting approaches utilize the results from the walk forward validation as inputs to check if positive returns can be obtained using a favourable trading strategy involving our predictions.

The backtesting was performed with the help of the **backtesting** module in Python. This module enabled the team to perform backtesting in an efficient manner by providing configurability for the trading strategy to be used with the conditions and the quantities for the trade. Scripts written using this module then performed the backtesting provided with the results from the optimized learning algorithms to generate equity returns for the provided time series data.

**7.1 Simple Moving Average Strategy**

An appropriate trading strategy was to be decided that would well utilize the outputs from our forecasting model and apply them to the market. To decide a good strategy, the team turned to past reports to understand some of the common methods used for backtesting financial time series data. Amongst others, a research paper by Kostiainen K. showed the use of a Simple Moving Average (SMA) strategy for use as backtesting [19]. In addition, Popov and Madlener demonstrated how the SMA strategy outperforms several other trading strategies considered for financial data [20].

Thus, the first method considered as part of this endeavour was the SMA strategy. As per this, two sets of SMAs are calculated across the range of financial data. The first is a slow-moving average, calculated over a larger range than the second, which is a fast-moving average. The moving averages are calculated simply by averaging values across a rolling window. According to this strategy, the financial instrument is said to enter a buy position when the slow-moving average crosses above the fast-moving average and similarly enter a sell position when the slow-moving average crosses under the fast-moving average.

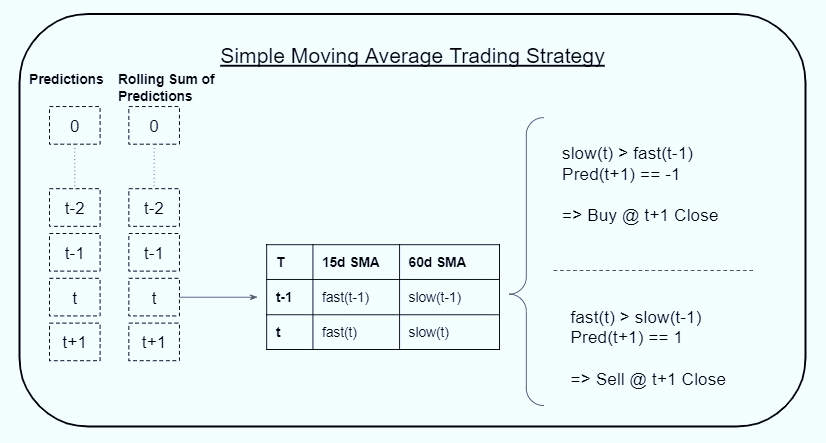
For our data, the slow-moving period was fixed at 60 days and the fast-moving period at 15 days. As the predictions did not provide continuous values, the SMA strategy was modified to work with the binary classification obtained. Thus, the rolling averages were calculated using the sum of the predictions over a window, like the approach in the Ensemble Voting Optimization.

Figure 30: SMA Trading Strategy

The SMA trading strategy worked as demonstrated in Figure 30 above. For the entire time, the sum of the predictions was calculated over a rolling window. For a given time period **(t)**, the average of these sums was calculated for the past 15-day and 60-day period to represent the fast and slow, respectively. If the 15-day sum would cross over the 60-day sum of the previous day it would indicate a sell signal. Similarly, if it went under, it would represent a buy signal. The additional element of the prediction for the following day was also considered to decide the trading signal.

However, this approach seemed to have limitations that would hinder the backtesting strategy for this project. The primary issue with this approach was the usage of past values having a higher weightage than the following day’s prediction to calculate whether to make a trade. Intuitively it would make more sense to perform the SMA with the true values of the market rather than the predictions which would make the predictions redundant for this specific trading strategy.

Additionally, this strategy works to make a single trade of either a buy or a sell in a day. It builds a sort of a buy and hold strategy where the commodity and bought and held for a certain number of days before it is sold. This is unfavourable to our forecasting approach as the predictions are only calculated for the next day and hence a strategy that aims to trade with expectations in the longer run is not optimal for our approach. Moreover, holding stocks over periods of time also leads to increased amounts of commission which was similarly detrimental to our strategy of aiming to obtain positive returns on our predictions.

**7.2 Market Intraday Strategy**

To evade such issues, a second and more appropriate strategy was chosen for our backtesting framework. This strategy incorporated intraday trading. This strategy overcomes the limitations of the SMA by not incorporating any of the previous values since the team hypothesizes that previous values would not be very helpful at this stage as they have already been used to make the prediction for the future time period. It completes the entire trade within the duration of a day which is in accordance with our forecasting duration. Moreover, the trading fee on intraday trading is only applicable to the values of shares sold instead of the total transaction thereby reducing costs too. Although quite uncommon for backtesting purposely, this strategy is suitable for our approach.

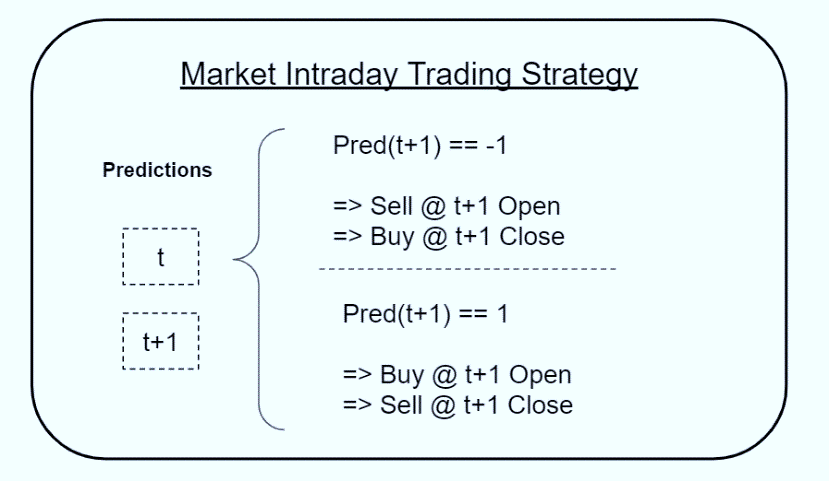
This strategy looked at the prediction for the following day and decided whether to make a short trade or a long trade within that day. If the prediction for the following day turns out to be positive, a long trade is made by buying the financial instrument at the opening price followed by selling it at the close price. Conversely, if the prediction for the following day is negative, a short sell is made at the opening price and a buy back is performed at the closing price. Figure 31 below demonstrates the functioning of this strategy given the predictions of a certain day and its following day.

Figure 31: Market Intraday Trading Strategy

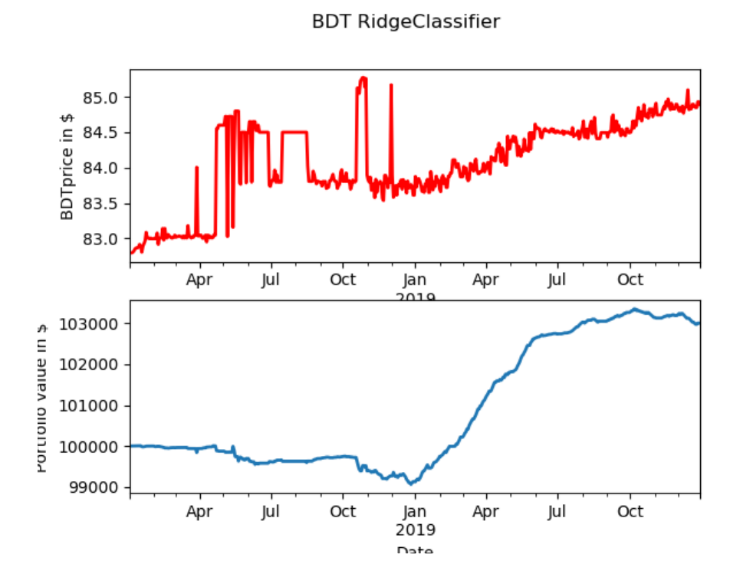
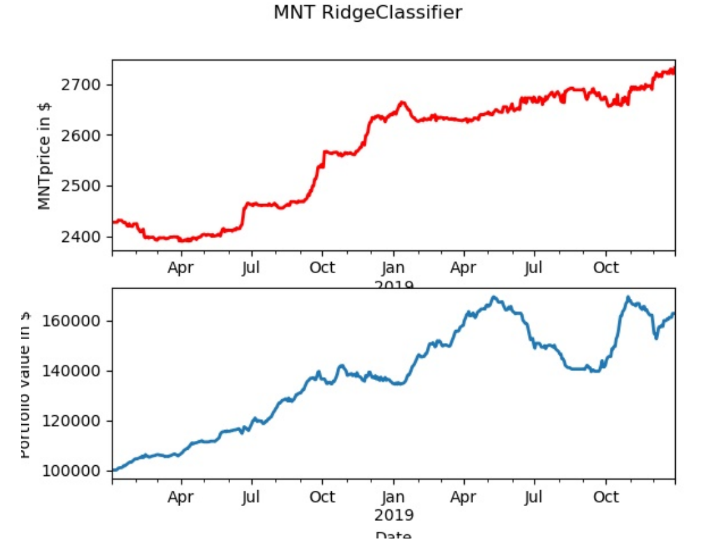
For testing purposes, the period selected is only the last year of the decade, 2019. The number of shares to be bought/sold is set to a 500 at every trade. The initial equity with which the backtesting procedure is conducted is set to $100,000. In addition, the trading fee is also implemented and is set at a value of 0.2% (0.002). The testing period for the backtesting strategy was set to 2019. The backtesting results shown as follows in the two figures below.

Figure 32: Backtesting Results (BDT)

Figure 33: Backtesting Results (MNT)

As seen in the figures, our prediction models combined with the trading strategies yield significant results. It is interesting to note how the gain in positive returns is proportional to the range of the actual close prices. Figure 32 shows a 3% equity gain for BDT whereas figure 33 shows a 60% gain for MNT over the same time period owing to the different scales of their indices. The backtesting also shows that high volatility often leads to significant jumps in equity gain/loss across the trade period. Similar results have also been seen with the market indices showing a significant equity gain using the proposed backtesting approach.

Thus, the backtesting strategy has been seen to be favourable for the time period selected using the most optimal predictions for each of the indices. This backtesting strategy demonstrates how the forecasted predictions obtained are viable to be used in the current market scenario to generate positive equity returns given a starting capital. This concludes the deliverables for the phases of this project.

**8. FUTURE RESEARCH**

Having obtained considerable results using the proposed project plan, our team aims to continue work on some aspects of the project with the aims of improving it for further use. Areas have been identified for potential further steps that can be worked upon and will be pursued by the team after completion of the deliverables of the project.

**8.1 Testing**

To continue verifying the integrity of our results, the execution of the model will be continued to see whether successful results can be obtained in the future time periods as well. This will ensure that the backtesting continues to support positive returns through different economic environments. Given the current economic climate of 2020 owing to the pandemic, backtesting over this period will help us gain more insights as work is continued in the future.

**8.2 Scalability**

The project being built with an object-oriented approach has made it easy to continue scaling on the modelling and backtesting approach. The scripts for data collection and pre-processing are built in such a way that they are to be run in a sequential order to generate the required results. By providing greater configurability to the project, there is capacity for greater code reusability. The team believes that the potential for further developing the codebase with ease for this project can lead to focussing further efforts into packing the project into a full scale application that can be used on a periodic basis to generate forecasts for efficient trading.

**6. CONCLUSION**

This project aims to create a tool that provides the service of forecasting financial data in the domain of cross-market analysis. Forecasts of market indices on an international level are to be obtained using historical time series data along with currency exchange rates, public news and social media sentiment. Previous works have been reviewed to gain a deeper understanding of the methodologies used and decide on the approach to be pursued. Based on research conducted, a suitable methodology was decided, and work was carried forward to achieve the desired goals.

Financial data was collected from the various APIs and scrapers were built to extract historic data from social networks. This collected data was further pre-processed using sentiment analysis tools and mathematical transformations to get the relevant inputs for the following phases. The prepared data was then subject to exploratory data analysis to identify patterns in the data and understand the nature of the data favourable to forecast indicating the returns of the index prices as the suitable variables.

Experiments with different prediction models for forecasting led to selection of best methods. Having progressed from basic regression methods, through time series models, classification models were finally selected as the suitable forecasting methods. These methods were further optimized using walk forward and ensemble learning approaches to enhance the results of forecasting. Having identified the most optimal models, these predictions were backtested using a viable trading strategy working on intraday returns.

The forecasting methods returned predictions with a better than average predictions. The backtested strategies having generated significant positive returns indicated the viability of our forecasting technique to use in the financial markets.

**9.1 Work Distribution**

The workload for this project was equally distributed between myself and my colleague Karan Mahajan as we worked through the different phases of the implementation. The parts of the project completed by me include making the website, data collection, numeric data pre-processing and the model experimentation phases. The parts completed by my colleague include sentiment analysis, textual data pre-processing, model optimization and model backtesting. To summarize, this is also represented in the table below.

|  |  |
| --- | --- |
| Shubhankar Agrawal (myself) | Karan Mahajan |
| * Project Website * Data Collection (Financial data and social media data) * Financial data pre-processing, feature engineering and Exploratory Data Analysis * Model Experimentation phases | * Performing Sentiment Analysis on collected social media data * Textual data pre-processing – Aggregating Sentiment scores * Model optimization techniques * Backtesting and Trading strategies |

Table 4: Work Distribution

**APPENDIX**

**Project Schedule**

|  |  |
| --- | --- |
| Date | Goals |
| September 15, 2019 | * Literature Review * Finalize Topic |
| September 20, 2019 | * Confirm project plan **(Deliverable 1)** * Set up project website **(Deliverable 1)** |
| October 15, 2019 | * Set up basic architecture for project * Investigate relevant tools necessary for implementation |
| October 30, 2019 | * Prepare automated scripts for collection of data * Deploy scripts on server |
| November 15, 2019 | * Collect all relevant social media data * Prepare for sentiment analysis |
| November 30, 2019 | * Aggregate results of sentiment analysis * Complete data collection |
| December 15, 2019 | * Perform data pre-processing and feature engineering |
| January 15, 2020 | * Extract features and trends for data |
| January 30, 2020 | * Prepare interim progress report **(Deliverable 2)** * Preliminary implementation **(Deliverable 2)** |
| February 15, 2020 | * Deploy initial machine learning approach * Deploy models based on neural networks * Analyse and evaluate performance of models * Start improving on top models |
| February 28, 2020 | * Deploy ensemble models and evaluate performance |
| March 15, 2020 | * Testing and improvements on models |
| March 30, 2020 | * Final review of software * Make changes if needed |
| April 30, 2020 | * Prepare final report **(Deliverable 3)** * Prepare final presentation **(Deliverable 3)** |
| May 10, 2020 | * Prepare project poster * Prepare project video |

**REFERENCES**

[1] G. Zhang, *Neural Networks in business forecasting*. USA: IGI Publishing, 2004.

[2] A. Ludhiyani, S. Joshi, R. Pathak, P. Parandkar, and S. Katiyal, "Quant techniques and regime analysis for increase in trade & business between India and Israel," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015: IEEE, pp. 442-447.

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[4] L. C. Boldt et al., "Forecasting Nike's sales using Facebook data," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 2447-2456.

[5] W. Chong and C. Pu, "Application of Support Vector Machines in Paying Rate Forecasting," 2006 International Conference on Management Science and Engineering, Lille, 2006, pp. 1494-1497.

[6] E. K. Laitinen and T. Laitinen, "Bankruptcy prediction: Application of the Taylor's expansion in logistic regression," *International review of financial analysis,* vol. 9, no. 4, pp. 327-349, 2000.

[7] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005: IEEE, pp. 192-196.

[8] W. Wang and S. Nie, "The Performance of Several Combining Forecasts for Stock Index," 2008 International Seminar on Future Information Technology and Management Engineering, Leicestershire, United Kingdom, 2008, pp. 450-455.

[9] J. McCluskey and Jianguo Liu, "US financial market forecasting using data classification with features from global markets," 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, 2017, pp. 965-969.

[10] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," *Stanford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf),* vol. 15, 2012.

[11] K. Zhang, L. Li, P. Li, and W. Teng, "Stock trend forecasting method based on sentiment analysis and system similarity model," in *Proceedings of 2011 6th International Forum on Strategic Technology*, 2011, vol. 2: IEEE, pp. 890-894.

[12] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, "Exploiting topic based twitter sentiment for stock prediction," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 24-29.

[13] S. Yıldırım, D. Jothimani, C. Kavaklıoğlu and A. Başar, "Classification of "Hot News" for Financial Forecast Using NLP Techniques," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4719-4722.

[14] “中文文本探勘初探：TF-IDF in R Language,” Mr. Opengate, 14-Apr-2016. [Online]. Available: https://mropengate.blogspot.com/2016/04/tf-idf-in-r-language.html. [Accessed: 05-Dec-2019].

[15] S. Pan, X. Jiang, J. Jiang and G. Long, "Cross-Domain Deep Learning Approach For Multiple Financial Market Prediction," presented at the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018.

[16] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[17] “Making Web Crawlers Using Scrapy for Python,” *DataCamp Community*. [Online]. Available: <https://www.datacamp.com/community/tutorials/making-web-crawlers-scrapy-python>.

[18] L. Cao and F. Tay, “Support vector machine with adaptive parameters in financial time series forecasting,” IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1506–1518, 2003.

[19] Kostiainen K. Development of Trading Algorithm Backtest Environment, 2016.

[20] M. Popov and R. Madlener, “Backtesting and Evaluation of Different Trading Schemes for the Portfolio Management of Natural Gas,” SSRN Electronic Journal, 2014.