Global Currency Monitoring System

BY

SATISH A/L PRABHAGAR @ NAGAIAH

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Global Currency Monitoring System

Academic Session:2023/2024

I ____

(SATISH A/L PRABHAGAR @ NAGAIAH)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

151,Kampung Tersusun______

Batu Lima , Tambun_________

31400,Ipoh Perak___________

__Abdulkarim Kanaan Jebna________

Supervisor's name

Date: _____________15/9/2023________

Date: 15/09/2023____________

FACULTY/INSTITUTE OF INFORMATION SYSTEMS

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __15/9/2023________________

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Satish a/l Prabhagar @Nagaiah***__________________________ (ID No: ***20ACB06649***) has completed this final year project entitled "__Global Currency Monitoring System__________" under the supervision of _____Dr Abdulkarim Janaan______________________________ (Supervisor) from the Department of ____________FICT______________, Faculty/Institute* of _Information System Engineering___________________ , and _____________________ (Co-Supervisor)* from the Department of _________ FICT _______________, Faculty/Institute* of _____________Information System Engineering ______________.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Satish a/l Prabhagar @ Nagaiah*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____

Name : Satish a/l Prabhagar @ Nagaiah

Date : ___________15/9/2023______________

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Abdulkarim Kanaan Jebna who has guided me throughout the process of understanding and developing my project which relates to the predictive model of using RSI to forecast Global currency values via machine learning . Dr Abdulkarim always clears my doubts and gives clear explanations if I ever come across a field that I would need to assistance in .

To a very special person in my life, who is none other than my dearest elder sister Alysheea Phoebie who has always been my pillar in my life in supporting me always . Finally, I must say thanks to my parents for their love, support, and continuous encouragement throughout the course.

ABSTRACT

In recent years, there has been a growing popularity in currency trading within the foreign exchange (Forex) market. Traders consistently seek novel strategies to gain an edge in predicting market trends and executing profitable transactions. This research focuses on the construction of a machine learning model that use a combination of technical indicators and fundamental factors to predict Forex prices and RSI values. The model's training method incorporates a neural network, which leverages past data to acquire knowledge of the patterns and interconnections among different market elements. Metrics such as accuracy, precision, recall, and F1 score are commonly employed in the evaluation of the proposed model. However, there are other measures that play a significant part in model evaluation, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). The results suggest that the model exhibits a higher level of proficiency compared to traditional statistical models when it comes to predicting RSI values and changes in Forex prices. The proposed model is evaluated in comparison to the baseline model in terms of its accuracy, and the results demonstrate that the proposed model outperforms the baseline model in terms of accuracy. Furthermore, the F1 score of the proposed model is compared to that of the baseline model, revealing that the suggested model achieves a higher F1 score in comparison to the baseline model. Furthermore, the present study aims to examine the impact of various technological aspects on the performance of the model. The results indicate that specific indicators, such as the moving average, have a significant role in predicting the values of the relative strength index (RSI) and Forex prices. This study highlights the potential of machine learning in the financial industry, particularly in the prediction of market movements and its application in facilitating trading decisions. In general, the research underscores the potential of machine learning inside the financial sector. The proposed methodology exhibits promise in augmenting the precision of Forex price and RSI value forecasts, hence potentially resulting in more profitable trades for traders. The acronym RSI denotes the relative strength index. In order to enhance the precision of the model, future research endeavours may explore the potential inclusion of basic elements with intricate technical indicators.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

β	beta
Ω	Ohm (resistance)

LIST OF ABBREVIATIONS

5G	Fifth Generation
API	Application Programming Interface
CPU	Central Processing Unit
GPIO	General Purpose Input Output
IOT	Internet of Things
IP	Internet Protocol
RAM	Random Access Memory

Chapter 1

Introduction

Currency, also referred to as money, serves as the principal unit of economic measurement employed globally in our dynamic globe. The term "currency" specifically pertains to fiat money that is now being utilised in the market.In a similar spirit, from the early 1970s, states began to adopt either fixed or floating exchange rate systems, thereby granting central banks or market forces the authority to influence global currency rates [1]. The foreign exchange (FX) market, encompassing global currencies, now has the distinction of being the most significant financial market worldwide in terms of trade volume. The measurement of market activity is determined by the volume of transactions conducted by buyers and sellers who are interested in acquiring currency pairs such as EUR/USD. This particular currency pair involves the national currency of Europe, the Euro, which is paired with the globally recognised currency known as the United States Dollar (USD). When traded on an exchange, all of these currencies are paired with the USD in order to assess the relative strength of other currencies. The GBP/USD currency pair serves as an indicator of the performance of the Great British Pound in relation to the US Dollar. The observation of the significant advancements and expansion of forex trading due to the pervasive influence of digitalization is a subject of great fascination. The accessibility of the internet has facilitated a wider reach of currency trading among individuals.

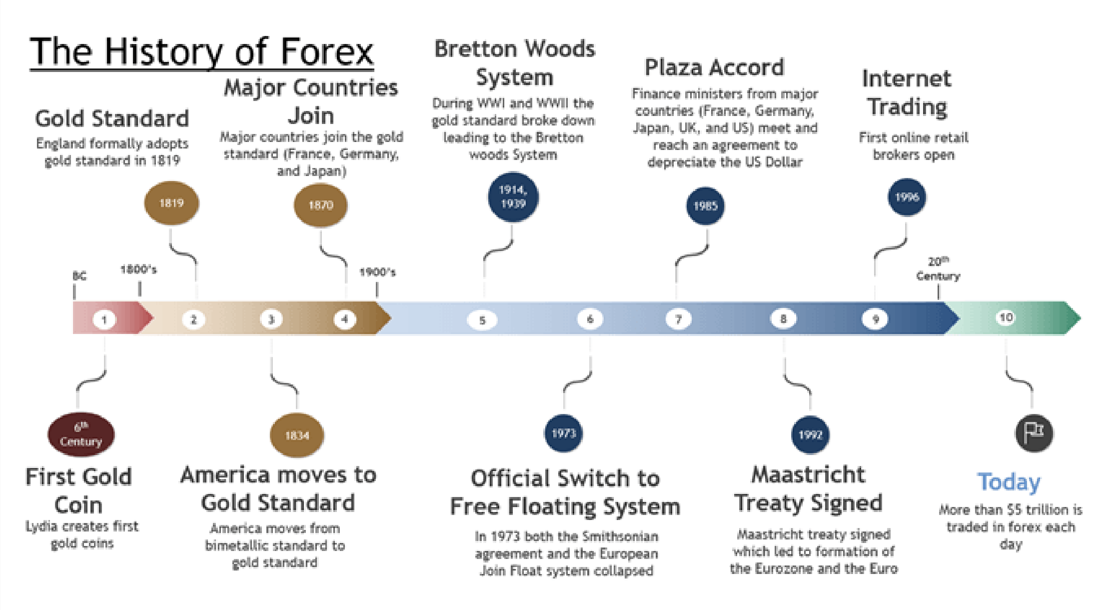


Figure 1.0 shows the illustrates the timeline of Forex History till todate , source www.dailyfx.com

Machine learning is a branch of artificial intelligence that employs algorithms to acquire knowledge from data and generate predictions. The utilisation of this approach has witnessed a notable surge in popularity within the financial sector, specifically in the domain of forecasting foreign exchange rates. Machine learning algorithms has the capability to discern patterns within past data and employ them for the purpose of forecasting future price fluctuations. This tool possesses significant utility for traders, as it has the potential to enhance decision-making processes and augment trading performance. Machine learning can also be employed for the purpose of discerning correlations between various currency pairs and additional aspects, such as economic indicators, so facilitating traders in gaining enhanced comprehension of the financial markets and enabling them to make more judicious trading choices. The monitoring of price can be consistently achieved through the use of indicators. Machine learning enhances the effectiveness of indicators by employing forecasting techniques to determine the future movements of an asset. Indicators play a crucial role in the prediction of forex prices as they offer a means to analyse the market and discern probable trading prospects. Indicators may be employed for the purpose of discerning patterns, quantifying the rate of change, and assessing the robustness of a currency pair. In addition, they can serve the purpose of identifying instances of overbought and oversold market circumstances, as well as determining prospective places of entrance and departure. The utilisation of indicators enables traders to acquire significant insights into the market, hence facilitating the enhancement of decision-making processes in forex trading.

1.1 Problem Statement and Motivation

Problem statement highlights the limitations that can be found in previous Machine learning models and algorithms used in it :

1.1.1XGBoost [3]

The Extrapolation Capability of Tree-Based Models Is Limited[4]

This issue may be seen as the most fundamental challenge associated with tree-based models in a generic context. The choice between employing a solitary decision tree, a random forest including one hundred trees, or an XGBoost model consisting of one thousand trees does not yield any discernible variance in outcomes.The XGBoost technique is not advisable for cases where continuous output is utilised, as it has the potential to generate predictions that are not statistically significant in relation to the intended output. In terms of generating predictions, these algorithms generally have difficulties in extrapolating target values beyond the boundaries of the training data due to the partitioning of the input space by tree-based models in each specific problem. In the context of classification tasks, this obstacle is generally not considered substantial. However, in the case of regression tasks, which involve forecasting a continuous output, it is evidently a constraint. The following is an illustration of a challenge encountered when employing XGBoost for the purpose of forecasting stock prices:

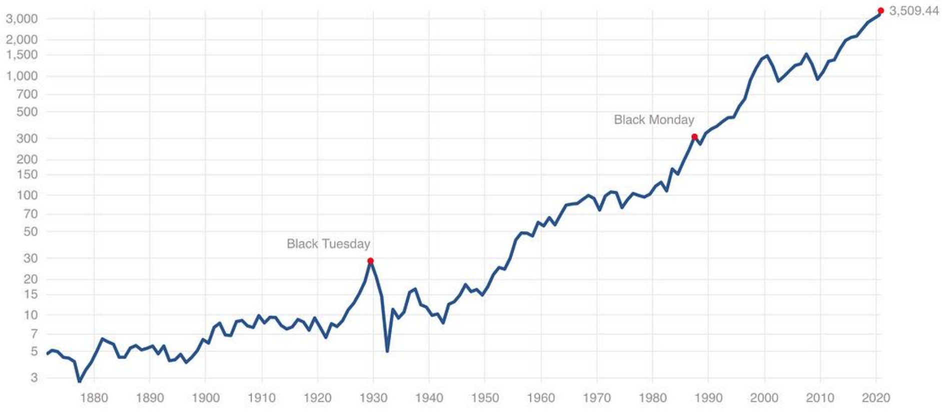


Figure 1.1 shows the S&P 500 Historical Prices. Source: Standard & Poor's reproduced from multpl.com.

The above shows the performance of S&P 500[5] from the year 1800 to 2020 , S&P 500 is a stock market index that track the performance of 500 large-capital companies listen on American stock exchanges, also considered to be a key indicator of overall health of U.S Stock market . If we examine the patterns of a well-known stock market index such as the S&P 500 over the course of the past fifty years, we will discover that the price of the index has experienced highs and lows but has, on average, increased over the course of that time period. According to the analysis of past data, the S&P 500 index has a historical annual return of approximately 10 percent, which indicates that the price increases by approximately 10 percent on average each year. If you try to estimate the price of the S&P 500 using XGBoost, you will see that it is possible that it may predict declines in prices, but that it does not catch the overall trend of increasing prices in the data. To be fair, predicting stock market prices is an exceedingly tough problem that even machine learning has not been able to solve. Yet, the point is that XGBoost is unable to anticipate rises in prices that go beyond the range that is contained in the training data.[6]

Extrapolation and over-fitting problems of Random Forest[7]

1. Disadvantages of RF are also covered in a lot of literature. The following four RF drawbacks have been widely noted:

2. It can over-fit values without an analyst seeing it, depending on the data and assumptions made about the data.

3. It only makes accurate predictions when given enough training data in the feature space. Poor performance might result from extrapolation, or making predictions outside the training domain.

3. It could be computationally costly, with the computational load rising exponentially as the number of covariates increases.

4. It is sensitive to errors and typos in the data and requires high-quality training data.

1.1.2. Linear Regression[8]

Linear regression is a method that can be employed to effectively model linear relationships between variables. The algorithm has a low computational time, possesses a straightforward implementation and comprehension, and may be simply elucidated. There is a potential for underfitting when employing simple models, which may result in the inability to accurately capture intricate patterns within the data.

The validity of the model estimation and regression predictions is contingent upon the utilisation of a certain data range [9]. Beyond that specified range, there is potential for the connection between the independent and dependent variables to undergo modifications. Put differently, there is uncertainty over whether the shape of the curve will undergo any alterations. If such were the case, our estimates would be deemed inaccurate.

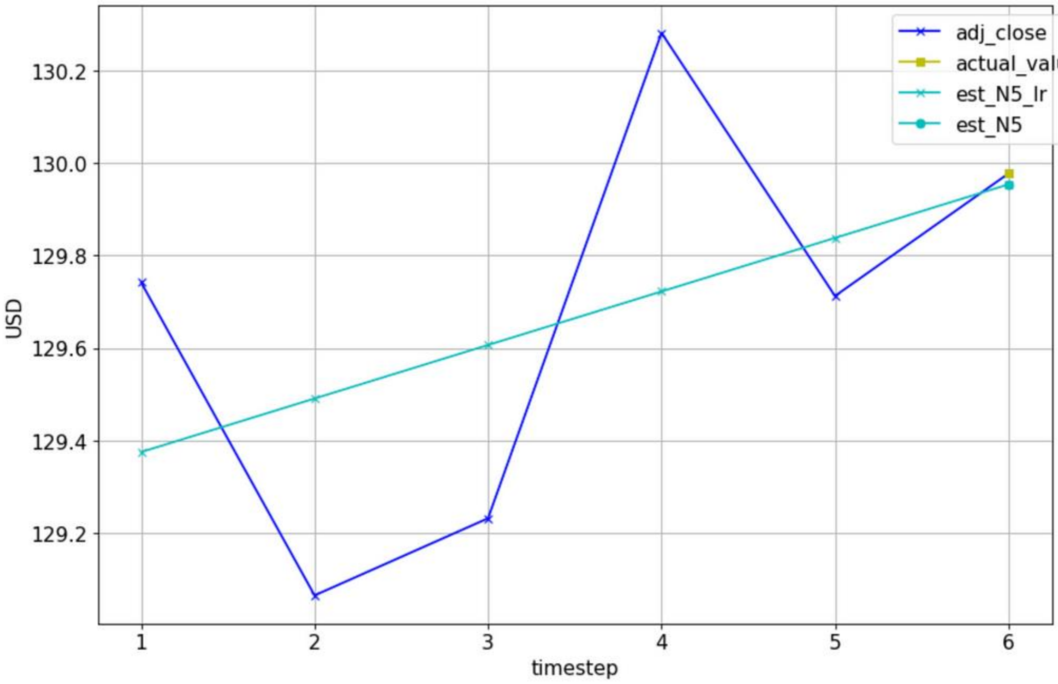


Figure 1.2 shows results of using Linear Regression in predicting Vanguard Total Stock Market ETF (VTI) by predicting the next value using linear regression with N=5 from 2015–11–25 to 2018–11–23 .

However, there exist significant limitations associated with the utilisation of linear regression in predicting currency prices [10].

- Foreign exchange (forex) values are subject to various influences, including economic data, geopolitical events, and market emotions. These factors play a significant role in determining the volatility of forex prices over time. However, capturing these intricate patterns might pose a challenge. It is conceivable that linear regression may not adequately account for all of these aspects, resulting in inaccurate predictions.

- Linear regression models are predicated on the assumption of a linear association between input and output variables. This assumption renders them susceptible to underfitting, since they may oversimplify the underlying relationship. The model's prediction ability may be compromised due to underfitting of the data, maybe resulting from a non-linear relationship.
- The user's text lacks sufficient information to be rewritten in an academic manner.The presence of outliers can have a substantial impact on the performance of a linear regression model due to the model's high susceptibility to their influence. The accuracy of predictions may be compromised due to the presence of outliers, which have the potential to introduce distortion to the regression line.
- The assumption is made that the input variables are independent of each other. Linear regression is predicated on the underlying premise that the variables included in the analysis exhibit no discernible correlation or relationship with one another. The presence of multicollinearity may give rise to certain concerns.
- Linear regression may encounter difficulties in handling high-dimensional data when the number of input variables becomes very big. The aforementioned phenomenon has the potential to result in computational inefficiencies and overfitting.

3. Failure in including indicators in machine learning models to improve accuracy of predictions

According to Soni (2011), the study utilised data collected from multiple news channels to investigate the application of TF-IDF features in predicting forex values for the subsequent trading day within the framework of Hidden Markov Models (HMMs). In order to calculate the word score, the authors performed computations on TF-IDF weights. Ultimately, a Hidden Markov Model (HMM) was established with the objective of determining the probability of a given sequence. The present model incorporates the inclusion of probabilities associated with the act of switching values. Nevertheless, enhancing the dataset's magnitude, utilising diverse machine learning techniques, or augmenting the quantity of technical indicators and input attributes can result in improved accuracy. The researchers of this study noticed a discernible pattern of both positive and negative predictions that exhibited partial concordance, while also demonstrating an error range of 0.2 to 4%.

Incorporating indicators into a machine learning model has the potential to enhance the accuracy of FX price predictions. This objective can be achieved by the enhancement of capturing intricate linkages, minimising the impact of extraneous factors and unpredictability, integrating domain expertise, enhancing the comprehensibility of the model, and considering the prevailing market dynamics.

1.2 Research Objectives

The aim of this project is to solely create a currency monitoring system that allows traders to use the help of machine learning to allow indicators to be more useful in terms of price predictions and helps to increase the accuracy of predicting possible trading signals towards a traders in the Forex market . Below are some of the objectives of my system :

I. Compare and contrast among different currencies that allows traders to find trading opportunities everyday using the help of predictive indicators and signals with the help of machine learning algorithms to forecast indicators .

II. Provide traders with information such as fundamental data(Current economical news) towards currencies .

III. Provide basic information of currency such as OHLC values .

IV. Predict indicators such as RSI and Exponential Moving Average .

V. Implement machine learning to identify trading opportunities with the help of Indicators to signify trend , buy or sell signals .

1.3 Project Scope and Direction

The scope of my project is to implement machine learning methods to trading indicators to help further increase and improve the use case of indicators such as RSI and Moving Average to help predict and gain insights about trend and possible buy/sell signals towards a particular currency in the forex market .

1.4 Contributions

Machine learning techniques have been established for a considerable duration. However, with the recent surge in interest around artificial intelligence (AI), machine learning has experienced a concurrent growth. This is because machine learning is an integral component of the AI framework [12]. Therefore, the integration of machine learning techniques in the field of trading has facilitated enhanced opportunities and enabled traders to make more informed decisions in completing trades across various markets. The utilisation of machine learning techniques enables the anticipation of outcomes with reduced bias, hence enhancing the precision of predictive models. The significance of market volatility and quick change in forex trading is noteworthy. The utilisation of computers eliminates human bias from the equation. Currently, the majority of predictive algorithm solutions in trading primarily focus on forecasting asset prices, hence enhancing decision-making capabilities by providing more informed insights. However, my technique places emphasis not solely on price, but rather on predicting the spectrum of buying and selling activities. This is achieved by utilising technical indicators, rather than relying solely on past price performances.

CHAPTER 2

Literature Reviews

2.1 Previous works on Machine Learning

2.1.1 XG Boost: To predict prices of assets , applicable to both forex or stocks[13]

The machine learning research in question was initially presented by MTSZKW on the Kaggle platform. This specific project employs the XG Boost Regressor algorithm, which is a component of the XGBoost toolkit. XGBoost is recognised as an optimised distributed gradient boosting library that has been specifically engineered to exhibit excellent efficiency, flexibility, and portability. The functionality of the system is facilitated by the utilisation of the Gradient Boosting framework. This framework operates by constructing a series of weaker prediction models in a sequential fashion, with each model attempting to forecast the residual errors left by the preceding model. A less robust learning model operates using the principle of ensemble learning, whereby the model exhibits increased flexibility in making predictions while simultaneously exhibiting reduced dependence on data (lower data sensitivity). The two most often employed techniques in ensemble learning are bagging and boosting. The present project has a greater emphasis on Boosting approaches, wherein models are trained in a sequential manner.

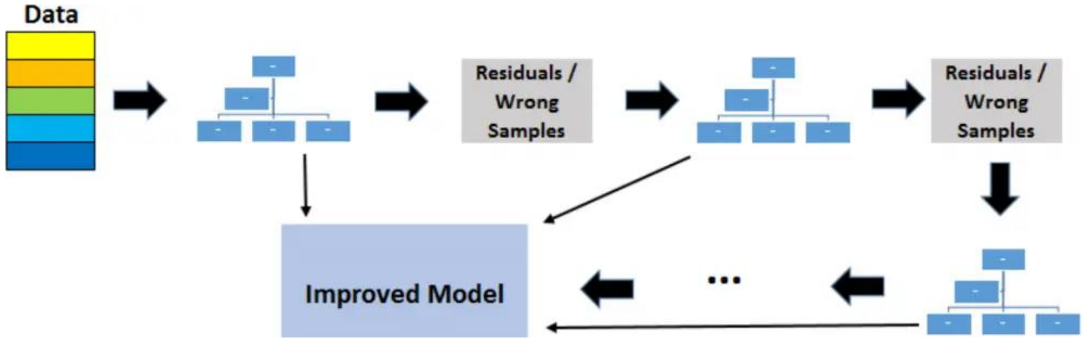


Figure 2.1 Shows the visualization of data is process via Boosting

The provided diagram illustrates the method by which the model detects residuals of the weak learners and then fits the data in a manner that allows the new learner to process an updated version of the data, excluding the previously recognised residuals. The ultimate model is a result of the measures implemented to minimise the residuals, leading to the attainment of a robust learning model. The diagram presented below provides an overview of the implementation process of the XGBoost method for constructing a predictive model.

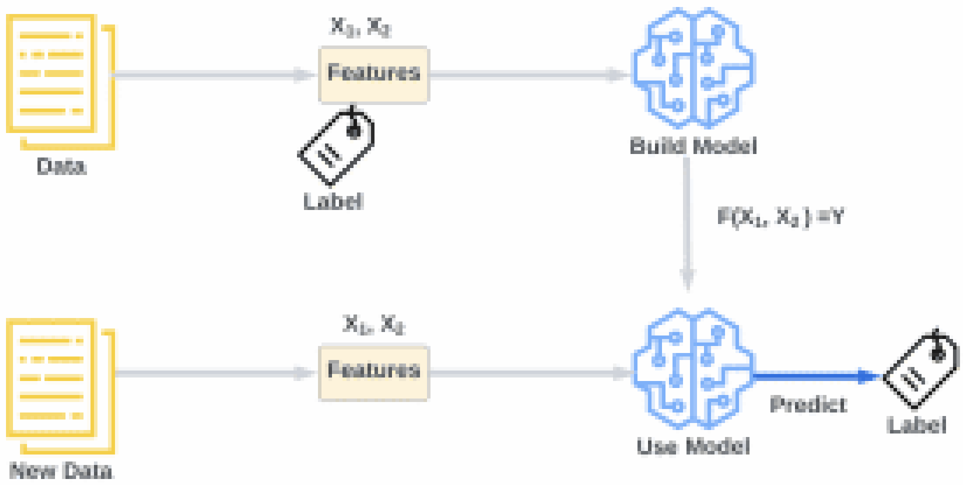


Figure 2.2 Shows the overview of XGBoosting

2.1.2 Strengths and Weakness

<u>Strengths</u>

XGBoost is a widely used algorithm mainly known for its quick and high accuracy predictive models [14] which one can be used for building machine learning and Ai .Below are some of the expertise of the algorithm :

1. The accuracy of XGBoost is superior to that of competing algorithms, especially when applied to structured datasets.
2. Large datasets are no problem for XGBoost because it was developed to be both scalable and efficient. Understanding the relative relevance of different features in the dataset can be aided by using XGBoost's built-in feature importance ranking.

3. XGBoost is versatile since it can be applied to both regression and classification problems, and it also accommodates a wide range of loss functions.
4. Overfitting can be avoided with the use of regularization methods including L1, L2 regularization, and early stopping, all of which are implemented in XGBoost.

XGBoost has become a popular algorithm that is used to solve complex problems that requires the help of multiple basic algorithms such as Regression , Random Forest and etc . Hence , works such as New York Taxi Fare Prediction is one example of how XG model was implemented to train , test and show the output of the prediction .

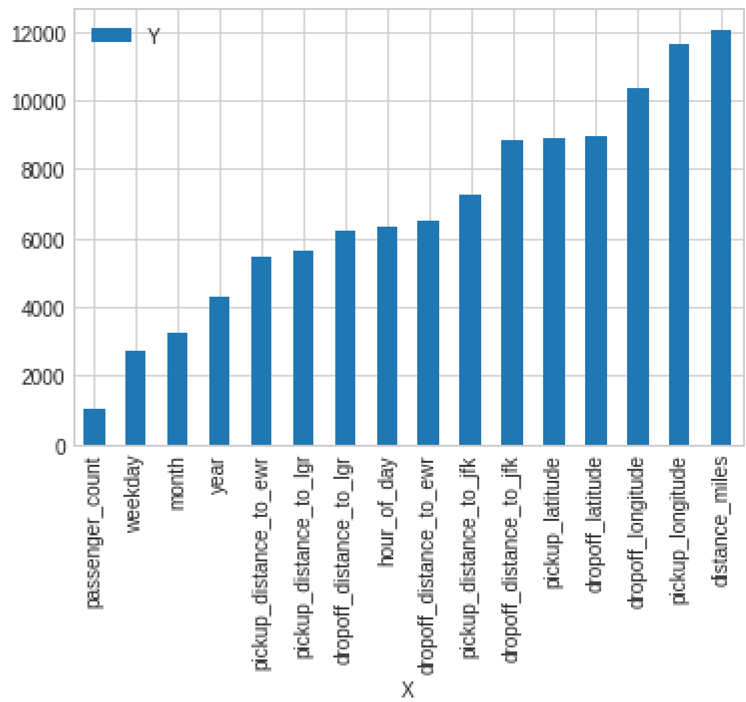


Figure 2.3 Shows the results of the XGBoost predictive model of the far amount according to features

<u>Weakness</u>

XGBoost may seem to be a good algorithm under specified conditions but it too has its limitations when it comes to machine learning . The limitations of XGBoost are stated below :

1.Complexity: XGBoost is a complex algorithm, and it requires a lot of tuning and optimization to achieve good performance.

2.Overfitting: Although XGBoost has built-in regularization techniques, it is still prone to overfitting, particularly when dealing with noisy datasets.

3.Time-consuming: XGBoost is a computationally expensive algorithm, and it can take a lot of time to train on large datasets.

4.Memory Usage: XGBoost requires a significant amount of memory to store its model parameters, particularly when dealing with high-dimensional data.

5.Black Box Model: XGBoost is a black box model, which means it is difficult to interpret how the algorithm arrived at its predictions.

2.2.1 Time Series Analysis

A subfield of both statistics and machine learning, "Time Series Analysis" focuses on temporally-indexed data. The measurements that make up a time series are taken at regular intervals in time, typically once each hour, day, or month. Below are some predictive model that are built using Time Series Analysis[17] :

- Autoregressive Integrated Moving Average (ARIMA) is a well-known method for time series forecasting. In order to model time series data, it employs both autoregressive (AR) and moving average (MA) features. In order to forecast future values from past data, ARIMA models use differencing to transform a non-stationary time series into a stationary one. Future values are predicted using "auto" correlations and moving averages over residual errors in the data.

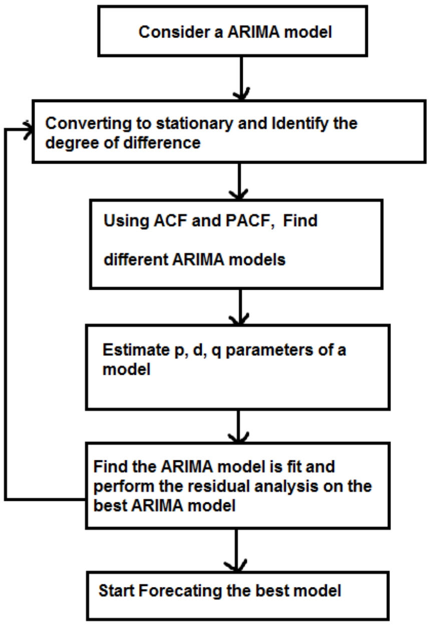
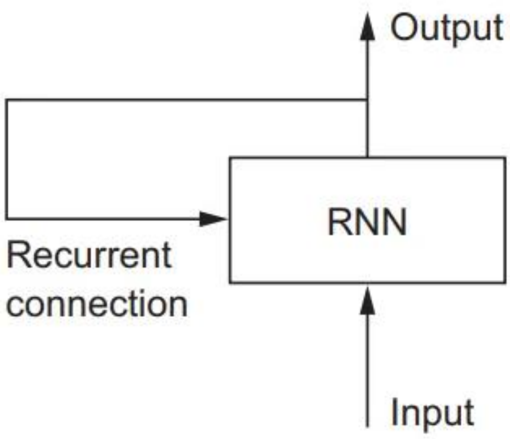


Figure 2.4 Shows the flow diagram of the development process using ARIMA model

- The recurrent neural network (RNN)[18] class known as Long Short-Term Memory (LSTM) excels at time series analysis. As such, it finds widespread application in areas as diverse as stock price forecasting, weather prediction, and NLP because of its ability to capture long-term dependencies in time



series data.

Figure 2.5 Shows the flow of how data is used to process in RNN model

In a recurrent system, the output from one time step is fed back into the input for the following time step. The model looks at both the current input and what it has learned from previous iterations at each stage.

- Facebook's Prophet[19] is an additive model-based time series forecasting system. It has in-built support for adjusting for trend shifts, holidays, and other events that may have an impact on time series data, making it ideal for handling seasonality in time series. Trend, seasonality, and holidays are the building blocks of the underlying algorithm, which is a generalized additive model. I think I've already remarked that FB Prophet does a fantastic job of capturing seasonality and trend, two key but hard-to-quantify components of a time series analysis. It's a decomposable model, so it's simple to get at the model's coefficients and assess how seasonality, trend, holidays, and other regressors affect the model's predictions.

- For both short-term and long-term time series forecasting, XGBoost, a gradient boosting method, is a great choice. It works well with time series data that has a large number of input characteristics, and it is resilient to missing data and non-linear correlations between the features and the outcome variable.

Time series techniques for machine learning require that the input data be transformed into a sequence of observations across time. After the data is in the right format, you may model the time series and forecast future values using the algorithm of your choosing. The quality of the data, the algorithm used, and the fine-tuning of the algorithm's parameters all contribute to the reliability of the predictions.

Strengths

- Capability to Accurately Forecast Future Values Based on Historical Data Time series algorithms can be quite effective in predicting future values based on historical data.

- As opposed to using time-consuming and error-prone manual approaches, time series algorithms can automate the forecasting process.

- Time series algorithms are a flexible technique for predictive modelling because they may be applied to the analysis of a wide variety of time-dependent data, from financial data to weather patterns.

- Trends and patterns in time-dependent data can be seen with the aid of time series techniques, making it simpler to spot unexpected occurrences.

- Time-Dependent Data Can Be Analyzed Instantaneously Businesses can react swiftly to shifts in the market or other time-sensitive events by using time series algorithms for real-time analysis of time-dependent data.

Weakness

- The complexity of time series algorithms means that their efficient usage typically necessitates the possessing of specific knowledge and abilities.

- The quality of the data used to train and test an algorithm has a significant impact on the reliability of time series forecasts. The accuracy of the algorithm's predictions may suffer if the input data is incomplete or contains noise.

- If time series algorithms are not adjusted appropriately or if there's too much noise in the data, overfitting can occur.

- Inapplicable to some situations, time series algorithms are tailored for processing information that changes over time.

- The interpretability of time series algorithms is important since it can be hard to grasp how the algorithm arrived at its predictions.

2.3.1 RSI Trading Strategy [20]

The Relative Strength Index (RSI) is a momentum indicator that can range from 0 to 100. When the value of the indicator is low, it shows that the asset has been oversold, and when the value is high, it suggests that the asset has been overbought.

When trading with RSI, the easiest approach to use it is to purchase when the 14-period RSI is less than 30 and to sell when the 14-period RSI is greater than 70. It tells us that whenever the RSI reaches the low barrier, we should anticipate an increase in that indicator.

Given this anticipation, we can make use of the time series prediction to verify that the RSI in the future will climb. Using the trained time series model, it is possible to estimate the RSI for the following 5 days based on the RSI for the previous 14 days.

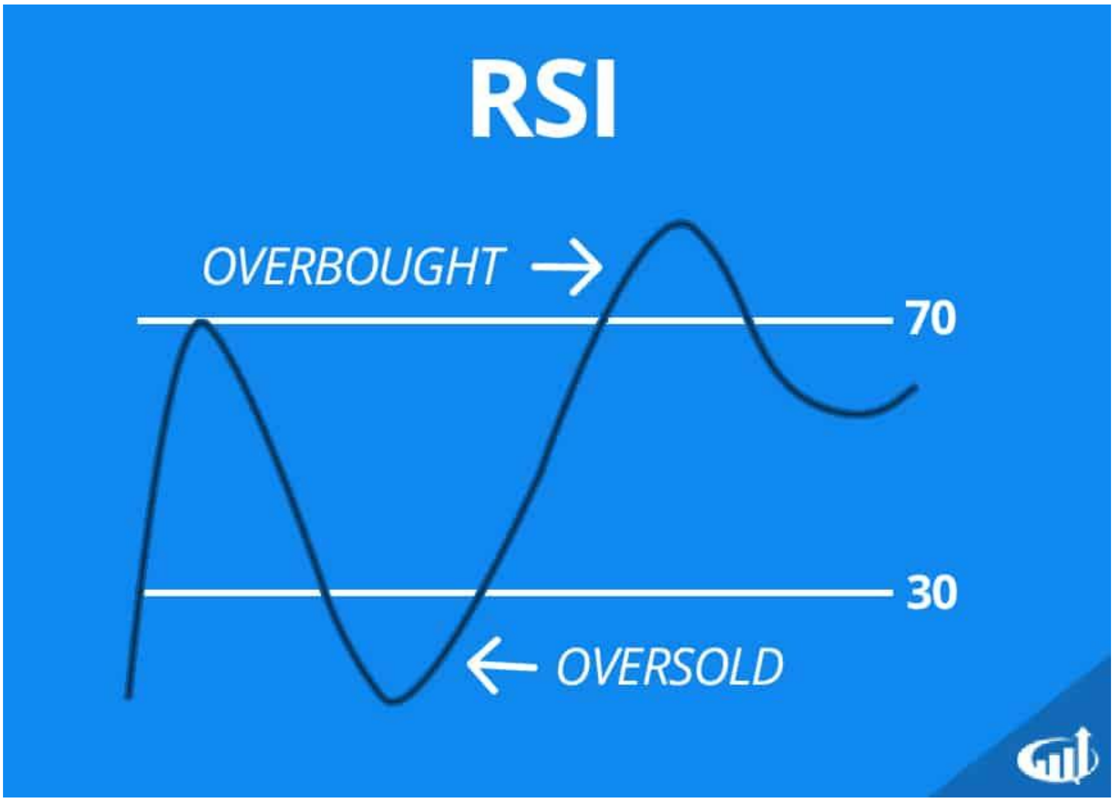


Figure 2.6 Shows the interpretation of RSI in trading terms

The Moving Average (MA) of the historical positive price difference (%) can be divided by that of the historical negative price difference (%) in order to calculate Relative Strength (RS) and Relative Strength Indicator (RSI).The formula of RSI is as below :

RS = (MA of positive difference) / (MA of negative difference)

RSI = 100–100/(1+RS)

2.1.3.1 Time-Series prediction RSI trading strategy

- **Buy signal**: If 14-period RSI < 30, and the mean of the 5-day prediction of RSI > the current RSI.

- **Sell signal**: if 14-period RSI > 70, and the mean of the 5-day prediction of RSI < the current RSI.

2.3.2 ANN(Artificial Neural Network)

Deep learning techniques are based on neural networks, which are often referred to as artificial neural networks (ANNs) or simulated neural networks (SNNs). These neural networks are a subset of machine learning [21]. The structure and nomenclature of the system are designed to closely resemble that of the human brain, thereby emulating the intricate communication patterns observed among neuronal cells.

An artificial neural network (ANN) consists of a node layer structure that includes an input layer[22], one or more hidden layers, and an output layer. Every individual node, often referred to as an artificial neuron, is interconnected with other nodes and possesses a corresponding weight and threshold. When the output of a node surpasses the predetermined threshold value, it becomes activated and initiates the transmission of data to the higher stratum of the network. If the condition is not met, there is no transmission of data to the subsequent network stratum.

The utilisation of training data is crucial for the maturation and enhancement of neural networks, leading to an incremental improvement in their accuracy as time progresses. When appropriately modified, these learning algorithms prove to be valuable tools in the fields of computer science and artificial intelligence, enabling efficient data classification and grouping processes. In contrast to manual identification conducted by human professionals, speech recognition and image recognition activities can be accomplished within minutes instead of hours. The search algorithm utilised by Google incorporates a widely recognised neural network.

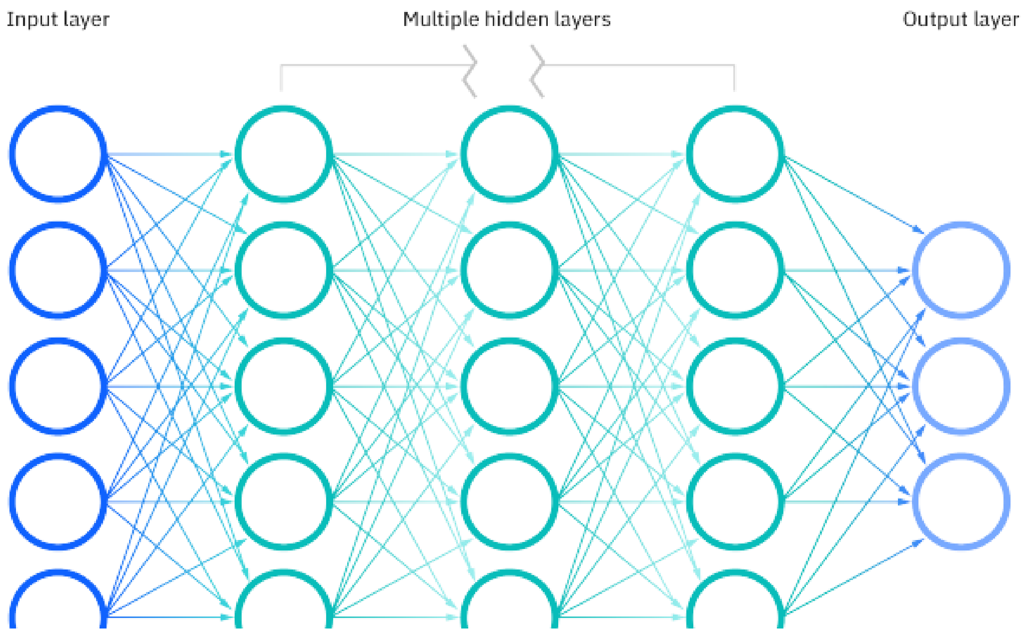


Figure 2.7 Illustrates the way neural networks work

https://www.ibm.com/topics/neural-networks

Previous studies have explored the utilisation of a combination of algorithms, including Neural-based Regressor, Recurrent Neural Networks Model, and Artificial Neural Networks (ANN), in the context of predicting stock prices using deep learning techniques. These studies have yielded promising outcomes, generating noteworthy results as output. The project in question was conducted by Abhijit Roy [23].

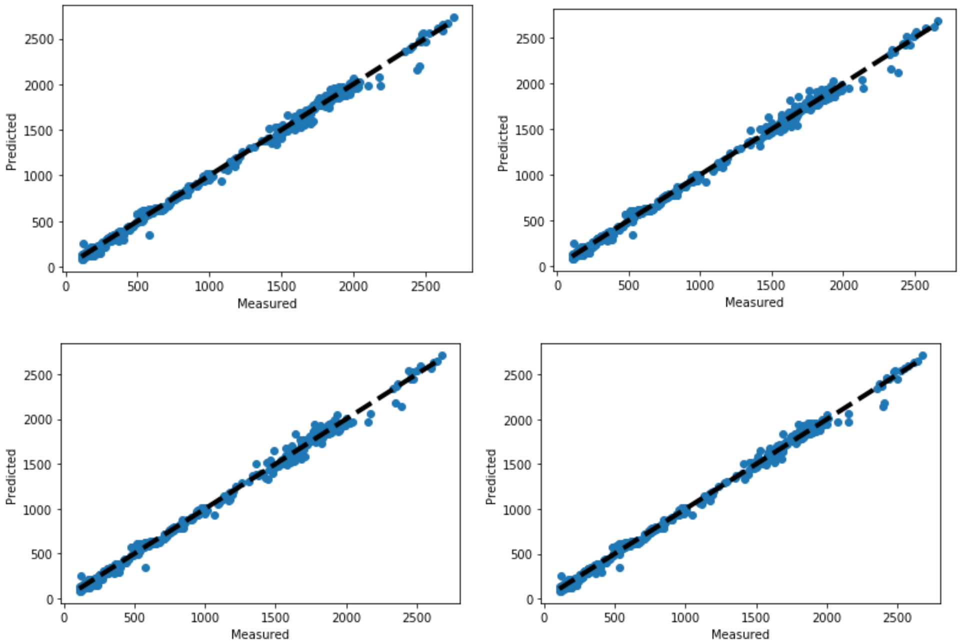
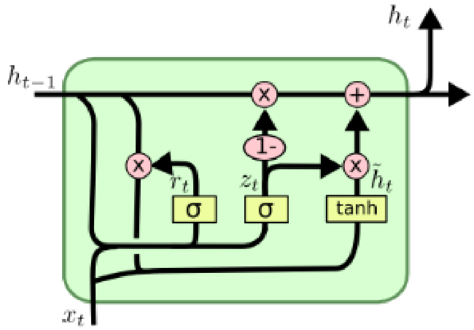


Figure 2.8 shows the regression results of OHLC values respectively using Neural-based regressor

Then comes the help of Recurrent Neural Network



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

RNN was used for past analysis towards the targeted stock . Hence working with the target stock feature only . Hence LSTM layers that work with RNN principles will be on a gated approach .

2.3.3 Beautiful Soup Web Scraping method [25]

The Beautiful Soup package is widely utilized in the Python programming language for the purposes of web scraping and parsing HTML and XML documents. The utilization of this technology offers a practical means of efficiently navigating and extracting data from web pages, becoming it a crucial instrument for the acquisition of information from online sources.

The process of online scraping entails the extraction of targeted data or information from web sites, encompassing various elements such as text, photos, links, and other relevant components. The task of accessing and manipulating the elements of a web page's HTML structure is made simpler by Beautiful Soup, which offers a collection of functions and methods for this purpose.

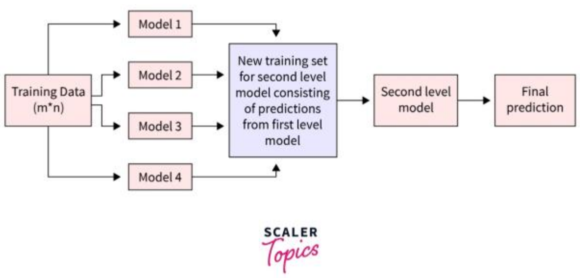
Beautiful Soup is a Python library that facilitates the parsing and searching of HTML texts with ease[25]. It proves to be a valuable tool for extracting data for uses such as data analysis, research, and several other applications. The tool's versatility and user-friendly interface render it a great asset for individuals engaged in web data manipulation using Python. Beautiful Soup is a significant tool for individuals in many professional roles, such as data scientists, researchers, and developers, as it facilitates the extraction of relevant information from the extensive array of web content available.

2.3.4 Stacking Technique

Stacking is an ensemble strategy in the field of machine learning that involves the amalgamation of predictions generated by numerous base models, with the aim of enhancing the accuracy of the final prediction. The proposed model is structured into two levels, with the first level comprising the basic models and the second level encompassing the meta-model. The foundation models encompass a variety of machine learning models, including decision trees, random forests, and support vector machines. The meta-model commonly consists of either a logistic regression model or a linear regression model.

The stacking technique involves the training of multiple base models using the provided training data. Each individual base model is trained to independently predict the target variable. The predictions generated by the basic models are subsequently utilised to train the meta-model. The meta-model acquires the ability to integrate the predictions generated by the basis models in order to generate a prediction that is more precise and reliable.

The utilization of stacking has the potential to enhance the efficacy of machine learning models across various domains. Nevertheless, it exhibits notable efficacy when applied to intricate models, such as those pertaining to deep learning.



An elementary illustration of a stacking model involves employing a random forest regressor as the meta-model, accompanied by two decision trees serving as the base models. The decision trees will undergo training using the provided training data, and afterwards, their predictions will be utilised for training the random forest regressor. The test data would be utilised for making predictions using the random forest regressor.

Stacking methodologies can be employed to amalgamate the predictions generated by a multitude of foundational models. The selection of base models is contingent upon the intricacy of the challenge at hand and the quantity of accessible data.

Advanced stacking techniques refer to the utilisation of sophisticated methods for combining many machine learning models in order to improve predictive accuracy. These techniques involve the creation of a meta-model that leverages the predictions of individual base models to

There exist several advanced stacking approaches that can be employed to enhance the performance of stacking models. One technique that is commonly employed is known as cross-validation stacking. The process of cross-validation stacking involves training the base models on distinct folds of the training data.

Subsequently, the forecasts generated by the basic models are employed to train the meta-model using a distinct subset of the training data. The aforementioned procedure is iteratively executed for each partition of the training dataset.

An further sophisticated stacking technique is referred to as stacked generalization. The technique of stacked generalization involves training the base models using the training data, and subsequently use the predictions generated by these base models to train the meta-model on the complete training data set. The aforementioned procedure is iteratively performed, wherein the forecasts generated by the meta-model are utilized to train the base models, and subsequently, the predictions derived from the base models are employed to train the meta-model.

In conclusion, it can be inferred that the information provided supports the notion that the user's

Stacking approaches are a potent methodology that can be employed to enhance the efficacy of machine learning models across several domains. Deep learning models, in particular, demonstrate notable efficacy when employing these methods. If one is seeking to enhance the efficacy of their machine learning model, I would recommend exploring the implementation of stacking methodologies.

Table 2.1 Specifications of laptop

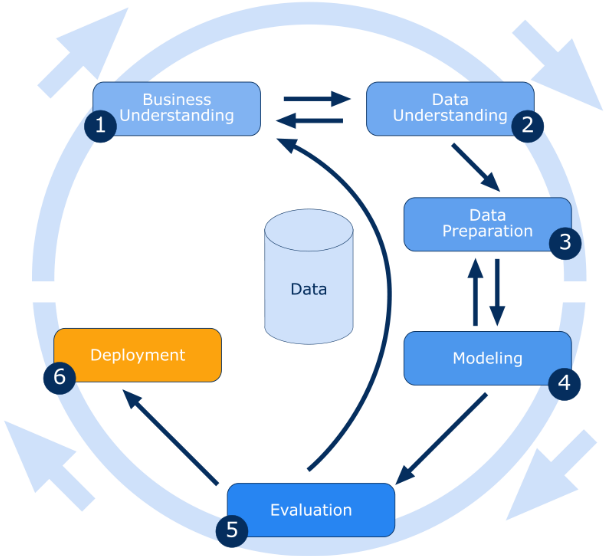
Description	**Specifications**
Model	Asus A456U series
Processor	Intel Core i5-7200U

Operating System	Windows 10
Graphic	NVIDIA GeForce GT 930MX 2GB DDR3
Memory	4GB DDR4 RAM
Storage	1TB SATA HDD

CHAPTER 3

Proposed Method/Approach

The utilisation of CRISP-DM (Cross-Industry Standard Procedure for Data Mining) is a common practise among machine learning professionals due to its systematic and organised framework for conducting data science projects [24]. The approach in question is characterised by its versatility and iterative nature, allowing for customization to suit different projects. Its primary goal is to ensure timely completion and cost-effectiveness, while still achieving the desired project objectives.



The CRISP-DM paradigm, which is widely used in the field of data mining, consists of six distinct steps. These stages include business understanding, data comprehension, data preparation, modelling, assessment, and deployment. The methodology employed in this study is characterised by an iterative approach. Subsequent to the implementation phase, the results are evaluated, and the process is then recommenced with a renewed understanding of the business problem and a newdataset.

Figure 3.1 Shows the flow of CRISP-DM

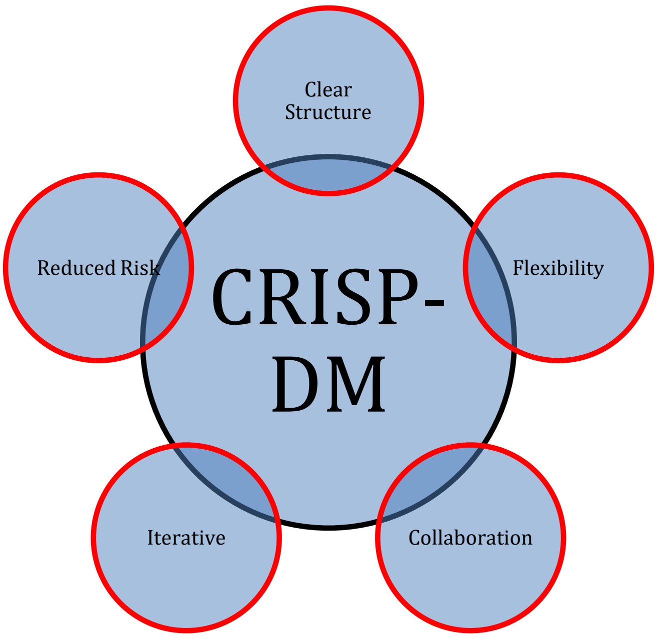


Figure 3.2 Shows the benefits of CRISP-DM approach .

The CRISP-DM framework provides a well-defined and structured methodology that efficiently guides data professionals across all stages of a data mining project. The utilisation of a structured approach assists in reducing the likelihood of inadvertently neglecting crucial stages.

Flexibility: The framework demonstrates a notable level of versatility, allowing for its use in many data mining and machine learning applications, regardless of the particular domain or sector. The system possesses the capacity to effectively manage both simple and complex jobs.

CRISP-DM recognises the iterative nature of data mining activities. The aforementioned procedure facilitates the reassessment of preceding stages, the enhancement of models, and the incorporation of additional data during the duration of the project.

The process is characterised by its complete nature, since it incorporates all aspects of a project. This includes understanding the business objectives, gathering and preparing data, conducting the modelling phase, evaluating the results, and ultimately implementing the final deployment. The all-encompassing character of this method enables the recognition and resolution of potential challenges that may emerge at different phases.

The CRISP-DM methodology places considerable emphasis on the understanding of the organisational context and objectives. This practise ensures that data mining efforts are successfully aligned with concrete business needs.

Effective collaboration among team members and stakeholders is contingent upon the presence of clear communication. This facilitates the sharing of ideas and information between domain experts, data scientists, and decision-makers, hence generating a conducive environment for cooperation.

The implementation of CRISP-DM enables improved risk evaluation and mitigation by segmenting the project into several phases. The identification and timely mitigation of project risks are fundamental components of the project management process.

The promotion of reproducibility entails the adoption of comprehensive documentation and transparency, hence permitting the replication of research efforts by other scholars and enabling the verification of produced results.

The CRISP-DM methodology has exhibited a track record of achievement and has gained extensive acceptance in both academic and industry contexts. Many enterprises rely on this framework as a reliable tool for their data mining efforts.

The CRISP-DM technique provides a systematic and adaptable framework for conducting data mining and machine learning activities, empowering organisations to derive significant insights and make well-informed decisions using data.

3.1 System Requirement

3.1.1 Hardware

The hardware involved in this project is computer and android mobile device. A computer issued for the process of 3D visualization and segmentation from MRI and CT datasets to obtain the 3D model objects, then it also used for applying AR technology on the 3D model objects. A mobile device is used for testing and deploying this AR application in learning human anatomy.

Table 3.1 Specifications of laptop

Description	Specifications
Model	Predictive model about RSI indicator in predicting forex

	prices .
Processor	Intel Core i5-7200U
Operating System	Windows 10
Graphic	NVIDIA GeForce GT 930MX 2GB DDR3
Memory	4GB DDR4 RAM
Storage	1TB SATA HDD

Table 1 shows the software requirements of my model

3.2 Dashboard Design Interface

Figure 3.3 Shows the dashboard interface using shinyapps .

Utilizing a dashboard interface within a web application form is a convenient and user-centric approach to leverage machine learning techniques for the purposes of data analysis and decision-making. Utilizing platforms such as www.shinyapps.com for model distribution can significantly improve the accessibility and extend the reach of your model.

Within this particular configuration, individuals have the capability to engage with the machine learning model via a dashboard that is accessible on the web. This interface enables users to provide input data, adjust model parameters, and observe outcomes in a visual manner, all without necessitating the requirement for coding proficiency or an extensive comprehension of machine learning methods. The primary characteristics and advantages encompass:

User-Friendly Interface: The online application offers an interface that is intuitive and easy to use, hence enhancing the accessibility of machine learning to a wider range of users, including individuals who may not possess technical expertise.

Interactivity is a key feature of the paradigm, as it allows users to actively participate by manipulating parameters, examining data, and observing real-time visualizations. This facilitates the development of a more profound comprehension of the data and the behaviour of the model.

Customization is a feature that enables users to personalize the dashboard according to their individual use cases. This functionality allows users to concentrate on the most pertinent portions of the data and analysis. The process of customization guarantees that the model is tailored to meet the specific requirements of a certain firm.

Real-time insights can be obtained by users through the input of new data into the online form, enabling prompt decision-making.

The deployment of the dashboard on www.shinyapps.com guarantees convenient accessibility from any location with an internet connection. The utilization of this technology obviates the necessity for users to locally install software.

The platform generally incorporates security measures to safeguard sensitive data and model parameters, so preserving the confidentiality of data and adherence to regulatory requirements.

Scalability is a key attribute of web-based applications, as they possess the capability to effectively accommodate a significant volume of users concurrently. This characteristic renders such systems well-suited for deployment in several contexts, ranging from small teams to enterprise-level environments.

The feedback loop allows users to offer input and insights immediately within the interface, so enabling the iterative process of enhancing and refining the model over time.

In general, the development of a machine learning model accompanied by a web application dashboard, utilizing platforms such as www.shinyapps.com, facilitates the democratization of data science. This enables a broader range of individuals within businesses and external entities to access and utilize advanced analytics in a practical manner.

CHAPTER 4

Preliminary Work

Preliminary work about my system has been carried out accordingly by using CRISP-DM Approach . There are a total of 6 phases used in the process as stated below :

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modelling
5. Evaluation
6. Deployment

4.1.1 Software

1.Google Collab Cloud
2. Jupiter Notebook
3. R Studio
4. Flask

4.1.2 Business Understanding

Machine learning is not a strategy that is guaranteed to be successful, but it is one that is quite effective in forex trading. It is able to conduct accurate price forecasts in real time, simplify the process of buying and selling, and reduce the risks associated with human trading. In addition to this, it can assist you in enhancing your market surveillance and maintaining consistency throughout several trading sessions. Using ML in forex trading is dependent on this particular facet. Because of this factor, more and more people are doing it. The technique of constructing a programme using ML is straightforward. There is no room for error on the side of humans, and the software may be adjusted to meet the needs of any given trader.

- Remove human bias towards trading

- Help make informed for trading -To find trends and forecast future currency values, machine learning can evaluate a lot of past data. Businesses can use this information to improve trading choices and possibly boost earnings.
- Reduce risk: Predicting forex prices with machine learning can also help businesses to reduce risk. By identifying potential price movements in advance, businesses can take steps to mitigate risk and protect their investments.

In terms of data mining , the below are the data mining goals for my project :
Build predictive model with 75% accuracy
Build predictive model with 75% recall
Build predictive model with 75% precision

4.2 Data Understanding

Dataset was downloaded from Kaggle as a csv file , the dataset contains a total of 174719 records .As default it contains a total 7 columns , and every record is the 15 minute display of Open , High , Low and Close values of the price , additionally it also has computed values such as Volume and RSI . According to date , it records information from 1/1/2010 3:30 up till 12/30/2016 21:45 .

	Time	Open	High	Low	Close	Volume	RSI
0	1/1/2010 3:30	1.43280	1.43302	1.43219	1.43276	8.983000e+08	46.520325
1	1/1/2010 3:45	1.43284	1.43293	1.43181	1.43209	7.218000e+08	30.931759
2	1/1/2010 4:00	1.43218	1.43302	1.43182	1.43273	8.282000e+08	48.637055
3	1/1/2010 4:15	1.43263	1.43294	1.43204	1.43223	6.486000e+08	40.008276
4	1/1/2010 4:30	1.43239	1.43302	1.43197	1.43282	5.849000e+08	51.045117
5	1/1/2010 4:45	1.43284	1.43295	1.43220	1.43236	5.635000e+08	44.215198
6	1/1/2010 5:00	1.43237	1.43294	1.43199	1.43268	6.168000e+08	49.297575
7	1/1/2010 5:15	1.43275	1.43294	1.43211	1.43277	7.279000e+08	50.659130
8	1/1/2010 5:30	1.43214	1.43298	1.43210	1.43268	5.939000e+08	49.235270
9	1/1/2010 5:45	1.43253	1.43306	1.43214	1.43303	6.993000e+08	54.581554
10	1/1/2010 6:00	1.43283	1.43335	1.43269	1.43334	6.817000e+08	58.727547
11	1/1/2010 6:15	1.43331	1.43332	1.43268	1.43326	7.087000e+08	57.274530
12	1/1/2010 6:30	1.43322	1.43333	1.43270	1.43315	9.255000e+08	55.250340
13	1/1/2010 6:45	1.43318	1.43331	1.43269	1.43286	8.124000e+08	50.212002
14	1/1/2010 7:00	1.43328	1.43342	1.43226	1.43304	9.123000e+08	53.072483
15	1/1/2010 7:15	1.43303	1.43334	1.43254	1.43316	9.352000e+08	54.931490
16	1/1/2010 7:30	1.43329	1.43344	1.43307	1.43328	9.741000e+08	56.775520

Figure 4.1 Shows the data frame of EUR/USD

Dataset was then further studied by calling .info(), .describe to understand features within dataset .

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174720 entries, 0 to 174719
Data columns (total 7 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    174720 non-null  object
 1   Open    174720 non-null  float64
 2   High    174720 non-null  float64
 3   Low     174720 non-null  float64
 4   Close   174720 non-null  float64
 5   Volume  174720 non-null  float64
 6   RSI     174720 non-null  float64
dtypes: float64(6), object(1)
memory usage: 9.3+ MB
```

Figure 4.2 Shows the datatype of each column

4.2.1 Data Preprocessing

Preprocessing was carried on to make sure all the data which are required is suitable and appropriate for modelling . As a common practice , missing values were checked to make sure there are no null values .

```
In [6]:  #Check 0 values

         data.isna().sum()

Out[6]:  Time      0
         Open      0
         High      0
         Low       0
         Close     0
         Volume    0
         RSI       0
         dtype: int64
```

Figure 4.3 Shows the results of null values in each column

In [7]:
```
data.describe()
```

Out[7]:

	Open	High	Low	Close	Volume	RSI
count	174720.000000	174720.000000	174720.000000	174720.000000	1.747200e+05	174720.000000
mean	1.268402	1.268916	1.267884	1.268401	2.281760e+09	49.830342
std	0.112896	0.112929	0.112855	0.112897	2.462833e+09	11.662201
min	1.035580	1.036280	1.035230	1.035600	1.500000e+05	7.673118
25%	1.135410	1.135860	1.134930	1.135410	8.853200e+08	42.101629
50%	1.302470	1.303010	1.301990	1.302470	1.589425e+09	49.898374
75%	1.356683	1.357143	1.356240	1.356690	2.922002e+09	57.608452
max	1.493230	1.493980	1.491850	1.493240	3.779797e+10	93.633547

Figure 4.4 Shows the overall info and statistics of each feature in the columns .

Based on the default dataset , the values for the Volume column were too big and requires normalization . Hence , it was normalized.

In [8]:
```
#dropping zero values

df = data.loc[data["Volume"] != 0]
df.tail()
```

Out[8]:

	Time	Open	High	Low	Close	Volume	RSI
174715	12/30/2016 20:45	1.05358	1.05376	1.05238	1.05247	1.911700e+09	41.953596
174716	12/30/2016 21:00	1.05244	1.05251	1.05179	1.05180	8.912800e+08	38.666813
174717	12/30/2016 21:15	1.05179	1.05204	1.05141	1.05191	8.146200e+08	39.504777
174718	12/30/2016 21:30	1.05191	1.05193	1.05140	1.05156	4.275700e+08	37.738050
174719	12/30/2016 21:45	1.05150	1.05303	1.05123	1.05150	1.419200e+09	37.429023

In [9]:
```
df["Volume"] = df["Volume"].astype("float") / 1e9
df
```

Out[9]:

	Time	Open	High	Low	Close	Volume	RSI
0	1/1/2010 3:30	1.43280	1.43302	1.43219	1.43276	0.89830	46.520325
1	1/1/2010 3:45	1.43284	1.43293	1.43181	1.43209	0.72180	30.931759
2	1/1/2010 4:00	1.43218	1.43302	1.43182	1.43273	0.82820	48.637055
3	1/1/2010 4:15	1.43263	1.43294	1.43204	1.43223	0.64860	40.008276
4	1/1/2010 4:30	1.43239	1.43302	1.43197	1.43282	0.58490	51.045117

Figure 4.5 Shows the normalization of Volume.

Next , every data type was checked and made sure it is appropriate for modelling and visualizing purposes.

```
In [10]:  df["Time"] = pd.to_datetime(df["Time"])
          df
```

Out[10]:

	Time	Open	High	Low	Close	Volume	RSI
0	2010-01-01 03:30:00	1.43280	1.43302	1.43219	1.43276	0.89830	46.520325
1	2010-01-01 03:45:00	1.43284	1.43293	1.43181	1.43209	0.72180	30.931759
2	2010-01-01 04:00:00	1.43218	1.43302	1.43182	1.43273	0.82820	48.637055
3	2010-01-01 04:15:00	1.43263	1.43294	1.43204	1.43223	0.64860	40.008276
4	2010-01-01 04:30:00	1.43239	1.43302	1.43197	1.43282	0.58490	51.045117
...	...	...	...	...	...	...	...
174715	2016-12-30 20:45:00	1.05358	1.05376	1.05238	1.05247	1.91170	41.953596
174716	2016-12-30 21:00:00	1.05244	1.05251	1.05179	1.05180	0.89128	38.666813
174717	2016-12-30 21:15:00	1.05179	1.05204	1.05141	1.05191	0.81462	39.504777
174718	2016-12-30 21:30:00	1.05191	1.05193	1.05140	1.05156	0.42757	37.738050
174719	2016-12-30 21:45:00	1.05150	1.05303	1.05123	1.05150	1.41920	37.429023

174720 rows × 7 columns

Figure 4.6 Shows the conversion of object datatype to datetime format .

4.2.1 Data Visualization

Now we proceed into visualization the dataset that we are working on .

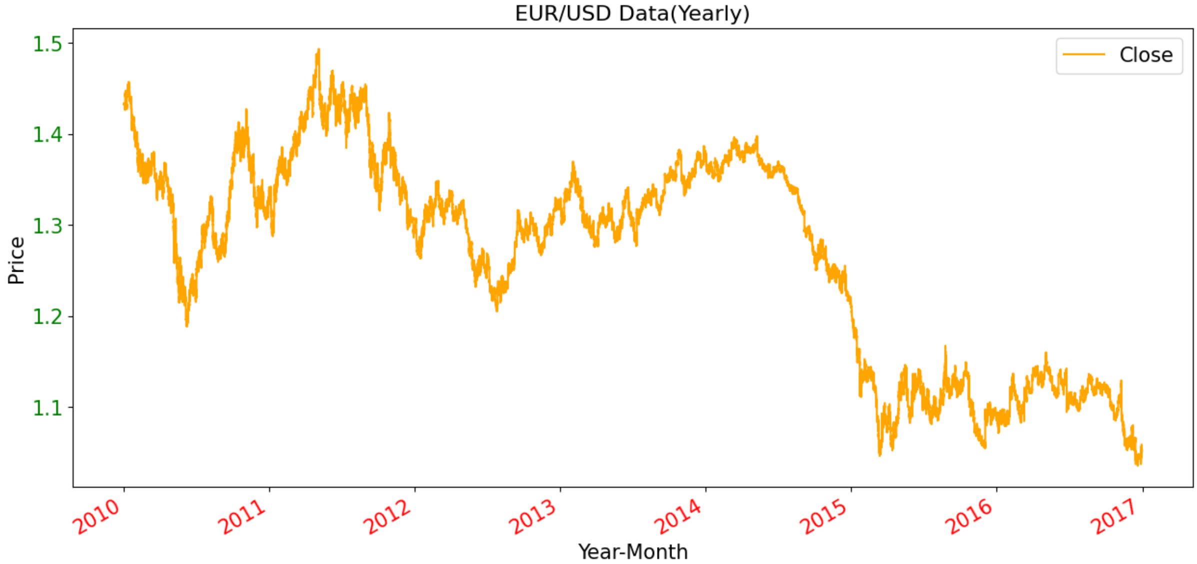


Figure 4.7 Shows the overview of the dataset for OHLC values against time .

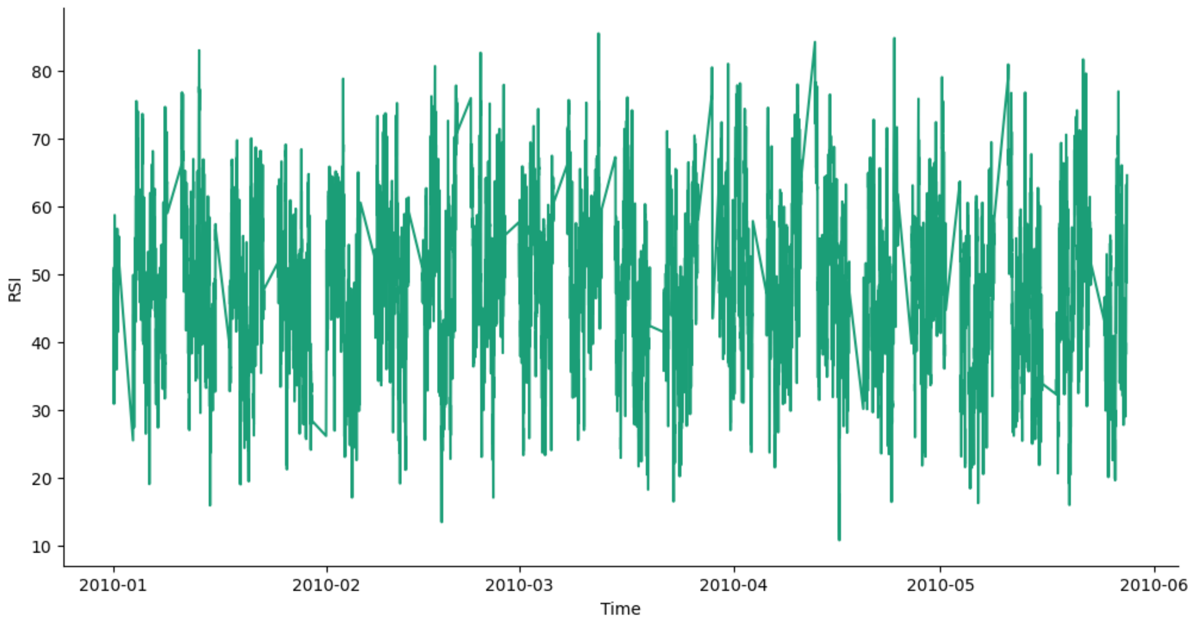


Figure 4.8 Shows the RSI values being plotted against time .

The above figure shows the volatility of RSI and fluctuates constantly.

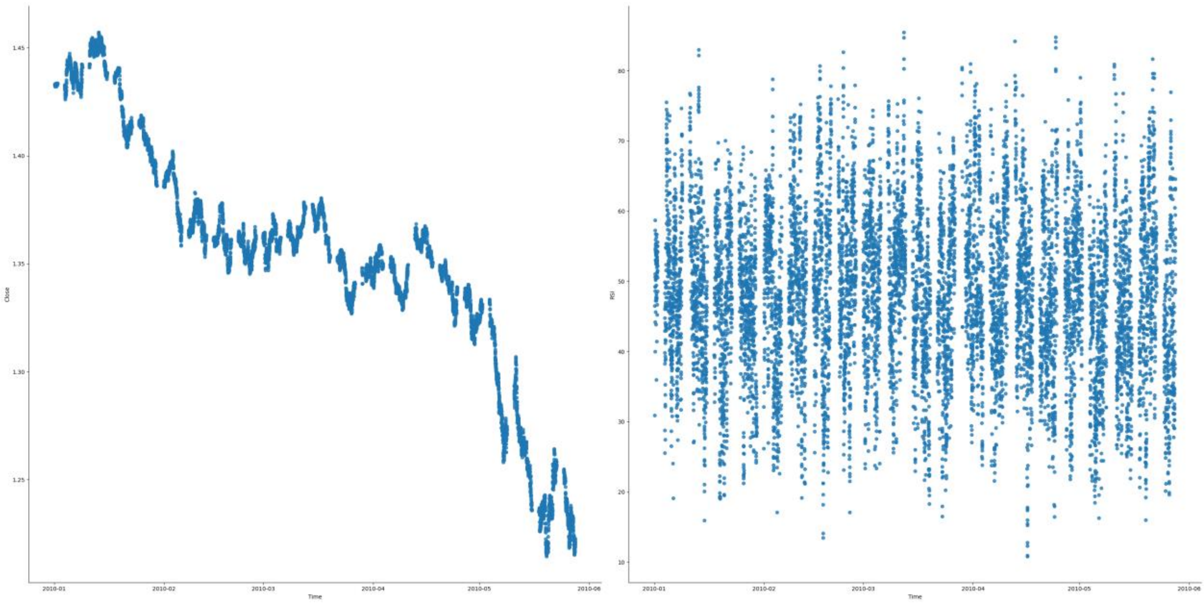


Figure 4.9 Shows the RSI oscillation and Price comparison .

Upon figure 4.9 , it shows of how RSI is much volatile compared to Price action .

However , if we take a look in a smaller timeframe , the RSI values are actually closely following price as a lagging indicator .

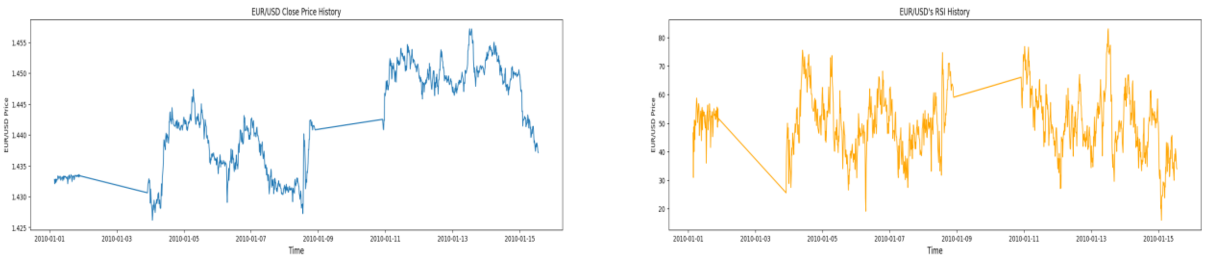


Figure 4.10 Shows the RSI values being plotted against time .

Now we can look out for Outliers .

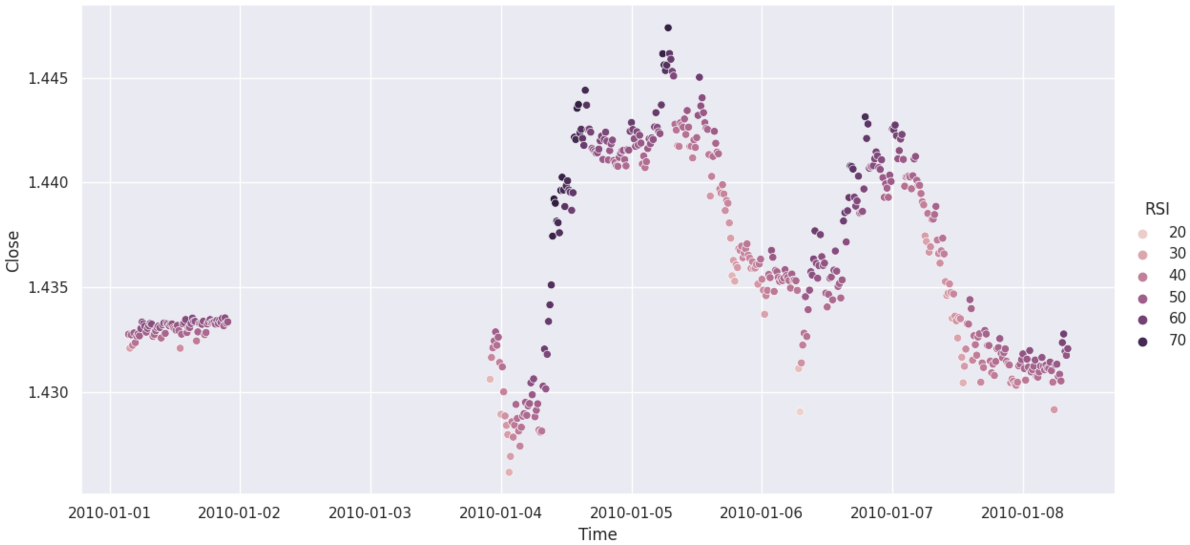


Figure 4.11 Visualizing outliers for RSI .

```
[179] # Remove the outliers from the dataset
      df = data.drop(outliers.index)
```

Figure 4.12 Visualizing outliers for RSI .

After ensuring that the data frame is now ready for preparation, the newest data is saved to a new csv file .

4.3.Data Preparation

In data preparation , the data frame is now prepared which will be finalized in this process for training and testing purposes in later stages .

Further, preprocessing is done for data columns .

```
In [41]:  data["Volume"]=data["Volume"].astype("string")
          data["Volume"].str.split(".")
          data.Volume

Out[41]: 0         0.8982999973
         1         0.7217999992
         2          0.828199995
         3         0.6486000004
         4         0.5849000015
                       ...
         174715     1.911700006
         174716    0.8912799988
         174717     0.814619997
         174718    0.4275700026
         174719     1.419199978
         Name: Volume, Length: 174720, dtype: string

In [42]:  data["Time"] = pd.to_datetime(data["Time"])
```

Figure 4.13 Conversion of data types of Volume and Time .

Feature Engineering

A new column was added to calculate the difference of Close prices .

```
In [45]:  #normalization for adjacent close

          # Normalize aclose value
          # We use this value to train model

          data['difference'] = data['Close'] - data['Close'].shift(1)
          diff_range = data['difference'].max() - data['difference'].min()
          data['difference'] = data['difference'] / diff_range

In [46]:  data

Out[46]:
```

	Time	Open	High	Low	Close	Volume	RSI	difference
0	2010-01-01 03:30:00	1.43280	1.43302	1.43219	1.43276	0.8982999973	46.520325	NaN
1	2010-01-01 03:45:00	1.43284	1.43293	1.43181	1.43209	0.7217999992	30.931759	-0.016975
2	2010-01-01 04:00:00	1.43218	1.43302	1.43182	1.43273	0.828199995	48.637055	0.016215
3	2010-01-01 04:15:00	1.43263	1.43294	1.43204	1.43223	0.6486000004	40.008276	-0.012668
4	2010-01-01 04:30:00	1.43239	1.43302	1.43197	1.43282	0.5849000015	51.045117	0.014948
...	...	...	...	...	...	...	...	...

Figure 4.14 Adding of new column called 'difference'

To normalize and also to categorize Close values that are connected towards yesterday Close price , I have also included a new column that shows helps read price as of for Increase and decreasing based on their past day close price.

```
In [48]:  def price_movement(data):
              """
              This function takes a DataFrame of close prices and returns a new
              indicating whether the price has increased or decreased from the

              Args:
                data: A DataFrame of close prices.

              Returns:
                A DataFrame with a column indicating whether the price has incr
                the previous row.
              """

              data['Price Movement'] = data['Close'].diff()

              data['Price Movement'] = data['Price Movement'].apply(lambda x: '

              return data
```

Figure 4.15 Adding of new column called 'price_movement'

Visualizing newly added columns .

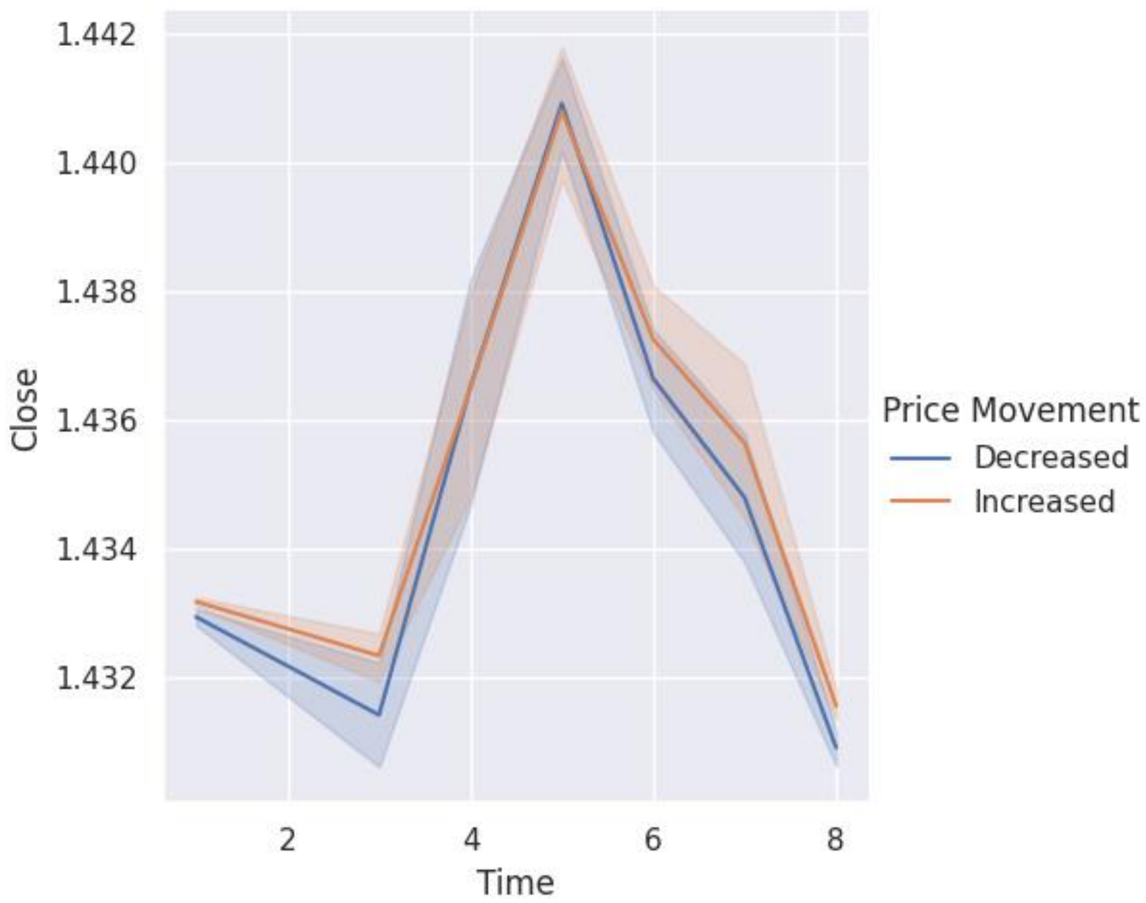


Figure 4.16 Price Movement easily illustrate Price action .

I also decided to make my data frame more readable and would possibly make data visualization way better by adding a new column which is called 'RSI_label' .This columns shows the reading of RSI (Relative Strength Index) in depicting the reading as 'Overbought' , 'Oversold' and neutral .If the RSI value is more than 70 , it is shown as Overbought , if lesser than 30 then its mapped as Oversold any value in between is stated as Neutral . Hence to map values into each category , a function was created and then applied into the new 'RSI_label' platform .

```
In [57]:  def RSI_label(df):
              if df['RSI'] > 60:
                  return 'Overbought'
              elif df['RSI'] > 30 and df['RSI'] <= 60:
                  return 'Neutral'
              elif df['RSI'] < 30 :
                  return 'Oversold'
              return 'NotDefined'
```

```
In [58]:  df['RSI_label'] = df.apply(lambda df: RSI_label(df
          df
```

Figure 4.17 Including RSI_label column to dataframe .

Rounding off values for RSI values .

```
In [60]:  df['RSI'] = df['RSI'].round(2)
          df

<ipython-input-60-a31ed20f17e6>:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[60]:

	Time	Open	High	Low	Close	Volume	RSI	difference	Move
1	2010-01-01 03:45:00	1.43284	1.43293	1.43181	1.43209	0.7217999992	30.93	-0.016975	Decre
2	2010-01-01 04:00:00	1.43218	1.43302	1.43182	1.43273	0.828199995	48.64	0.016215	Incre
3	2010-01-01	1.43263	1.43294	1.43204	1.43223	0.6486000004	40.01	-0.012668	Decre

Figure 4.18 Rounding off values .

Including smooth moving average[SMA20, SMA10] as a feature to smoothen price action due to fluctuations ,this includes the 20 day moving average and 10 day moving average .

Out[64]:

	Time	Open	High	Low	Close	Volume	RSI	difference	Price Movement	RSI_label	SMA_10	SMA_20
1	2010-01-01 03:45:00	1.43284	1.43293	1.43181	1.43209	0.7217999992	30.93	-0.016975	Decreased	Neutral	NaN	NaN
2	2010-01-01 04:00:00	1.43218	1.43302	1.43182	1.43273	0.828199995	48.64	0.016215	Increased	Neutral	NaN	NaN
3	2010-01-01 04:15:00	1.43263	1.43294	1.43204	1.43223	0.6486000004	40.01	-0.012668	Decreased	Neutral	NaN	NaN
4	2010-01-01 04:30:00	1.43239	1.43302	1.43197	1.43282	0.5849000015	51.05	0.014948	Increased	Neutral	NaN	NaN
5	2010-01-01 04:45:00	1.43284	1.43295	1.43220	1.43236	0.5635000038	44.22	-0.011654	Decreased	Neutral	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
174715	2016-12-30 20:45:00	1.05358	1.05376	1.05238	1.05247	1.911700006	41.95	-0.028883	Decreased	Neutral	1.052946	1.053744
174716	2016-12-30 21:00:00	1.05244	1.05251	1.05179	1.05180	0.8912799988	38.67	-0.016975	Decreased	Neutral	1.052819	1.053549
174717	2016-12-30 21:15:00	1.05179	1.05204	1.05141	1.05191	0.814619997	39.50	0.002787	Increased	Neutral	1.052739	1.053362
174718	2016-12-30 21:30:00	1.05191	1.05193	1.05140	1.05156	0.4275700026	37.74	-0.008867	Decreased	Neutral	1.052631	1.053154
174719	2016-12-30 21:45:00	1.05150	1.05303	1.05123	1.05150	1.419199978	37.43	-0.001520	Decreased	Neutral	1.052531	1.052970

174719 rows × 12 columns

Figure 4.19 SMA_10 and SMA20 being included to dataframe .

Upon visualizing , SMA20 was chosen as the best Moving average that follow prices actions.

Figure 4.20 20-day moving average able to capture the trend of price .

Using seasonal decompose , which is a well known time series analysis to check for 20 day Moving average trend .

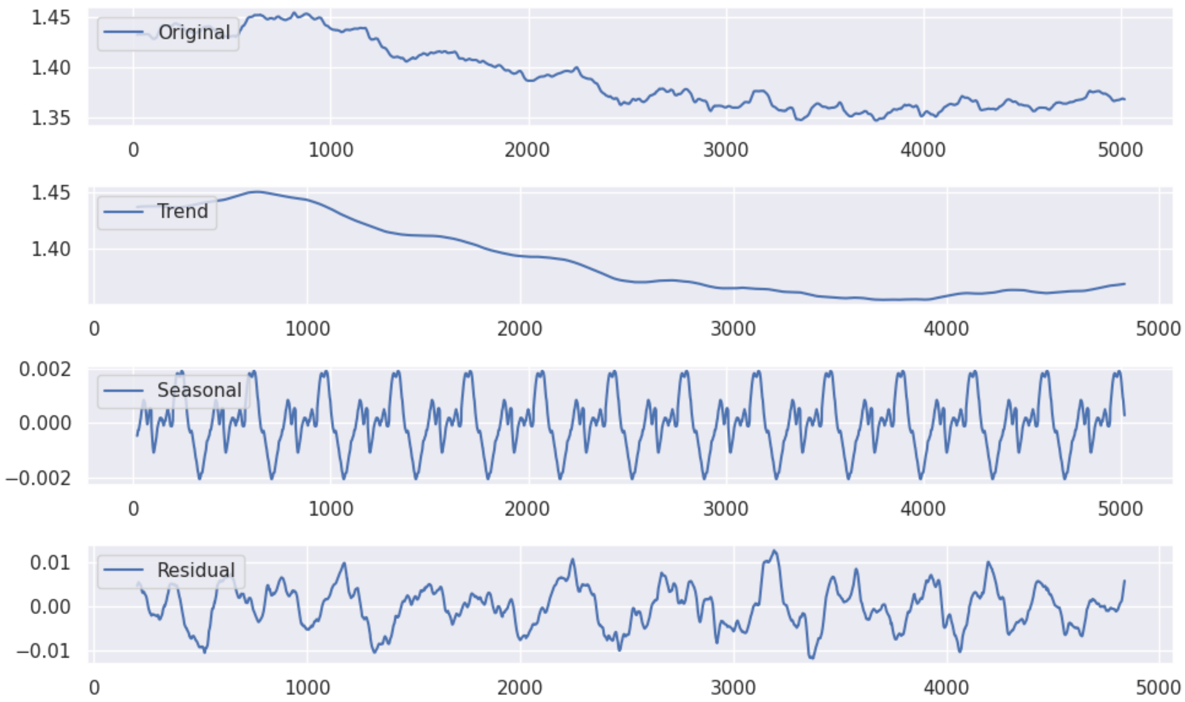


Figure 4.21 Seasonal decomposition of 20 day moving average.

Using seasonal decompose , which is a well known time series analysis to check for RSI.

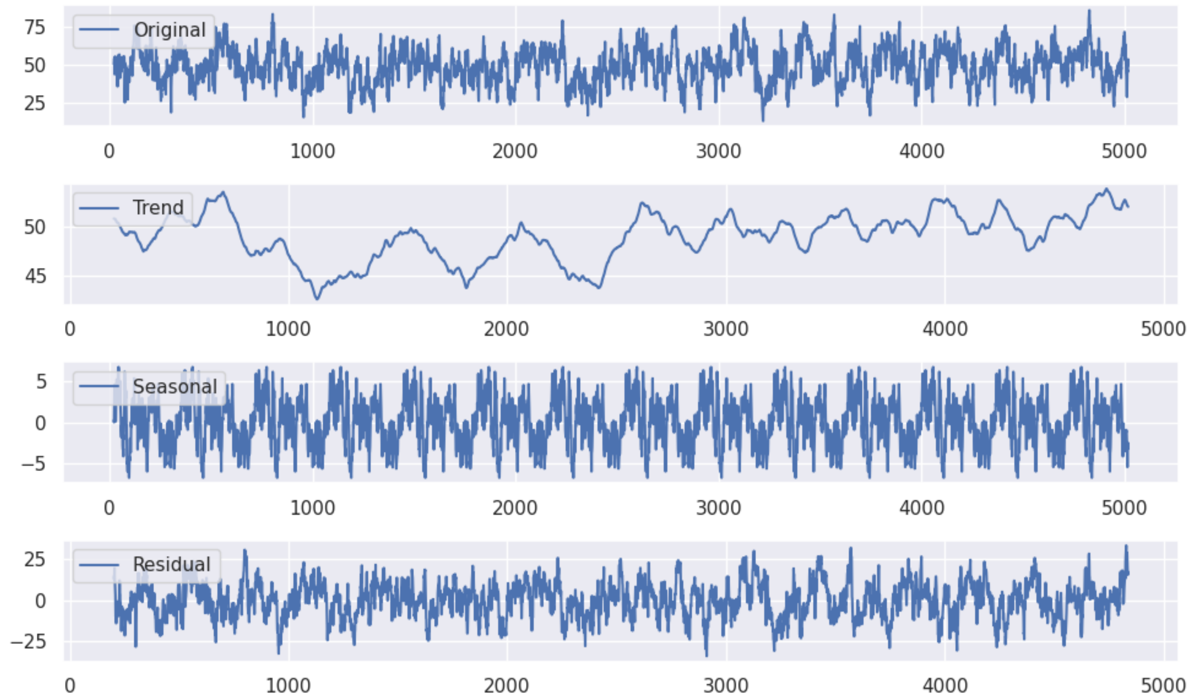


Figure 4.22 Seasonal decomposition of RSI .

4.4.Data Understanding (again)

Data Understanding is now again used after pre-processing data , but this time , graphical data is used to present and study the co-relevance of each feature towards target approach .

```
In [106]:  df.corr(method='pearson')
```

```
<ipython-input-106-432dd9d4238b>:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will def
y valid columns or specify the value of numeric_only to silence this warning.
```

Out[106]:

	Open	High	Low	Close	RSI	difference	SMA_10	SMA_20
Open	1.000000	0.999985	0.999985	0.999976	0.024309	-0.002667	0.999950	0.999872
High	0.999985	1.000000	0.999974	0.999987	0.025649	0.001030	0.999936	0.999857
Low	0.999985	0.999974	1.000000	0.999987	0.025727	0.000916	0.999936	0.999856
Close	0.999976	0.999987	0.999987	1.000000	0.026630	0.004140	0.999931	0.999851
RSI	0.024309	0.025649	0.025727	0.026630	1.000000	0.338723	0.018137	0.012052
difference	-0.002667	0.001030	0.000916	0.004140	0.338723	1.000000	-0.002104	-0.002452
SMA_10	0.999950	0.999936	0.999936	0.999931	0.018137	-0.002104	1.000000	0.999960
SMA_20	0.999872	0.999857	0.999856	0.999851	0.012052	-0.002452	0.999960	1.000000

Figure 4.23 Shows the information about latest data frame and their correlevance.

Line Charts were used to see the relevance of Close price of EUR/USD towards RSI(predicting value).

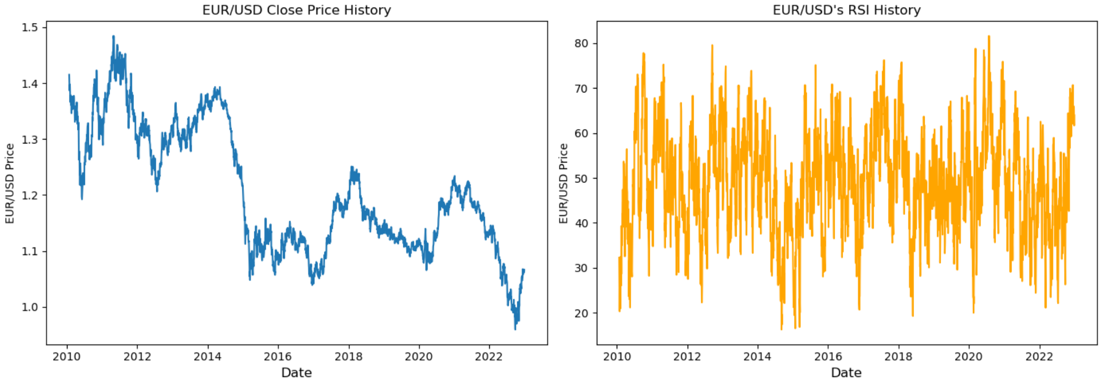


Figure 4.24 Shows the visualization of EUR/USD .

Here it appears to shows that Close price appears to fall , but RSI seems to have more Volatility despite the falling price , but towards the end in year 2022 , it can be seemed to be following price pattern quite precisely.

However , a deeper look was taken to get a better understanding of these 2 fields . Using a smaller timeframe . of first 40 days .

Figure 4.25 Shows how RSI is affected by price movement and used for trading.

Based on the figure above , it can be seen that , using a golden rule of doing the opposite of what retail traders are mostly doing .We can see as RSI reached below 30 reading, it falls under the oversold category . Hence , if price is know to be oversold , one would expect price to go up the next day in this context . As shown in the figure , the price did crawl its way up slightly to show that RSI shows relevance towards following price action and its useful to help traders make their next move.

If we were to see a take a smaller timeframe and plot the graphs in the same axis , we can see that RSI cools down as Price oscillates simultaneously .

4.5.1Modelling (Model 1)

Under modelling , the technique of stacking , which I will be first preparing a model where it predicts the target variable of “RSI ”using Random Forest Regression . Then , once I have the predicted values , I will be creating a new model which then Predicts the “Close” value of with predicted RSI values from previous model . Target variable for model 1 is “RSI”.

The RSI values were shifted so that previous day RSI is used to predict following day RSI .

```
data
```

	Time	Open	High	Low	Close	Volume	RSI	difference	Price Movement	RSI_label	SMA_10	SMA_20
0	2010-01-01 08:30:00	1.43264	1.43333	1.43221	1.43293	1.13100	50.45	0.004560	Increased	Neutral	1.433033	1.432853
1	2010-01-01 08:45:00	1.43292	1.43335	1.43251	1.43308	1.21080	53.00	0.003800	Increased	Neutral	1.433015	1.432902
2	2010-01-01 09:00:00	1.43325	1.43343	1.43272	1.43315	1.16870	54.19	0.001773	Increased	Neutral	1.433015	1.432924
3	2010-01-01 09:15:00	1.43323	1.43332	1.43270	1.43308	1.35320	52.76	-0.001773	Decreased	Neutral	1.433037	1.432966
4	2010-01-01 09:30:00	1.43324	1.43334	1.43249	1.43258	1.13630	43.84	-0.012668	Decreased	Neutral	1.432991	1.432954
...	...	...	...	...	...	...	...	...	...	...	...	...
174695	2016-12-30 20:45:00	1.05358	1.05376	1.05238	1.05247	1.91170	41.95	-0.028883	Decreased	Neutral	1.052946	1.053744
174696	2016-12-30 21:00:00	1.05244	1.05251	1.05179	1.05180	0.89128	38.67	-0.016975	Decreased	Neutral	1.052819	1.053548
174697	2016-12-30 21:15:00	1.05179	1.05204	1.05141	1.05191	0.81462	39.50	0.002787	Increased	Neutral	1.052739	1.053362
174698	2016-12-30 21:30:00	1.05191	1.05193	1.05140	1.05156	0.42757	37.74	-0.008867	Decreased	Neutral	1.052631	1.053154
174699	2016-12-30 21:45:00	1.05150	1.05303	1.05123	1.05150	1.41920	37.43	-0.001520	Decreased	Neutral	1.052531	1.052969

174700 rows × 12 columns

```
[157] data['RSI'] = data['RSI'].shift(-1)
      data['RSI_label'] = data['RSI_label'].shift(-1)
      data
```

	Time	Open	High	Low	Close	Volume	RSI	difference	Price Movement	RSI_label	SMA_10	SMA_20
0	2010-01-01 08:30:00	1.43264	1.43333	1.43221	1.43293	1.13100	53.00	0.004560	Increased	Neutral	1.433033	1.432853
		1.43292	1.43335	1.43251	1.43308	1.21080	54.19	0.003800	Increased	Neutral	1.433015	1.432902

Figure 4.26 show the comparison of shifted values and original dataframe.

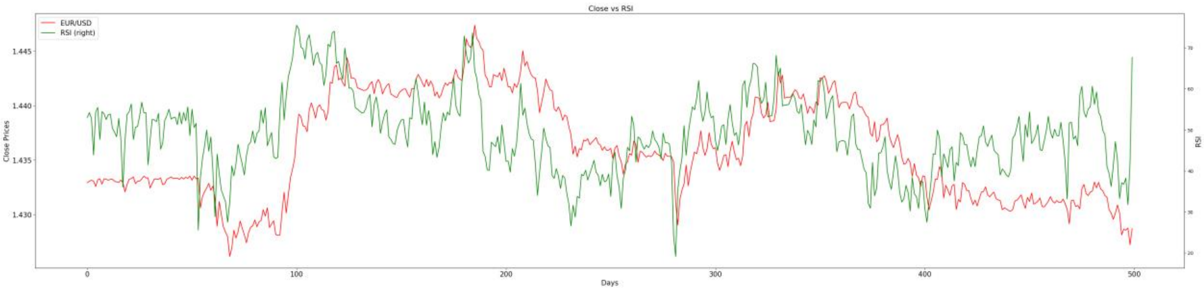


Figure 4.27 Visualizing the lagged indicator .

Before splitting , it is good to check the relevance of the features to target variable .

```
# Afterwards, it is recommended to revisit the correlation matrix analysis:
corr_matrix = data.corr(numeric_only=True)
corr_matrix["Close"].sort_values(ascending=False)

Close         1.000000
High          0.999987
Low           0.999987
Open          0.999976
SMA_10        0.999931
SMA_20        0.999851
RSI           0.024259
difference    0.004139
Volume       -0.232279
Name: Close, dtype: float64
```

The attribute Time was popped out from the dataframe for easier visualizing purposes later on .

```
date_time=data.pop('Time')
date_time
```

```
0         2010-01-01 08:30:00
1         2010-01-01 08:45:00
2         2010-01-01 09:00:00
3         2010-01-01 09:15:00
4         2010-01-01 09:30:00
                  ...
174694    2016-12-30 20:30:00
174695    2016-12-30 20:45:00
174696    2016-12-30 21:00:00
174697    2016-12-30 21:15:00
174698    2016-12-30 21:30:00
Name: Time, Length: 174699, dtype: datetime64[ns]
```

Figure 4.28 Popping of datetime .

Splitting data for training and testing for modelling .

```
# Extract features and target variable
X = data[['Close', 'Open', 'High','Low','difference','SMA_20','RSI_label']]  # Add other features as needed
y = data['RSI']
```

```
[181] from sklearn.preprocessing import MinMaxScaler

      scaler = MinMaxScaler()
```

```
[182] X = scaler.fit_transform(X)
```

```
[183] # Split the dataset into training and testing sets
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Feature scaling was also implemented to normalize the values in dataframe.

Fitting dataframe in Linear Regresison model .

```
[187] from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error

      # Create and train the Linear Regression model
      lr_model = LinearRegression()
      lr_model.fit(X_train, y_train)

▾ LinearRegression
LinearRegression()
```

```
y_pred_lr = lr_model.predict(X_test)
```

Figure 4.29 Implementing Linear Regression algorithm to model for training .

Visualizing predicted values versus the actual values .

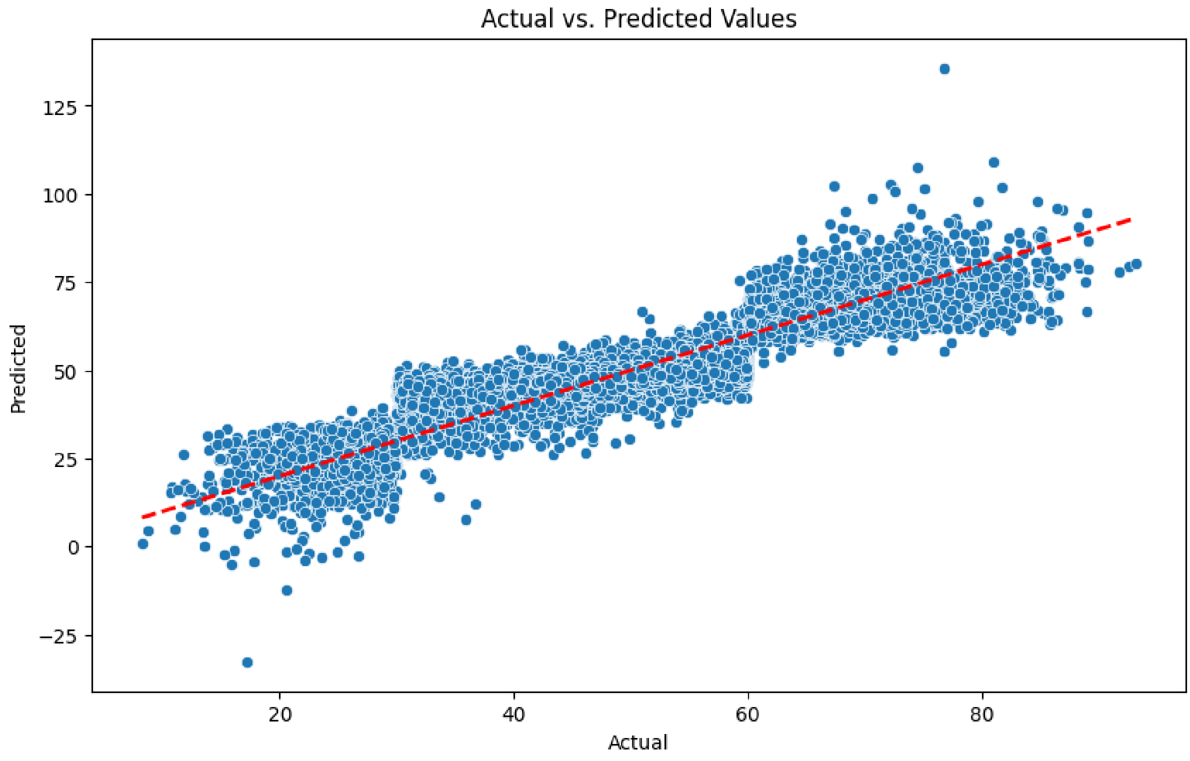


Figure 4.30 Actual vs Predicted values lineplot .

There are some outliers found but are very minimal in this case . RSI values are able to learn the trend of the RSI movement along with price .

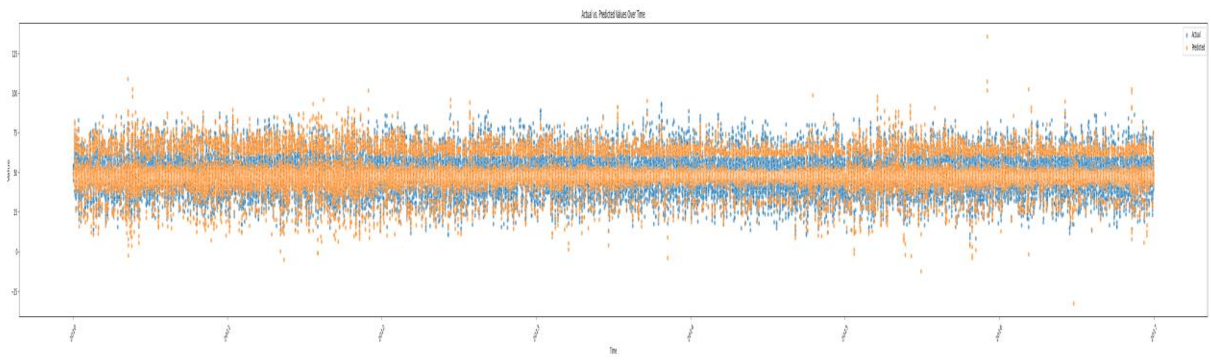


Figure 4.31 Actual vs Predicted against time .

The orange values shows the predicted values and the blue ones are the real values . Here is able to see the model being able to learn and to not overfit . Below is a closer look .

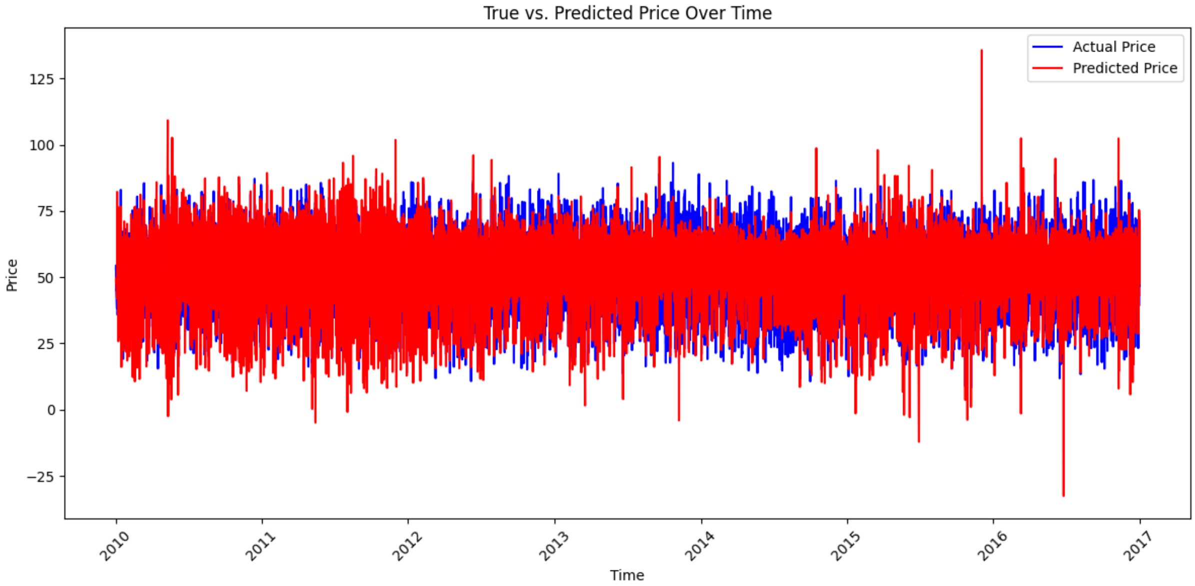


Figure 4.32 Actual vs Predicted against time(smaller timeframe) .

A new dataframe was created to compare and contrast the predicted values with the actual values .

results

	Actual	Predicted
154343	50.75	44.891674
32960	40.90	45.545797
126197	47.12	47.927421
98212	42.99	44.176118
9659	56.77	47.838751
...	...	...
85300	66.96	66.758276
157114	77.30	67.167687
144396	52.52	48.433883
171067	69.88	65.988492
14106	65.46	66.044831

34940 rows × 2 columns

Figure 4.33 Actual vs Predicted dataframe .

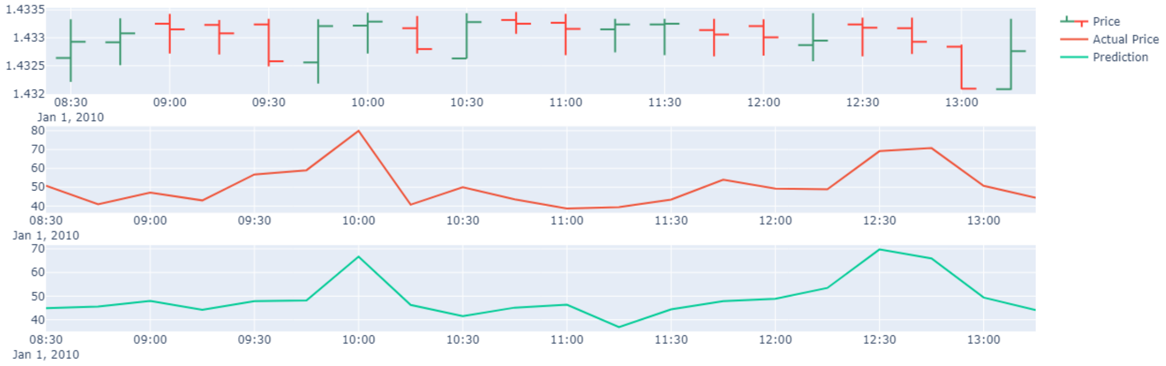


Figure 4.35 Actual vs Predicted against time(20 day timeframe) .

Now , we use Random Forest Regression to predict . The performance of the model is slightly better than Using linear regression .

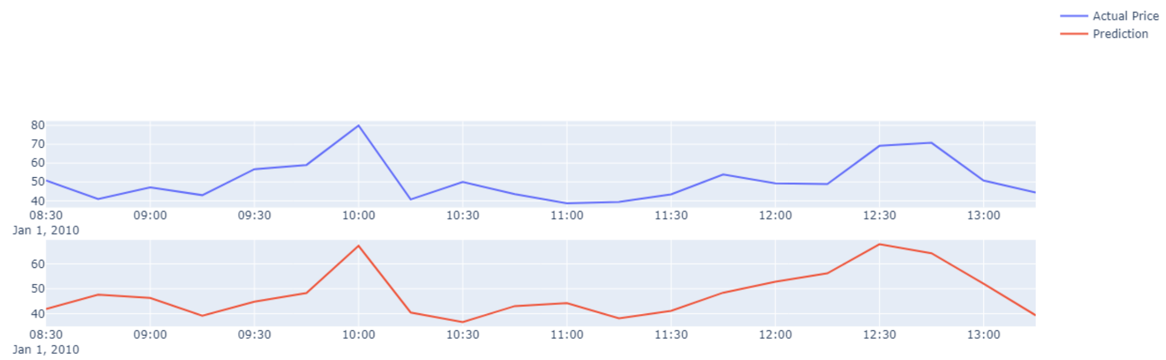


Figure 4.36 Actual vs Predicted against time(2 day timeframe) using Random Forest Regression .

A side by side comparison was then display to see then performance even better .

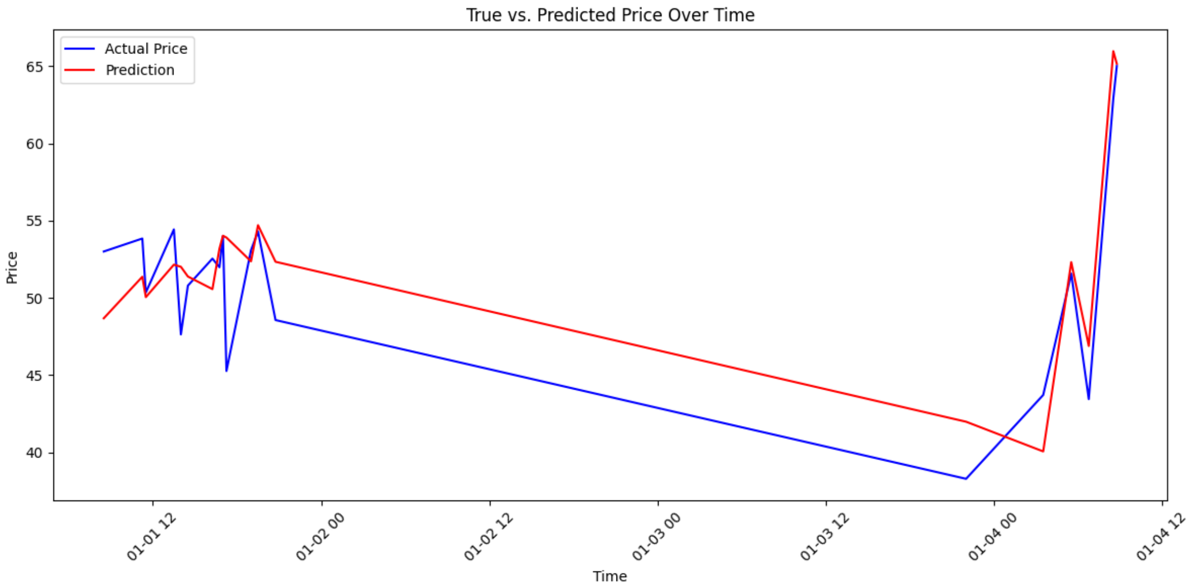


Figure 4.37 Actual vs Predicted against time(20 day timeframe) using Random Forest Regression .

Since Random Forest performed better, the results of the predicted values were saved to a new csv file .

```
# Calculate the mean squared error for logistic regression
mse_rf = metrics.mean_squared_error(y_test, y_pred_rf)
mae_rf = metrics.mean_absolute_error(y_test, y_pred_rf)
rmse = np.sqrt(mse_rf)
r2 = metrics.r2_score(y_test, y_pred_rf)
print("Mean Squared Error:", mse_rf)
print("Mean Absolute Error:", mae_rf)
print("Root Mean Squared Error:", rmse)
print("R-squared:", r2)

Mean Squared Error: 28.545213694008872
Mean Absolute Error: 4.219131886090441
Root Mean Squared Error: 5.342772098265924
R-squared: 0.7889570592355619

[210] data['Predicted Results'] = results_rf['Predicted_rf']

[225] new_df.to_csv('data_with_predictions.csv', index=False)
```

Figure 4.38 Predicted values were record and saved to a new csv file .

4.5.2 Modelling (Model 2)

Splitting of data for training and testing for model 2 .

```
[ ] # Extract features and target variable
    X = data[['P_RSI', 'Open', 'High','Low','difference','SMA_20']]  # Add other features as needed
    y = data['Close']
```

```
[ ] from sklearn.preprocessing import MinMaxScaler

    scaler = MinMaxScaler()
```

```
[ ] X = scaler.fit_transform(X)
```

```
[ ] # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 4.39 Actual vs Predicted against time(smaller timeframe) .

Predicting the model .

```
[ ] from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_absolute_error

    # Create and train the Linear Regression model
    lr_model = LinearRegression()
    lr_model.fit(X_train, y_train)

    ▾ LinearRegression
    LinearRegression()
```

z

```
[ ] y_pred_lr = lr_model.predict(X_test)
```

Figure 4.40 Shows the model being predicted .

```
[ ] results
```

	Actual	Predicted
3248	1.26480	1.264898
27114	1.12800	1.128028
16816	1.29317	1.293179
12538	1.25242	1.252412
3219	1.27052	1.270611
...	...	...
3634	1.36348	1.363371
6138	1.41216	1.412151
4808	1.31317	1.313259
3858	1.39568	1.395835
12085	1.24090	1.240957

6988 rows × 2 columns

Figure 4.41 Actual vs Predicted 'Close' values using Linear Regression model .

Using random forest regression to predict .

```
# Create a Random Forest Regression model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42) # Y

# Fit the model on the training data
rf_model.fit(X_train, y_train)

# Make predictions on the testing data
y_pred_rf = rf_model.predict(X_test)
```

Figure 4.42 Actual vs Predicted 'Close' values using Random Forest Regressor model .

CHAPTER 6

MODEL 1 Evaluation :

	Random Forest	Linear Regression
MSE	28.545213694008872	33.18640992522371
MAE	4.219131886090441	4.612107509801478
RSME	5.342772098265924	5.760764699692542
R2	0.7889570592355619	0.7546433661660304

Table 3.1 Shows the evaluation metrics of both algorithms for model 1 .

MODEL 2 Evaluation (Final):

	Random Forest	Linear Regression
MSE	1.1581627491128025e-07	1.4913476479457126e-08
MAE	0.0001762738694905612	3.794320037721796e-05
RSME	0.0003403179027193254	0.00012212074549173505
R2	0.9999908545075131	0.9999988223495601

Table 3.2 Shows the evaluation metrics of both algorithms for model 2 .

Once again, Random forest performed better than linear regression . Hence the Random forest regression algorithm serves as the best algorithm for the current model .

The results are as follows :

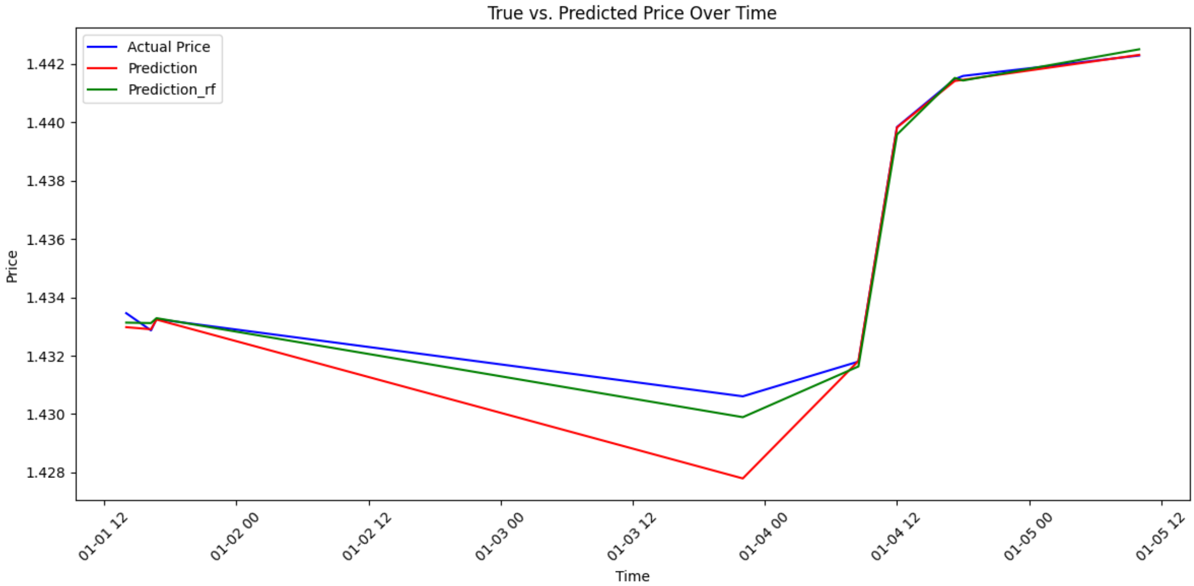


Figure 4.42 Results of 2 models and the actual price prediction vs time .

As the figure depicts see the Random Forest Regressor has outperformed Linear Regression and is used as the final predictive model .

CHAPTER 7

Conclusion

In conclusion , the project's core focus was to use 2 important indicator which is RSI and SMA to predict the future price of currencies . I am able to use both of the indicator as to my advantage to predict the close price of currencies by implementing the stacking method in machine learning . In terms of modelling , both Model 1 and Model 2 are used together and have given out good results in terms of MAE, MSE RSME and R2 score . However no system is perfect and with more time it can surely be further improvised accordingly.

This project stands out compared to other projects as technical indicators are normally given less importance towards trading forex prices in these days. However , through my project , it can be understood that technical indicators can be very helpful if used properly and using data mining and data understanding towards each aspect of the technical indicator helps to add predictive nature towards forex price forecasting rather than only using it as a tool .

Throughout the project , there were several complications , upon using the CRISP-DM approach , all process were well managed . However , more effort is required to increase my models performance overall and better algorithmic solutions can be implemented in search of better performing models .

Overall, my project which focuses on forex price that are compared towards Global Currency monitoring system can be said to be strongly affected by machine learning method in producing predictive results that helps to make trading better in a way .In contrast , predictive results used in this project are non financial advices . All the predictive results and techniques are only for educational purposes only. The projects requires more work to be accomplished where I can find more reliable results to be used for predicting RSI values and EMA values and producing a system where one can use it as a ticker tool with predictive results .

REFERENCES

[1]D. Bradfield, "The History of Forex," *DailyFX*, Dec. 19, 2018. https://www.dailyfx.com/education/beginner/history-of-forex.html

[2]"How Digitisation has Transformed the Forex Market," *How Digitisation has Transformed the Forex Market | FinTech Magazine*, Apr. 29, 2022. https://fintechmagazine.com/financial-services-finserv/how-digitisation-has-transformed-the-forex-market

[3]"XGBoost Documentation — xgboost 1.7.5 documentation," *XGBoost Documentation — xgboost 1.7.5 documentation*. https://xgboost.readthedocs.io/en/stable/

[4]OpenGeoHub, "Extrapolation is tough for trees (tree-based learners), combining learners of different type makes…," *Medium*, Jul. 17, 2021. https://medium.com/nerd-for-tech/extrapolation-is-tough-for-trees-tree-based-learners-combining-learners-of-different-type-makes-659187a6f58d

[5]"SPX | S&P 500 Index Overview | MarketWatch," *MarketWatch*, Apr. 22, 2023. https://www.marketwatch.com/investing/index/spx

[6]guest_blog, "Introduction to XGBoost Algorithm in Machine Learning," *Analytics Vidhya*, Sep. 06, 2018. https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

[7]G. L. Team, "Random forest Algorithm in Machine learning: An Overview," *Great Learning Blog: Free Resources what Matters to shape your Career!*, Feb. 19, 2020. https://www.mygreatlearning.com/blog/random-forest-algorithm/

[8]R. Python, "Linear Regression in Python – Real Python," *Linear Regression in Python – Real Python*. https://realpython.com/linear-regression-in-python/

[9]"Linear Regression (Python Implementation) - GeeksforGeeks," *GeeksforGeeks*, Mar. 19, 2017. https://www.geeksforgeeks.org/linear-regression-python-implementation/

[10]A. Wilson, "Stock Prediction Using Linear Regression," *Medium*, Dec. 09, 2020. https://medium.com/analytics-vidhya/stock-prediction-using-linear-regression-cd1d8351f536

[11]P. Soni, Y. Tewari, and D. Krishnan, “Machine Learning Approaches in Stock Price Prediction: A Systematic Review,” *Journal of Physics: Conference Series*, vol. 2161, no. 1, p. 012065, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012065.

[12]M. Trends, “Machine Learning in Forex Trading,” *Analytics Insight*, Jun. 22, 2021. https://www.analyticsinsight.net/machine-learning-in-forex-trading/

[13]“XGBoost for stock trend & prices prediction,” *XGBoost for stock trend & prices prediction | Kaggle*. https://www.kaggle.com/code/mtszkw/xgboost-for-stock-trend-prices-prediction

[14]C. Sun, Z. Chen, Y. Qin, and B. Wang, “Multi-step Time Series Forecasting Based on Informer-XGBoost-GA,” *Journal of Physics: Conference Series*, vol. 2333, no. 1, p. 012009, Aug. 2022, doi: 10.1088/1742-6596/2333/1/012009.

[15]“Pros And Cons Of Time Series Analysis 2023 - Ablison,” *Ablison*, Mar. 02, 2023. https://www.ablison.com/pros-and-cons-of-time-series-analysis/

[16]“Customize Dates on Time Series Plots in Python Using Matplotlib,” *Earth Data Science - Earth Lab*, Nov. 19, 2019. https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python/customize-dates-matplotlib-plots-python/

[17]T. Binhuraib, “An Introduction to Time Series Analysis with ARIMA,” *Medium*, Mar. 16, 2021. https://towardsdatascience.com/an-introduction-to-time-series-analysis-with-arima-a8b9c9a961fb

[18]“A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks,” *Built In*, Feb. 28, 2023. https://builtin.com/data-science/recurrent-neural-networks-and-lstm

[19]“S&P 500 || Time Series Forecasting with Prophet,” *S&P 500 || Time Series Forecasting with Prophet | Kaggle*. https://www.kaggle.com/code/janiobachmann/s-p-500-time-series-forecasting-with-prophet#Time-Series-Forecast-with-Prophet

[20]“Relative Strength Index (RSI) Indicator Explained With Formula,” *Investopedia*, Mar. 31, 2023. https://www.investopedia.com/terms/r/rsi.asp

[21]“What are Neural Networks? | IBM,” *What are Neural Networks? | IBM*. https://www.ibm.com/topics/neural-networks

[22]A. Roy, “Stock Price Prediction Based on Deep Learning,” *Medium*, Jan. 09, 2021. https://towardsdatascience.com/stock-price-prediction-based-on-deep-learning-3842ef697da0

[23]abr-98, “GitHub - abr-98/Stock_prediction_hybrid_model: A hybrid of ANN, RNN and Regressor models to predict stock prices,” *GitHub*, Jul. 26, 2020. https://github.com/abr-98/Stock_prediction_hybrid_model

[24]N. Hotz, “What is CRISP DM? - Data Science Process Alliance,” *Data Science Process Alliance*, Sep. 10, 2018. https://www.datascience-pm.com/crisp-dm-2/

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3,2023	Study week no.:3
Student Name & ID:2006649	
Supervisor: Dr Abdulkarim Janaan a/l Jebna	
Project Title: Global Currency Monitoring System	

1. WORK DONE Successful searched for a bigger dataset that contain more data and proper volume and RSI information . Researched about Time Series models and algorithms .
2. WORK TO BE DONE Prapare data frame for modeling . Normalize values .
3. PROBLEMS ENCOUNTERED Choosing the correct algorithm to predict , was confused on having many options of algorithms to be used .

4. SELF EVALUATION OF THE PROGRESS Needed more progress on my side .

Satish

Supervisor’s signature	Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3,2023	**Study week no.:5**
Student Name & ID:2006649	
Supervisor: Dr Abdulkarim Janaan a/l Jebna	
Project Title: Global Currency Monitoring System	

1. WORK DONE Cleaned datasets , and added new features , RSI_Label, difference , and Price Movement mainly to depict the dataframe better and do more visualizations for better understanding .
2. WORK TO BE DONE Prepare 2 models , one with RSI and one without , ,implementing stacking method.
3. PROBLEMS ENCOUNTERED Tried using time series custom model to create sliding window generator , which is function that take s 7 day price to predict , however it wasn’t working and model was too complex .
4. SELF EVALUATION OF THE PROGRESS Need to study more on other algorithms and shouldn’t focus on complex models , simple models actually can perform better .

Satish

Supervisor's signature　　　　　　　　Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3, 2023	**Study week no.:7**
Student Name & ID:2006649	
Supervisor: Dr Abdulkarim Janaan a/l Jebna	
Project Title: Global Currency Monitoring System	

1. **WORK DONE**

   Further tried working on the ARIMA but the model was too dependent on the parameters and there were problems with the import versions , hence I choose to opt for other algoritms

2. WORK TO BE DONE

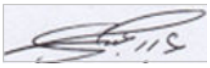
Study and implement model using Prophet , Facebook's machine learning model .

3. PROBLEMS ENCOUNTERED

ARIMA model was too complex to be used , and required longer working time . The MSE was not as good as previous models hence , it was opted out .

4. SELF EVALUATION OF THE PROGRESS

Good progress overall , need to put more effort on deployment too.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3,2023	**Study week no.:9**
Student Name & ID:2006649	
Supervisor: Dr Abdulkarim Janaan a/l Jebna	
Project Title: Global Currency Monitoring System	

1. WORK DONE

Successfully managed to complete my Prophet algoritm model .

2. WORK TO BE DONE

The MAE , MSE evaluation of the model was not good , hence need to try implementing other algoritms , however , upon discussing with supervisor , simpler models might perform better such as Random Forest Regressor and Linear Regression .

3. PROBLEMS ENCOUNTERED

Prophet model that I created was not reaching a good MSE value , it was underfitting instead .

4. SELF EVALUATION OF THE PROGRESS

Time was catching and Model 2 needs to be created sooner or later hence more effort is needed .

Satish

_________________________	_________________________
Supervisor's signature	Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3 , 2023	**Study week no.:11**
Student Name & ID:2006649	
Supervisor: Dr Abdulkarim Janaan a/l Jebna	
Project Title: Global Currency Monitoring System	

1. WORK DONE Finally tried using simpler models like linear regression and random forest regression for model 1 and model 2
2. WORK TO BE DONE Evaluation and deployment .
3. PROBLEMS ENCOUNTERED MSE and MAE values were very small and model was doing good prediction .
4. SELF EVALUATION OF THE PROGRESS Need to use Flask documentation to deploy model.

Satish

Supervisor’s signature   Student’s signature

POSTER


GLOBAL CURRENCY MONITORING SYSTEM
Objectives
• Create a currency monitoring system for Forex traders
• Enhance the usefulness of indicators in price predictions
• Improve the accuracy of trading signal predictions
• Compare and contrast different currencies for trading opportunities
CRISP - DM
Business Understanding
Data Understanding
Data Preparation
Evaluation
Modeling
Deployment
Methodology
PRICE PREDICTING
RSI
Simple Linear
Time Series

PLAGIARISM CHECK RESULT

Document Viewer

Turnitin Originality Report

Processed on: 15-Sep-2023 16:33 +08
ID: 2166524029
Word Count: 7910
Submitted: 3

Global Currency Monitoring System By Pang Chi Chong

Similarity Index	Similarity by Source	
11%	Internet Sources:	6%
	Publications:	5%
	Student Papers:	7%

exclude quoted | exclude bibliography | exclude small matches | mode: quickview (classic) report | print | refresh | download

1% match (Internet from 18-Jul-2021)
https://fict.utar.edu.my/documents/FYP/IIPSPW_template/IIPSPW_Report_Template_CN.docx

1% match (student papers from 16-May-2022)
Submitted to University of Hertfordshire on 2022-05-16

1% match (student papers from 03-May-2023)
Submitted to University of Salford on 2023-05-03

1% match (student papers from 07-Apr-2023)
Submitted to Monash University on 2023-04-07

1% match (student papers from 13-Dec-2021)
Submitted to Washington State University System on 2021-12-13

1% match (Internet from 03-May-2023)
https://www.analyticsvidhya.com/blog/2021/02/using-multiple-features-in-time-series-prediction-with-cnn-gru/

<1% match (Luttfi A. Al-Haddad, Alaa Abdulhady Jaber, Sinan A. Al-Haddad, Yousif M. Al-Muslim. "Fault diagnosis of actuator damage in UAVs using embedded recorded data and stacked machine learning models", The Journal of Supercomputing, 2023)
Luttfi A. Al-Haddad, Alaa Abdulhady Jaber, Sinan A. Al-Haddad, Yousif M. Al-Muslim. "Fault diagnosis of actuator damage in UAVs using embedded recorded data and stacked machine learning models", The Journal of Supercomputing, 2023

<1% match (student papers from 05-Sep-2022)
Submitted to University of Birmingham on 2022-09-05

<1% match (student papers from 21-Apr-2023)
Submitted to National College of Ireland on 2023-04-21

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Satish a/l Prabhagar @ Nagaiah
ID Number(s)	2006649
Programme / Course	IA
Title of Final Year Project	Global Currency Monitoring System

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: __11_____ **%** **Similarity by source** Internet Sources: ___6____________% Publications: ___5____________ % Student Papers: ___7____________ %	
Number of individual sources listed of more than 3% similarity: _0________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

______________________________ ______________________________

Signature of Supervisor — Signature of Co-Supervisor

Name: _Abdulkarim Kanaan Jebna — Name: __________________________

Date: __15/09/2023 — Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB06649
Student Name	Satish a/l Prabhagar @ Nagaiah
Supervisor Name	Dr Abdulkarim Janaan a/l Jebna

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

______ Satish _______________

(Signature of Student)

Date:15/9/2023