ANALYSIS AND PREDICTION OF STOCK VALUE USING MACHINE LEARNING

Ali Shafqat

03-243202-024

A thesis submitted in fulfillment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Department of Computer Sciences

BAHRIA UNIVERSITY LAHORE CAMPUS

November 2022

Approval for Examination

Scholar's Name: Ali Shafqat Registration No. 42861

Program of Study: Master of Science in Computer Science

Thesis Title: Analysis and Prediction of stock value using machine learning

It is to certify that the above scholar's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for examination. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index _____% that is within the permissible limit set by the HEC for the MS degree thesis. I have also found the thesis in a format recognized by the BU for the MS thesis.

Principal Supervisor's Signature: _____

Date: _____

Name: Dr. Ghulam Mustafa

Author's Declaration

I, Ali Shafqat hereby state that my MS thesis titled "Analysis and Prediction of stock value using machine learning
"

is my own work and has not been submitted previously by me for taking any degree from this university.

Bahria University (Name of university) or anywhere else this country/world.

At any time if my statement found to be incorrect even after my graduation the university has the right to withdraw/cancel my degree.

Name of Scholar: Ali Shafqat

Date: _____

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis titled "____Analysis and Prediction of stock value using machine learning_____"

is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Bahria University towards plagiarism. Therefore, I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of scholars are placed who submitted plagiarized thesis.

Scholar / Author's Sign: _____

Name of the Scholar: _____Ali Shafqat_____

Dedication

To my beloved mother and father

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Dr. Ghulam Mustafa , for encouragement, guidance, critics and friendship. I am also very thankful to my co-supervisors Professor Zunnurain Hussain and Associate Professor Dr. _____ for their guidance, advices and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

Librarians at Bahria University also deserve special thanks for their assistance in supplying the relevant literatures. My fellow postgraduate students should also be recognized for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members.

ABSTRACT

Stock market is one of the biggest, If not the biggest investment platform in the world. To invest in right shares, requires knowledge, study and information. This study consists of using publicly available historic data of cement sector's companies from Pakistan Stock Exchange. The purpose behind this research is to design a machine learning model using optimize it for our data to predict shares/equities value. We are using 4 algorithms including Logistic regression, Artificial Neural Networks so that we predict if stock value goes positive or negative and Long short-term memory and Linear regression to forecast stock price. All models have been validated with two different data split ratios. Model based on linear regression outperformed all with 1.41 RMSE, which we validated separately on Indian stock on 3 companies. Using this we got 52.07 RMSE on other market’s data as a generic model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Σ	-	Sigma/Summation
√	-	Square root
||	-	Mod
LR	-	Linear Regression
LSTM	-	Long short-term memory
SVM	-	Support Vector Machine
ANN	-	Artificial Neural Network
RSI	-	Relative Strength Index
MACD	-	Moving Average Convergence/Divergence
RIDOR	-	Ripple Down Rule Learner

TABLE OF APPENDICES

CHAPTER 1

INTRODUCTION

Technology has made every industry faster, efficient, better and connected more than ever. These industries use different software/programs that uses data not just to improve and develop their position in industry. With so much advancement in computer science in this era. We can not only use computers to manage and do things faster but can also use them to predict anything when given enough and accurate data. One of these industries are Stock Market or Stock Exchanges. A public traded company register itself in stock market where people can buy or sell its stocks that trade on a stock exchange. Stocks also referred as equities whose value can be depends on number of factors which includes company's size, its earning, profitability, reputation in market and its share's demand in market. As of September 21, New York stock exchange is the biggest market in the world and has equity market capitalization of just over 28.4 Trillian [1]. The value/price also fluctuate depending on these and many other factors.

Financial Analysts acknowledge that if we study visible and, in some cases, hidden economic facts and relationships can be observed. It is believed that the history of the market has patterns that give hints to the future of these stocks or finances [2]. To understand these theories these the patterns need to be studied and for that analysis of these facts exists.

There are mainly three types of analysis. First, The fundamental analysis, and second, the technical analysis and Lastly Sentimental Analysis. In Fundamental Analysis, we make use of Financial Ratios computed from the Financial Statements of the Company. It is used to calculate the real value or the Intrinsic value of an underlying asset. It is

generally used for a longer time frame. The examples of Fundamental Analysis tools are the P.E ratio, Price to Book ratio, etc. In Technical Analysis, we make use of Market data which is available in the form of Charts. It is used to calculate the real-time and Future movement of the stock. It is generally used for a short time frame. Examples of technical analysis tools are RSI, Moving Averages, MACD, etc. Fundamentals can be used to know the real worth of a stock, while the Technical can be used to time the entry and exit from the Stock. They can also be used in a combined form too [3]. In sentiment analysis we study the opinions/perception of a specific stock or asset. It can give idea at future price action at times. This is also an example of how trading psychology can affect a market, assisting as a forecasting tool to determine possible future price changes in a particular asset. The Sentiment not necessarily can always predict changes of a share. However, it is used with technical analysis which gives much better understanding to determined possible scenarios. stock sentiment can be influenced by various factors, which not only include industry related by also economic and political news and even social media. These factors help influence stock sentiment as they impact stock market volatility, trading volume and company earnings [4].

Our study is related to the technical analysis. The objective is to study and analyze the behavior of a firm's stock, optimize a machine learning algorithm according to it and get results to determine whether the value increase or decrease. The Target market of our study is PSX (Pakistan Stock Exchange) which established in early 2016 by merging three biggest stocks exchanges that were Lahore Stock Exchange, Islamabad Stock Exchange and Karachi Stock Exchange. Machine Learning Algorithm that are used in existing studies/research are Multi-Layer Perceptron algorithm [5], Logistic regression for stock performance [6] and Ripple Down Rule learner (RIDOR) Classifier learning [7].

1.1 Problem Statement

The stock market is one of the biggest investment platforms in the world. However, the stock market can be volatile and share values fluctuate all the time, so returns are never guaranteed. This makes investment risky. For this, multiple studies exist to help, and work is still getting done to make things easier for anyone who wants to buy equities. The target of this research is to predict these equities values with good accuracy rate.

1.2 Significance of Research

Millions of people are involved in share trading/buying and selling in the world. Even though everyone wants to make a profit out of it. Many people do end up losing it instead. To identify which stock to invest in many studies and research have been done and still ongoing to make it as accurate and safe investment as possible. However, as volatile stock market is, and research is a never-ending process. Much research are being done on ML to make a flexible system that can accurately calculate ups and downs of the market. Some studies have been able to achieve up to 90% accurate results in different scenarios and conditions so that's makes its scope very limited and still very far from a perfect solution for this problem.

Our research aims to contribute by developing a model with good accuracy and run it on another market's data to study how results differ from one market to another.

CHAPTER 2

LITERATURE REVIEW

The Several studies and related works have been done previously to predict different prices of different objects like, houses, cars, air tickets and Stock Market Prices around the world using different methodologies and approaches decade (2011–2021) with varying results of accuracy from 50% to 90%. One of the latest studies published in 2021 of collective approaches has been done in which systematics of ml-based approaches are explained and test of a generic framework. It includes findings from the last decade starting 2011 to 2021 that were taken from online digital DBs which includes ACM and Scopus [8]. Following Algorithms were used in these studies.

- Support Vector machine
- Naïve Bayes
- Fuzzy Algorithms
- Artificial Neural Network
- Genetic Algorithms
- Deep Neural Networks
- Regressions Algorithms
- Hybrid approaches

The study showed SVM is the highest used technique for SMP [9]. ANN and DNN are also widely used due to their more accurate and faster predictions. The study also shows that by including textual data with market data, The accuracy rate improved. Some other

studies form Pakistan's region and stock market that weren't part of aforementioned research.

In 2016 Mehak Usmani, Hasan Adil, Kamran Raza and Saad Azhar applied multiple Machine Learning Techniques on limited Karachi Stock Exchange Dataset for a comparative study and the result suggest that KSE works on a pattern approach which are recognizable and predictable by using different machine learning techniques the result generated on such limited dataset is as follow: (Single Layer Perceptron (SLP), 60%), (Multi-Layer Perceptron (MLP), 77%), (Radial Basis Function (RBF), 63%) and (Support Vector Machine (SVM),60%) [5] respectively.

In 2018 Syed Shahan Ali, Muhammad Mubeen, Irfan Lalc and Adnan Hussain used logistic regression on PSX dataset from 2011 – 2015 containing 109 listed non-financial companies. The research consists of studying sales growth, earning performance, D/E and P/B ratios. Altman and ohlson's regression model are improved in this research. The predicted/ dependent variable returns 0 and 1 for being positive and negative value. These values are returned on the basis of p value which is the probability of a company's outcome is Good. A value of 0.5 or greater for p is considered as good. After that H and L (Hosmer and Lexmeshow) method to compare the difference between observed and predicted data which can only be applied on Binary values [9]. This test is run using simulation and is based on chi square method to avoid duplication in the subpopulation. For this, the observations based on divided based on the expected probabilities. In this case, the method has divided observation into 10 ordered groups. The assumption is that the difference in expected and observed value is zero. Results showed 4% value difference between chi squad and p value which means it is very unlikely to have occurred given the null hypothesis. The study showed that there's no statistically difference observed and predicated result and were able to attain 89.77% of accuracy and were able to predict good and bad stock for investment purposes [6].

In 2017 Wasim Akram and Muhammad Imran used Ripple-down-rule-learner on stock market data set of Pakistan to attain 90% of accuracy and to boost the trend of stock market price prediction. The proposed approach used base classification algorithm which was used for heart disease for medical research by Koklu et al which was used by Shriwas

et al. RIDOR was used for the case of uncertainties in data. At start RIDOR generates defaults values with minimum error rate. These values are used for certain situations and when these situations arrive, by using incremental rule, these values loop through to keep the error to a minimum and produce accurate results. The dataset was taken of year 1997 to 2016 with 4656 trading days. Date was divided for selection parameter, training and testing purposes. The testing data was used for validation. Technical indicators were observed in three perspectives of 3, 5 and 10 days respectively which is used to find relative strength index, momentum simple and weighted moving average. JRIP, Random Forest, SVM and Naïve Bayes were also tested in this study. By using base classification and RIDOR gives efficient results in uncertainties. This new approach helps them get a 90% accuracy where other Machine Learning techniques were (JRIP, 89%), (SVM, 89%) and (Naïve-Bayes, 89%) [7].

2.1 Research Gap

One of the most recent research works by Rouf, Nusrat, Majid B [8] which does comparative analysis of multiple studies of a decade and implement with their respective data. The data used ranged from year 2000 to year 2020 with accuracy 70% to 90%. In this research we will develop a model with better accuracy and run it on another market's data to study how results differ from one market to another.

2.2 Aim and Objectives

Based on our literature review we have learned that there are many studies for SMP with various algorithm. The highest accuracy level we got is 89.7% on PSX data using fundamental analysis variables. However, all these studies share one common pattern which is each and every study only use 1 selective market data to validate. Which doesn't tell us much how these authentic these models are when it compares to another market. Our objective is to run our designed model on Indian Market data to see how it performs.

The aim of this research is to create a model using models that includes ANN, Logistic Regression, LSTM and Linear Regression that gives us least error rate and most accuracy according to our dataset.

2.3 Key Research Questions

Text Following are the questions that will be addressed in this research work.

1. How to develop a model using Neural Networks Classifier with improve accuracy that predict a company's stock value?

2. Once model is validated how it performs on others market data?

CHAPTER 3

METHODOLOGY

Research can be done in multiple ways. There are various methods, techniques to find Data, Information and other people's research which helps us to find the answers to our research related questions. We are following Figure 3.1 steps in our research related methodology. First, we find, and Literature review existing studies related to our work, then we will thoroughly study it to find the research gap, improve the existing work or to find a better approach to solve a problem. When done with it Gather or Collect data to be used in our research. We need to optimize or preprocessed our dataset to get the best possible results. Once the model (LR, ANN and LSTM) is ready it needs to be validated for error corrections or verify results. At last, Analyze the results and discuss the findings and conclude the research

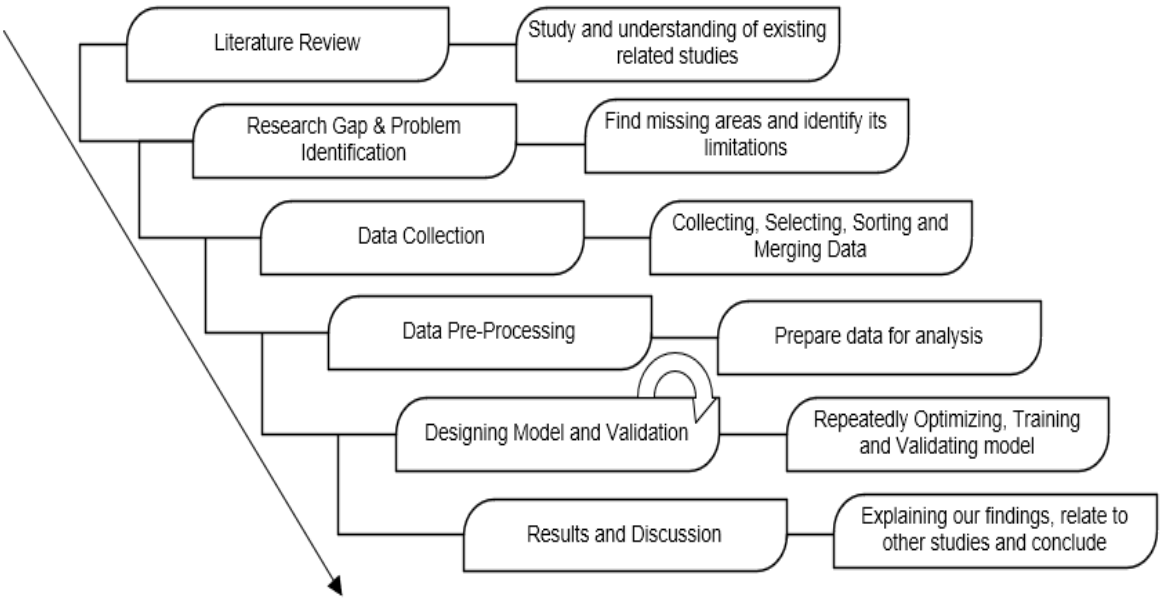

Figure 3.1: Propose Methodology

3.1 Literature Review:

The First Phase starts with discovery and exploration of the existing research related to our study understand what is already done in the relevant field.

3.2 Find Research Gap and problem finding:

The second phase literature review will be used to find out the research gap and shortcoming of the research. Literature review is necessary to find the problem/gap for our work.

3.3 Data collection:

Collecting data is one of the most important part of any research for it to be validated. So, this phase solely focusses on getting the required data that is needed for the research. For our research We have used Pakistan Cement Sector Companies. We have used online platform investing.com to get the required data. The Data is collected for each company separately. Once the required data is obtained all of it combined into one file.

3.4 Data Pre-Processing:

In this phase collected data gets prepossessed and optimized. We preprocess data to remove or eliminate null data, to standardize dataset which in our case isn't needed as all our dataset is taken from the same source. To handle categorial data we either use label encoding or convert strings binary vector stream. With numerical data in one or more columns, to assign the same weight we need to normalize or standardize those columns. For classification we convert change percentage to positive and negative or 0 and 1. After data preprocessing unnecessary or redundant data gets remove for better and more accurate analysis. With the removal of Noise and artifacts not just help us to manage and understand things better but can also help towards more accurate results.

3.5 Designing model and Validation:

In this phase, we use the processed data and usually execute multiple models to check which how these models handle data. We are using Neural Network Classifier and tweaking it (if needed.) according to our desired results, the model needs to be validated for which Test train split Validation will be used in which specific percentage of data removed/extracted before training the model. Once the trained model is ready that removed data will be used to validate its performance. Due to having relatively small dataset we are using both 80-20 and 90-10 split for validation [10],[11]. As we are using two different problem types in this research. Both will be evaluated via different measures.

3.5.1 Accuracy

Accuracy is a metric used to evaluate a classification model. When our model predicts if the share price will simply increase or decrease. Accuracy will be calculated simply by the number of correct predictions divide by total number of predictions.

Accuracy = Correct Predictions ⁄ Total number of predations

3.5.2 Precision

Precision is a metric also used for classification model. It measures a model by calculating correctly identifying the positive class. Precision can be calculated by the following formula [12].

Precision = True Positive ⁄ (True Positives + False Positives)

3.5.3 MSE/RMSE

Metrics like Accuracy and Precision cannot evaluate a regression model as the output in such cases doesn't belong to a class. For a regression problem we used metrics like MSE/RMSE to evaluate the regression model. MSE refer to the mean square error and RMSE refers to root mean square error. These are absolute measure of the goodness for the fit which is calculated by taking the different between forecasted value and the actual value and square them (eliminates any negative value) and taking its average

MSE formula = (1/n) * Σ (actual – forecast) ^2

RMSE = √ ((1/n) * Σ (actual – forecast) ^2)

As MSE is an error, The lower the error the better our model is and unlike classification metrics like accuracy, MSE is not limited to 100.

3.5.4 MAE (Mean Absolute Error)

Mean absolute error is very similar to MSE. It takes sum of square of absolute value of error whereas MSE takes sum of square of error. The Mean absolute error can be calculated by the following formula [13].

MAE formula = (1/n) * Σ |actual – forecast|

3.6 Results and discussions:

This is the final phase in which we examine and discuss results and future work related to the research. In order to better understand, we explain our results and then set

those results into the context of the literature. The discussion states the new findings in the research and then explains and compares the results with the previously proposed model with our proposed model. Future work refers to the points out how new approaches in the current study can extend and improves it. This can give other researchers valuable information for a new idea or direction.

CHAPTER 4

DATA ANALYSIS/RESULTS/FINDINGS

This research starts with collecting data on which model will be based on. Once we have data which can be collected from one or sources. The only requirement is that data needs to be from a reliable source which can be private or public. we can start working on optimizing it according to model needs. We have gathered data from Investing.com which is owned by Fusion media limited which has the data of 250 exchanges across the entire world. According to Similar web and Alexa it’s one of the top three global financial websites [14]. The dataset contains Pakistan's cement sector companies e.g., Lucky cement, Pioneer cement, D. G. Khan cement and 14 other companies registered in Pakistan Stock Exchange However two companies Dadabhoy Cement Industries and Dandot Cement Company which have been defaulted at the time of this research are not included. After combining all, the complete data set consist of 71873 rows which makes up to 7670 days per company. A snap of data is added below.

Name	Date	Open	High	Low	Vol full	value chai	Price
LUKC	6/23/2021	909	917	901	236750	0	902.12
LUKC	6/22/2021	917.9	918.89	908	202710	0	909.97
LUKC	6/21/2021	923	923.9	912	266460	0	912.75
LUKC	6/18/2021	918.5	933	915.85	523260	1	925.05
LUKC	6/17/2021	914.99	924.9	912	521050	1	913.23
LUKC	6/16/2021	914	916.99	909	373460	0	910.85
BEST	4/9/2021	156.1	160	156.05	8700	1	160
BEST	4/8/2021	156.1	158.5	155.04	1500	0	155.52
BEST	4/7/2021	159.5	159.5	153.02	3300	0	156.01
BEST	4/6/2021	155.01	160	152.01	27300	1	159.27
BEST	4/5/2021	151	154	150.11	1700	1	153.98
BEST	4/2/2021	156.26	156.5	151.1	4000	0	152.73
ATOC	12/27/2021	135	137	134	7700	0	137
ATOC	12/24/2021	138.5	138.5	135.01	1700	1	137.75
ATOC	12/23/2021	143.84	143.85	136.11	13000	0	136.11
ATOC	12/22/2021	138.71	138.71	138.71	100	0	138.71
ATOC	12/21/2021	141.8	141.8	136.3	2700	1	139.4
ATOC	12/20/2021	135	138.8	134	9700	1	138.74

Figure 4.1: Data Chunk

4.1 Data Analysis

The Original Dataset consists of Open (The price of share when trading starts), High (The highest price of a share of the day), Low (The Lowest price of the share of that day), Price (The final price when trading ends) vol (The total number of shares traded in a day) and change (percentage in the price difference between two days). The data ranges from Jan, 2001 to Dec, 2021. However, some existing data is extended and newly added (derived from known information) to simplify and normalize the dataset. The modified data include Vol, Change, and Date. In Volume, we have values in thousand K and millions M form. To remove the dependency of K and M we can either move K and M to a new feature which will be considered a separate feature by our model or simply convert the vol to full digits values. In our case, we have settled for the simpler approach which is modifying the volume to full values. In Change, we have the change of price between days in percentage. This value is mostly between -10% to +10% but isn't limited to it because share price can increase to twice or more or decrease to half or less in very rare oceans which significantly changes the range. In these instances, it becomes very hard to predict

the change in price. For this, we have converted Change from percentage to a value of 0 and 1. This conversion not just simplifies change but also predicting increase/decreases rather than change makes it a classification problem. The data is given in series of datetime which is typically used in timeseries problem, but the classification model doesn't support datetime feature. To use this feature, we have to convert datetime into a timestamp sequence which can be transformed.

The only new data is added in original dataset is the name feature. As we mentioned before the collected data consist of multiple companies separately. After that we merged all the data which means we can have inconsistent values of price, open, low and high but with the same date. The companies' name feature can help our model differentiate in such conditions while adding a feature it can map to data can improve its overall accuracy.

4.2 Applied Methods and Experimentation

For this research, we handled the problem with two different approaches i.e., Regression and classification, not only that we used multiple existing model algorithms on our dataset. This helps us to understand which works better and having the same dataset makes things easier and unbiased. The reason to use multiple algorithms is to get the most accurate results. The models used were Linear Regression, Logistic regression, LSTM, and multilayer neural networks. Logistic Regression is selected from [5] Study. Implementing it on different technical data instead of fundamental. ANN is selected to learn the hidden and complex patterns in our dataset. The ability to overcome underfitting/overfitting by controlling the change estimated error each time the model weights are updated also referred as learning rate or adjusting other parameters including batch size, units, dropout and number of iterations. Linear regression is selected because on observing the data it was observed the general trend of stock price was increased with respect to time. LSTM is considered to be one of the best algorithms for time series problem. because of the looping constraint that captures sequential information. All methods and algorithms start with some common steps. These steps start with importing needed libraries. After that dataset needs to be imported for which we can use built-in pandas or sklearn (Depending on if we read SK 2d array or data frame) to read CSV file. When reading the CSV file, we can define header (Row no to infer as column name) or can use the name to enter column names,

Separators (delimiters to use for data separation) and dtype (to define data types of columns). Once we have the dataset, we need to generate variable vectors by dividing columns for inputs. and output. Now, with features separated, we need to split some of the data for testing. The goal is to build a model that performs well on new data. For this, we need unseen data which hasn't been used for training the model that is why we are using a test train split for model validation. Train test split validation procedure allows us to simulate how a model would perform on new data. 20% of total rows are separated with random chunks from training data with random state sets to 0 to get different data train test split each time code is executed. Once we get our features it's typical to handle null, missing, and outlier values which are not present in our data. However, we need to perform feature scaling. In Feature scaling, we transform data which typically consists of strings to convert into a range of numbers. In some cases, if data contain a vast range of values, it can confuse our model to assign uneven weightage of features. To ensure that model gives equal weightage to the inputs we normalize or standardize our data. In normalization, we rescale the values into the range from 0 to 1 in a dataset or column whereas a standardized dataset will have a mean of 0 and a standard deviation of 1 with no specific positive or negative values limit. After these steps, the procedure starts to differ depending upon the algorithm.

With logistic regression, we estimated the probability of an event occurring. In our case, this is done by calculating the probability of hikes and falls of the share. The highest probability tells us if the price will rise or fall which logistic regression is used with classification. While using Logistic regression we were only able to get 68.6% accurate results. Syed Shahan Ali also used logistic regression for their study and was able to get 88% accuracy results with it. However, they used fundamental variables which gives the 6 additional features or independent variables than us. These features are Earning per share, Price to Book Value, Return of Equity, Current Ratio, Debt of Equity, and Percentage changes in Net Sales. Also, 109 firms' data were used for their research.

An artificial neural network is a machine learning algorithm based on human brain neurons. We used this information processing technique to classify the value of a stock using our data. MATLAB Classifier tool and Tensor flow Python library were used to

implement the model. A neutral network may consist of 3 layers. Input (raw information that feeds into the networks), Hidden layer (Between input and output layers in which neurons assign weighted using an activation function), and finally output layer (gives us the final prediction). First, we need to initialize the model and then needs to define its layers. In our case, with multi variable dependency on price. 2 layers is enough as the data is not much complex with fewer features than number of neural networks. The number of neutrons determined by the number of 6 inputs with all possible dependency combinations. This could be 1 on 1 or anywhere from 2 to 6 combination. Separate combination of 6 inputs gives us 20 while all non-repeatable combination could be up to 30. This gives us the number of neurons in our network. Lastly, we have 1 output layer with 1 unit. The hidden layer uses Relu activation function while the output uses the sigmoid activation function. An activation function determines if a neuron activates or not by using the weighted sum. Once layers are added we need to define how the model will compile for which we need to define optimizer, and loss functions. An Optimizer is a method that changes the attributes of a network like weights and learning rates while the loss function quantifies the difference between the produced output and actual output. The less the loss, the better a model gets. We trained our model with 1000 epochs to ensure it doesn't underfit our data. Consistent loss drop was observed till 600 epochs with 0.577 on binary cross entropy loss function after that loss keep fluctuate between 0.578 to 0.573. A batch size of 32 was used which means 32 samples from our trainer data get used before reassigning the weights of the nodes. The designed model figure is added below.

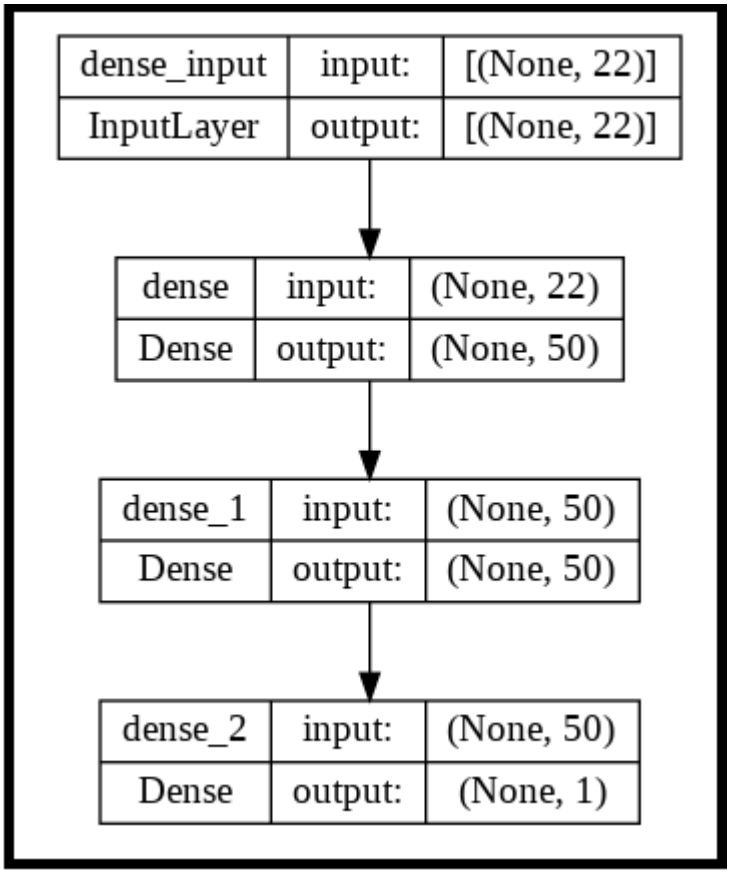

Figure 4.2: ANN Model

The model took almost 40 minutes to train which can change depending on the hardware of the computer our environment is running on. We are validating our model in the matrices of accuracy which gave us 73% in this case in the training phase. While when testing it with test split data it gave us 71% accurate results.

LSTM (Long short-term memory) is the first algorithm we used that trains a model for regression problem in our research. LSTM is good at processing long sequences because it contains an internal mechanism called gates. These gates can identify the data which are important to include or exclude according to their relevancy. By doing that, it can pass relevant information down the long chain of sequences to make predictions which in our case is useful as it contains long sequence price values which respect to date [15]. Just like an artificial neural network once the model is initialized. We need to define its layers. For our study, we added two hidden layers with the 32 units and one output layer

with 1 unit. The activation function used is Relu. The designed model figure is added below.

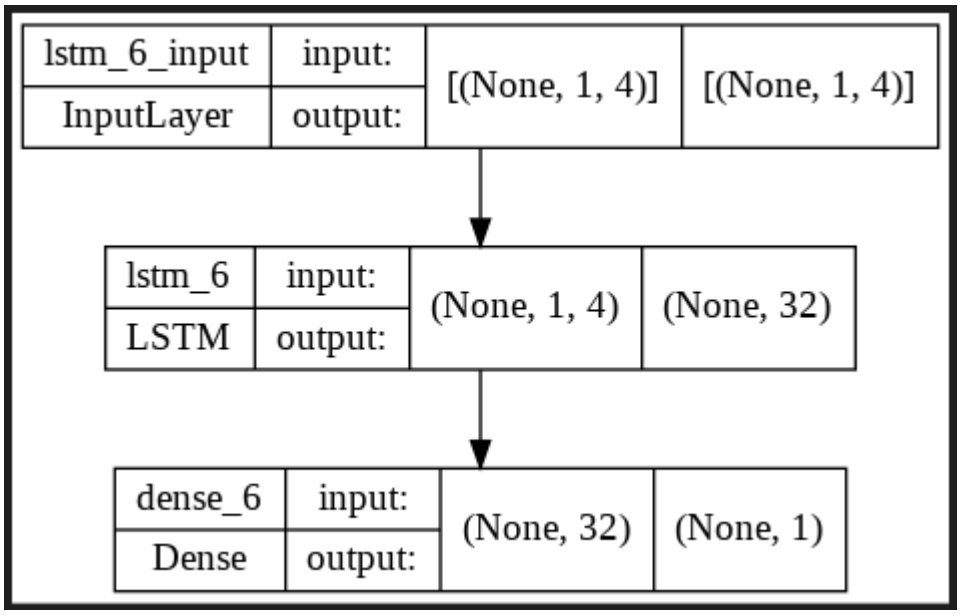

Figure 4.3: LSTM Model

After that, we compile/fit our model with 250 iterations. When training with 80% Data LSTM showed signs of overfitting very early in the training. For this we changed Activation function from Adams to RMSprop with learning rate of 0.0025 and 750 iterations before showing signs of overfitting. Adam calculates adaptive learning rate for each parameter and converges faster than RMSprop because of the way Adam store both individual learning rate of RMSprop and the weighted average of momentum makes it faster in training which is why, when training with the 80% data them momentum our neural network gets overfitted much faster in Adam. For this model, the validation matric is mean square error. The mean square error represents the distance or how close a set of points are which in our case is forecasted values with respect to a regression line. The lower this error is, the better the forecast. The model takes around 10 minutes to train with at least 0.76 MSE on training data validation and 2.28 MSE on testing data validation. The output given by the model is shown in figure 4.1 below.

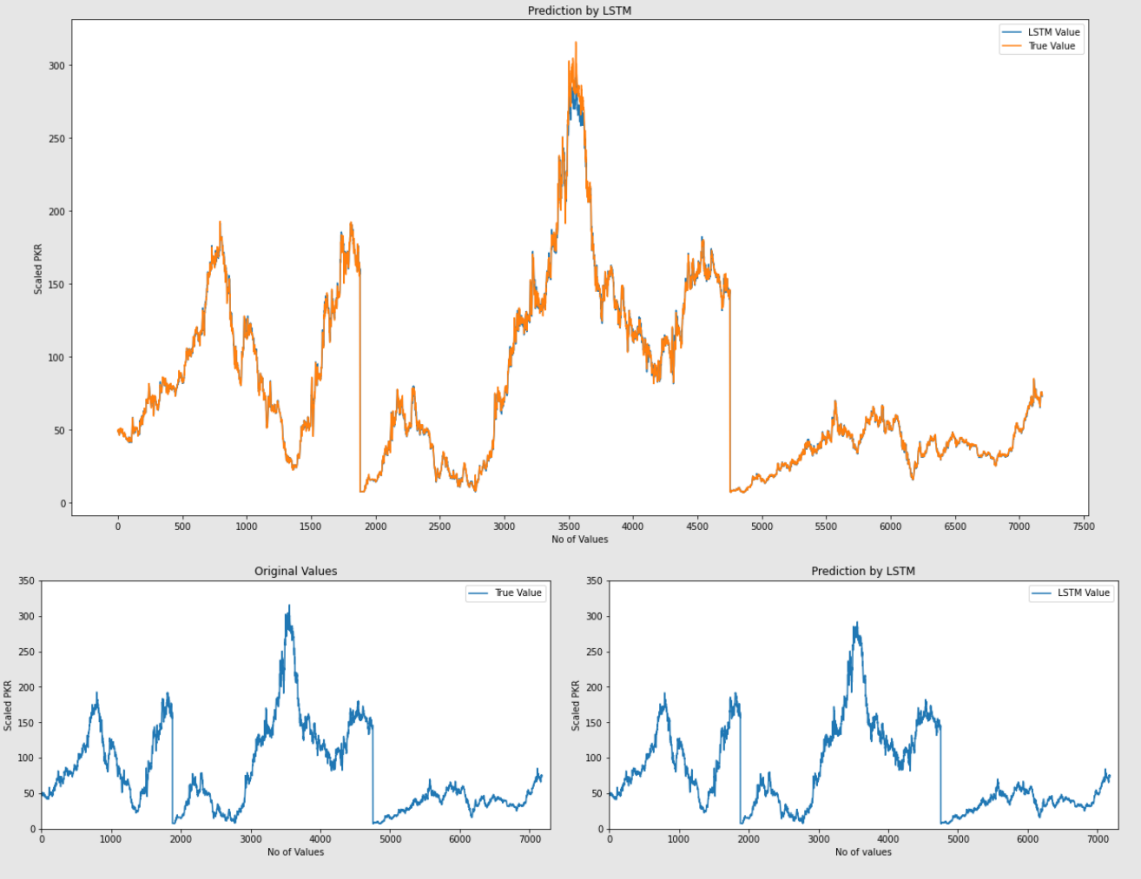

Figure 4.4: LSTM Predicted and Actual value Graph

Linear regression is one of the most commonly used type of predictive analysis that fits a straight line that minimizes the difference between predicted and actual output values. Linear regression works by calculating the regression coefficient, regression slope, and Test value. A regression coefficient is the same thing as the slope of the line of the regression equation which is determined by the relation between dependent and independent variables by calculating the slope. the t-test is a statistical hypothesis testing technique that is used to test the linearity of the relationship between the response variable and different predictor variables. In other words, it is used to determine whether or not there is a linear correlation between the response and predictor variables. The t-test helps to determine if this linear relationship is statistically significant. The formula for the one-sample t-test is t = (m – m0) / SE. in this formula m is the linear slope or the coefficient value obtained using the least square method, and m0 is the hypothesized value of linear slope or the coefficient of the predictor variable. The value of m0 = 0. SE represents the

standard error of estimation and t is the t-test statistic. The stand error of estimation can be estimated using the following formula: SE= S /√N. In this, S is the standard deviation and N is the total number of data points [16]. In our research, we perceive the relationship between the dependent variable is not as simple to give as a straight-line slope for positive or negative due to the fluctuation in price. However, the independent variables like open, low, and high can have a relationship with respect to price. We recognize high will always be greater than price and low will always be smaller than price or in some very rare instances, one will be equal to the price. In the worst-case scenario, both will be equal to the price but that has a very low probability of occurrence. So, keeping that in mind we generate distribution graphs for all variables and train our linear regression model. It is one of the most efficient models and only takes few minutes to train. LR gives us 1.41 MSE on Testing data and 0.68 on Training data.

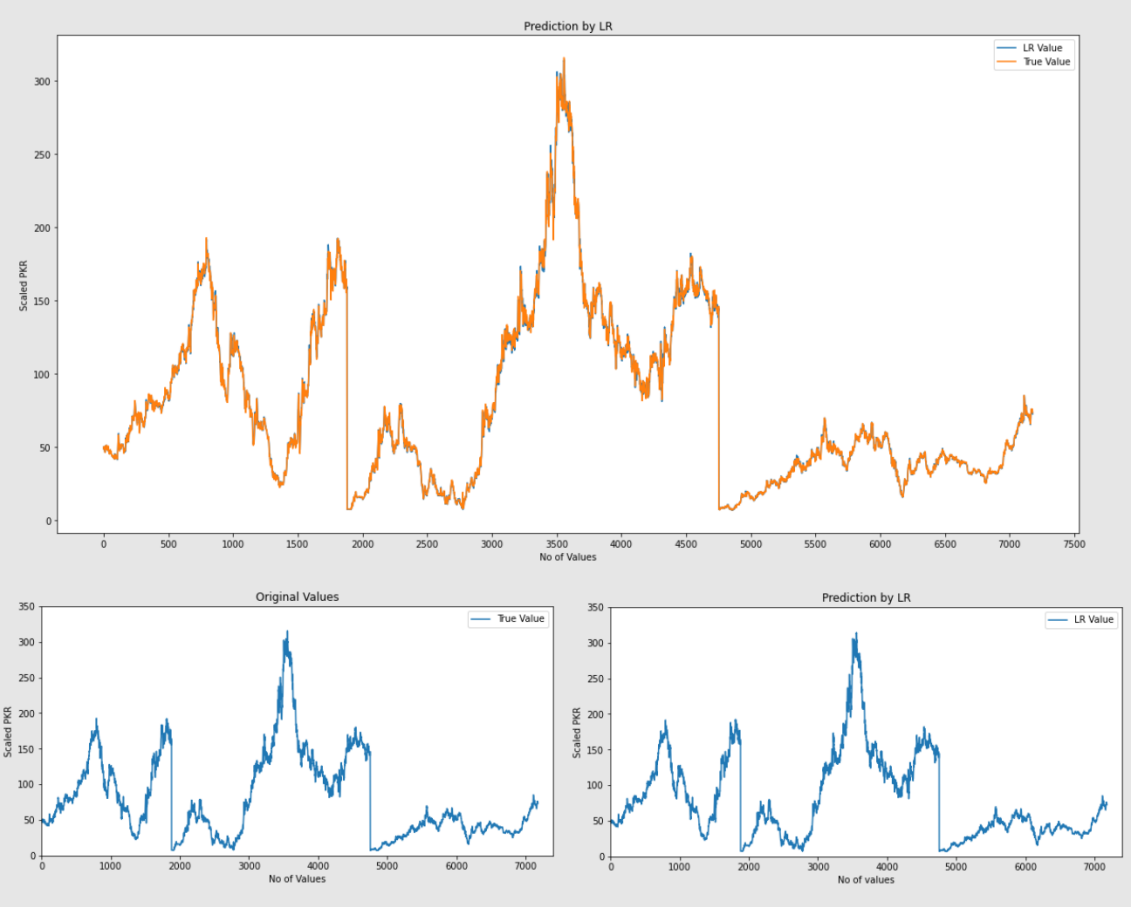

Figure 4.5: Linear Regression Predicted and actual values graph

4.1 Validating Model with Separate Market's Data

To Answer our second research question that how our model performs on other markets' data? we have used 3 Indian stock market companies. These companies include JK Lakshmi Cement Ltd, India Cements Ltd and Everest Industries Ltd. The Data ranges from January 2013 to December 2021. The total data consist of 9068 rows combined. This data is collected from the same source as our training dataset for consistency. Forecasting stock price using our model trained with 90% data gave the following results

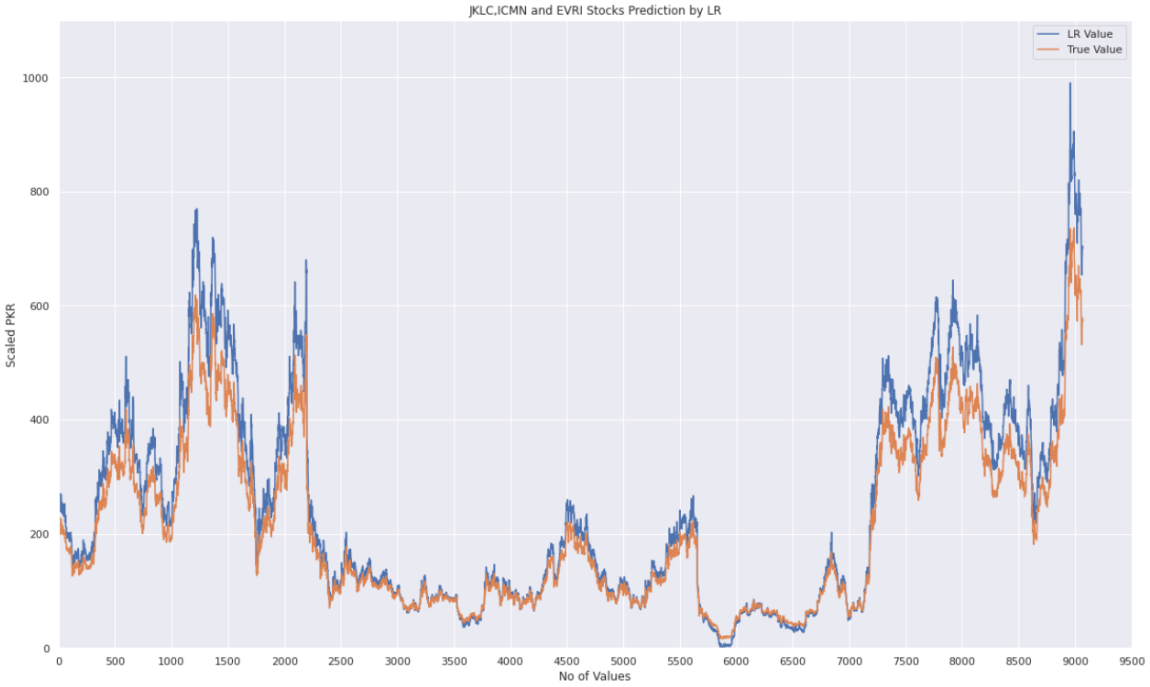

Figure 4.6: Predicted value for JKLC, ICMN and EVRI

The model gave us Absolute error and Root Mean square error of 36.94 and 52.07 respectively. This error is much more when compared with 1.32 which we got when validated previously but we need to keep in mind previous validation data was part of the original dataset our machine trained on. Even though we extracted that data before training so that during validation it was the unseen data for the model that doesn't change the fact that data was a part of the machine that was used for model training. In this scenario, not only the price is unknown to the model but it's also not aware of the company. While the accuracy metric is not anywhere close to using any real-life application. A separate market means the trends follow different principles and situations. One positive outcome that our

model has given is that it has most successfully been able to predict the upwards and downward trends on multiple levels for both companies. On KERI especially, the model not just predicted the trend but around 2018 forecasted values are very close to the predicted values. With more data and a better-optimized model someday we might be able to generate one generic model for all the markets.

4.2 Results

We have used four different methods in our research. These methods include Logistic regression, Artifice neural networks, Linear regression, and long short-term memory with classification and regression approaches. To evaluate and validate both approaches we had to use different performance measures. For classification, we use Accuracy and precision whereas regression is evaluated in terms of mean square error. All the results are shown in the table below. After validating all four models. We imported this dataset and passed it to our model to observe the results.

Table 4.1: Classification Results

Performance Metrics	Algorithm	Model Validation	
		Training Data	Testing Data
Accuracy	ANN	69.19	67.51
	Logistic Regression	62.55	62.11
Precision	ANN	72.13	69.39
	Logistic Regression	73.74	71.95

Table 4.2: Classification Results on 20% data validation

Performance Metrics	Algorithm	Model Validation on 20% data	
		Training Data	Testing Data
Accuracy	ANN	70.20	68.94
	Logistic Regression	62.34	61.76
Precision	ANN	7 bb2.20	73.46
	Logistic Regression	73.74	73.01

Classification model results of training and testing data are compared in Table 4.1 and Table 4.2. The results didn't differ much after changing validation percentage. Both models performed better on training data which is quite typical. The difference between training and test score showed that the total data was just enough to avoid underfitting/underfitting but not enough to have accurate enough results

Table 4.3: Regression Results

Performance Metrics	Algorithm	Model Validation	
		Training Data	Testing Data
MAE	Linear Regression	0.56	0.59
	LSTM	0.41	1.16
RMSE	Linear Regression	0.68	1.41
	LSTM	0.76	2.28

Table 4.4: Regression Results on 20% data validation

Performance Metrics	Algorithm	Model Validation on 20% Data	
		Training Data	Testing Data
MAE	Linear Regression	0.35	1.46
	LSTM	0.38	1.95
RMSE	Linear Regression	0.68	2.59
	LSTM	1.09	3.93

Regression model results of training and testing data are compared in Table 4.3 and Table 4.4. Both LSTM and LR showed signs of overfitting when validated with 80-20 split. LR error rate significantly improved in 90/10 split against both training and testing data. While LSTM showed only slight improvement. Just like previous models these two also performed better on training data. While LSTM did perform very well on training data. Those results couldn't keep up with the testing data. The score difference showed the model was overfitted in this case. Further training the model only increased the gap. Moreover, it took the longest to train which makes it the worst in terms of efficiency. More data is required to better fit LSTM in this case. The linear regression model with 1.41 MSE is the best performant among all four models. Not only that it is also the most efficient (Along with logistic regression).

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1 Discussion

This study has shown four working Machine learning models. For Logistic regression and Artificial Neural networks, which gave us the accuracy of 67% and 61% which isn't enough to persuade us because there isn't much room for improvement. These results showed us that changing the problem type from regression to classification is possible however it doesn't necessarily mean that we can get improved results from it. The other two algorithms for the regression problem showed us much better results, only linear regression showed consistency in both training and testing. LSTM gave us a decent 0.76 RMSE on training data which could have been further improved with more iterations but when the model was validated on testing data, it RMSE of greater than 5 which indicates that the model is overfitting. In the case of overfitting, the performance on the training sets continues to improve whereas for validation it improves to a point and then begins to degrade after hitting an inflection point on the loss graph. The model fitting had been able to improve to 3.3 by increasing the batch size, increasing training data with 90-10 split and reducing the number of epochs. Improved error rate on both training and testing data. After attempting to optimize the model and Adjusting parameters we determined that model needs more data for a better fit. The Linear regression-based model turns up to be the most accurate and balanced model not only that it took the least amount to train which makes it the most efficient model along with logistic regression. This is the reason, the LR model is selected for India's market data prediction. We got 52.07 RMSE which didn't give us much to work on, but we could theorize a few reasons for this high error rate. The most

logical reasoning is India's stock market with a volatility score is 24.87 is much more volatile than Pakistan's market whose score is 18.91. This higher volatility makes it harder to predict. This can be seen in the results where the model is able to predict a few trends but the point at which trends changes has a noticeable difference due to the fluctuation of price caused by the market's volatility.

Overall, this study has shown positive results to predict stock market trends of a company forecasting stock value using the LR model but leaves much to be desired when validating the model on other market data.

5.2 Conclusion

The stock market prediction to this day remains a complicated task due to its volatility and dozens of factors. This research used two methods to forecast the value of stock in the cement sector from the Pakistan Stock Exchange. This study only used technical data that is more readily available compared to fundamental. The regression problem method gave results much better than the classification approach that followed referenced studies to classify the value to be positive or negative. Linear Regression showed the best results among all four algorithms. The model has been able to predict the price with 1.41 root means square error and is expected to improve with more data. The same model when validated on separate stock market data only able to predict the trend of that particular company to some extent.

5.3 Future Work

A lot of work is ongoing in the field of AI and ML for safe investment in stock prediction. This study has been conducted and performed on a small dataset of 75K. At the time of the study, this is pretty much all the data available in the PSX cement sector. This has caused some models to underperform especially LSTM. A similar analysis with a bigger dataset could help significantly improve these results. A model trained on more than 1 market data could also help to cover more than one segment of the market. The effort of creating a generic model for multiple stock markets should keep going. A market with

similar economic indicators and close volatility scores could help to close the gap in market differences.

REFERENCES

[1] Ali, A., The World's 10 Largest Stock Markets. [online] Visual Capitalist. Available at: <https://www.visualcapitalist.com/the-worlds-10-largest-stock-markets/>, 2021.

[2] Roberts, Harry V. "Stock-Market 'Patterns' and Financial Analysis: Methodological Suggestions." The Journal of Finance, vol. 14, no. 1, [American Finance Association, Wiley], pp. 1–10, https://doi.org/10.2307/2976094, 1959.

[3] Deyagond, S. The basics of stock market—Part 1. [online] Medium. Available at: <https://medium.com/the-phi/the-basics-of-stock-market-part-1-626c1bf68755>,2020.

[4] Costa, T., Using Sentiment Analysis to Examine Stocks. [online] DailyFX. Available at: <https://www.dailyfx.com/education/understanding-the-stock-market/stock-market-sentiment-analysis.html>, 2020.

[5] Usmani, M., Adil, S., Raza, K. and Ali, S. Stock market prediction using machine learning techniques. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016.

[6] Ali, S. S., Mubeen, M., Lal, I.., & Hussain, A., Prediction of Stock Performance by Using Logistic Regression Model: Evidence from Pakistan Stock Exchange (PSX). Asian Journal of Empirical Research, 8(7), 247–258, 2018.

[7] Akram, W. and Imran, M., Pakistan stock exchange prediction using RIDOR classifier. INTERNATIONAL JOURNAL OF ADVANCED AND APPLIED SCIENCES, 4(9), pp.130-137, 2017.

[8] Rouf, N., Malik, M., Arif, T., Sharma, S., Singh, S., Aich, S. and Kim, H., Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. Electronics, 10(21), p.2717, 2021.

[9] Hosmer, D.W. and Lemeshow, S. Applied Logistic Regression. John Wiley & Sons, Inc., New York. 1989.

[10] Nguyen, Quang & Ly, Hai-Bang & Lanh, Ho & Al-Ansari, Nadhir & Le, Hiep & Van Quan, Tran & Prakash, Indra & Pham, Binh. (2021). Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. Mathematical Problems in Engineering. 2021. 10.1155/2021/4832864.

[11] Joseph, V., Optimal Ratio for Data Splitting. [online] Diva-portal.org. Available at: <http://www.diva-portal.org/smash/get/diva2:1526845/FULLTEXT01.pdf>, 2022.

[12] Vickery, R., 8 Metrics to Measure Classification Performance. [online] Towards Data Science. Available at: <https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa>,2021.

[13] Wu, S., What are the best metrics to evaluate your regression model?. [online] Towards Data Science. Available at: https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>, 2020.

[14] Investing.com., About Investing.com. [online] Available at: https://www.investing.com/about-us/>,2017.

[15] Phi, M. Illustrated Guide to LSTM's and GRU's: A step by step explanation. [online] Towardsdatascience. Available at: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>, 2018.

[16] Kumar, A. Linear regression t-test: Formula, Example - Data Analytics. [online] Data Analytics. Available at: https://vitalflux.com/linear-regression-t-test-formula>, 2022.

APPENDIX A

Source code

```
#Logistic regression

import pandas as pd

from sklearn.preprocessing import LabelEncoder

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import drive

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

drive.mount('/content/drive')

# load dataset

pima = pd.read_csv("/content/drive/MyDrive/Dataset/Datasetneededdate.csv", header=0)

feature_cols = ['Name', 'Date', 'Open', 'High','Low','Vol full']

X = pima[feature_cols] # Features

y = pima['value change'] # Target variable

encoder = LabelEncoder()
```

```
# Encoding Categorical Labels

X['Name'] = encoder.fit_transform(X['Name'])

import datetime as dt

X['Date'] = pd.to_datetime(X['Date'])

X['Date']= X['Date'].map(dt.datetime.toordinal)

scaler = StandardScaler()

X = scaler.fit_transform(X)

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)

from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)

logreg = LogisticRegression()

# fit the model with data

logreg.fit(X_train,y_train)

y_pred=logreg.predict(X_test)

y_predtraining = logreg.predict(X_train)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

print("Precision:",metrics.precision_score(y_test, y_pred))

print("Train Accuracy:",metrics.accuracy_score(y_train, y_predtraining))

print("Train Precision:",metrics.precision_score(y_train, y_predtraining))
```

```
#ANN

import numpy as np

import pandas as pd

import tensorflow as tf

from google.colab import drive

from sklearn.preprocessing import LabelEncoder

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

from sklearn.model_selection import train_test_split

drive.mount('/content/drive')

data = pd.read_csv("/content/drive/MyDrive/Dataset/Datasetneeded.csv",header = 0, dty
pe = {"name" : "category"} )

#Generating Variable Vectors

X = data.iloc[:,:-1].values

Y = data.iloc[:,-1].values

county =  np.count_nonzero(Y)

num_rows, num_cols = X.shape

yrows = Y.shape

print("XRows",num_rows,"Columns:",num_cols,"\nYRows",yrows)

ct =ColumnTransformer(transformers=[('encoder',OneHotEncoder(),[0])],remainder="pa
ssthrough",sparse_threshold=0)
```

```
X = np.array(ct.fit_transform(X))

#Performing Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X = sc.fit_transform(X)

X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.1,random_state=0)

#Initialising ANN

ann = tf.keras.models.Sequential()

 #Adding First Hidden Layer

ann.add(tf.keras.layers.Dense(units=30,activation="relu"))

 #Adding Second Hidden Layer

ann.add(tf.keras.layers.Dense(units=30,activation="relu"))

#Adding Output Layer

ann.add(tf.keras.layers.Dense(units=1,activation="sigmoid"))

#Compiling ANN

ann.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])

#Fitting ANN

model = ann.fit(X_train,Y_train,batch_size=32,epochs = 500)

y_pred= ann.predict(X_test)
```

```
y_predtraining = ann.predict(X_train)

y_predbool = y_pred.round(decimals=0, out=None)

y_predtrainingbool = y_predtraining.round(decimals=0, out=None)

print("Accuracy:",metrics.accuracy_score(Y_test, y_predbool))

print("Precision:",metrics.precision_score(Y_test, y_predbool))

print("Recall:",metrics.recall_score(Y_test, y_predbool))

print("Training Accuracy:",metrics.accuracy_score(Y_train, y_predtrainingbool))

print("Training Precision:",metrics.precision_score(Y_train, y_predtrainingbool))

print("Training Recall:",metrics.recall_score(Y_train, y_predtrainingbool))

#LSTM

#Importing the Libraries

import pandas as pd

import numpy as np

%matplotlib inline

import matplotlib. pyplot as plt

import matplotlib

from sklearn.preprocessing import MinMaxScaler

from keras.layers import LSTM, Dense, Dropout

from sklearn.model_selection import TimeSeriesSplit

from sklearn.metrics import mean_squared_error, r2_score
```

```
import matplotlib. dates as mandates

from sklearn.preprocessing import MinMaxScaler

from sklearn import linear_model

from keras.models import Sequential

from keras.layers import Dense

import keras.backend as K

from keras.callbacks import EarlyStopping

from tensorflow.keras.optimizers import RMSprop

from keras.models import load_model

from keras.layers import LSTM

from keras.utils.vis_utils import plot_model

from google.colab import drive

from google.colab import drive

drive.mount('/content/drive')

#Get the Dataset

df=
pd.read_csv('/content/drive/MyDrive/Dataset/StockFinalDataset.csv',index_col='Date',parse_dates=True,infer_datetime_format=True)

df = df.drop(['value change'],axis=1)

df=df.iloc[::-1]

df.head()
```

```
#Set Target Variable

output_var = pd.DataFrame(df['Price'])

#Selecting the Features

features = ['Open', 'High', 'Low', 'Vol full']

scaler = MinMaxScaler()

feature_transform = scaler.fit_transform(df[features])

feature_transform=pd.DataFrame(columns=features,data=feature_transform,index=df.in
dex)

feature_transform.head()

timesplit= TimeSeriesSplit(n_splits=9, test_size=7180, gap=5000)

for train_index, test_index in timesplit.split(feature_transform): X_train, X_test =
feature_transform[:len(train_index)],feature_transform[len(train_index):
(len(train_index)+len(test_index))]

 y_train,y_test =output_var[:len(train_index)].values.ravel(), output_var[len(train_index):
(len(train_index)+len(test_index))].values.ravel()

trainX =np.array(X_train)

testX =np.array(X_test)

X_train = trainX.reshape(X_train.shape[0], 1, X_train.shape[1])

X_test = testX.reshape(X_test.shape[0], 1, X_test.shape[1])

ytraincopy = y_train

#Building the LSTM Model
```

```
lstm = Sequential()

lstm.add(LSTM(32,input_shape=(1,trainX.shape[1]),activation='relu',
return_sequences=False))

lstm.add(Dense(1))

lstm.compile(loss='mean_squared_error',optimizer='Adam',metrics=
['mean_squared_error'])

plot_model(lstm, show_shapes=True, show_layer_names=True)

history=lstm.fit(X_train, y_train, epochs=200, batch_size=256, verbose=1, shuffle=True)

#LSTM Prediction

y_pred= lstm.predict(X_test, verbose=1)

y_test = y_test.reshape( [7180,1] )

Ypredtrain = lstm.predict(X_train)

ytraincopy = ytraincopy.reshape(59692,1)

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

# calculate root mean squared error

MSE_R=mean_squared_error(y_test, y_pred)

print('Mean Square Error: ', MSE_R)

MAE_R=mean_absolute_error(y_test, y_pred)

print('Mean Absoulte Error: ',MAE_R )

RMSE_R=np.sqrt(mean_squared_error(y_test, y_pred))

print('Root Mean Square Error: ',RMSE_R )
```

```
#Linear Regression

import numpy as np #For numerical analysis

import pandas as pd #For reading data stored in various file formats

import matplotlib.pyplot as plt  #For visualizations

import missingno

import sklearn

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn import linear_model

from sklearn.metrics import mean_squared_error, mean_absolute_error

from google.colab import drive

drive.mount('/content/drive')

df = pd.read_csv("/content/drive/MyDrive/Dataset/StockFinalDataset.csv",index_col='D
ate',parse_dates=True,infer_datetime_format=True)

df = df.drop(['value change'],axis=1)

df = df.drop(['Name'],axis=1)

df=df.iloc[::-1]

df.head()

df = df.rename(columns={"Date":"full-date","Vol full":"vol-full"})
```

```
df['Open'] = df['Open'].astype(str).map(lambda x : x.split(' ')[0]).replace('nan' , np.nan).astype(np.float)

df['Open'].astype(np.float)

df['High'] = df['High'].astype(str).map(lambda x : x.split(' ')[0]).replace('nan' , np.nan).astype(np.float)

df['High'].astype(np.float)

df['Low'] = df['Low'].astype(str).map(lambda x : x.split(' ')[0]).replace('nan' , np.nan).astype(np.float)

df['Low'].astype(np.float)

#Set Target Variable

output_var = pd.DataFrame(df['Price'])

#Selecting the Features

features = ['Open', 'High', 'Low', 'vol-full']

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

feature_transform = scaler.fit_transform(df[features])

feature_transform= pd.DataFrame(columns=features, data=feature_transform, index=df.index)

feature_transform.head()



from sklearn.model_selection import TimeSeriesSplit

timesplit= TimeSeriesSplit(n_splits=9, test_size=7180, gap=7180)
```

```
# timesplit= TimeSeriesSplit(n_splits=5,test_size=14360)

for train_index, test_index in timesplit.split(feature_transform):

    X_train, X_test = feature_transform[:len(train_index)], feature_transform[len(train_
index): (len(train_index)+len(test_index))]

    y_train, y_test = output_var[:len(train_index)].values.ravel(), output_var[len(train_i
ndex): (len(train_index)+len(test_index))].values.ravel()

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

from sklearn.linear_model import LinearRegression

regression_model = LinearRegression()

regression_model.fit(X_train, y_train)

y_pred = regression_model.predict(X_test)

PredTrain = regression_model.predict(X_train)

MSE_R=mean_squared_error(y_test, y_pred)

print('Mean Square Error: ', MSE_R)

MAE_R=mean_absolute_error(y_test, y_pred)

print('Mean Absoulte Error: ',MAE_R )

RMSE_R=np.sqrt(mean_squared_error(y_test, y_pred))

print('Root Mean Square Error: ',RMSE_R)

MSE_R=mean_squared_error(y_train, PredTrain)
```

```
print('training Mean Square Error: ', MSE_R)

MAE_R=mean_absolute_error(y_train, PredTrain)

print('training Mean Absoulte Error: ',MAE_R )

RMSE_R=np.sqrt(mean_squared_error(y_train, PredTrain))

print('training Root Mean Square Error: ',RMSE_R )

import matplotlib.pyplot as ploat

import matplotlib

convertedtested = y_test.tolist()

plt.figure(figsize=(20, 10))

plt.plot(y_pred, label='LR Value')

plt.plot(y_test, label='True Value')

plt.locator_params(axis='x', nbins=20)

#plt.rcParams["figure.figsize"] = (6.4,4.8)

plt.title('Prediction by LR')

plt.xlabel('No of Values')

plt.ylabel('Scaled PKR')

plt.legend()

plt.show()
```

Thesis-Plag

ORIGINALITY REPORT

9%	7%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	vitalflux.com Internet Source	2%
2	medium.com Internet Source	1%
3	Submitted to Rochester Institute of Technology Student Paper	<1%
4	Submitted to Ghana Technology University College Student Paper	<1%
5	ir.uitm.edu.my Internet Source	<1%
6	Submitted to Middlesex University Student Paper	<1%
7	github.com Internet Source	<1%
8	iiste.org Internet Source	<1%
9	Submitted to South Bank University	