PERSONAL ASSISTANT FOR STOCK MARKET PREDICTION

By

SHAIVAL RAJAN SHAH
15BCE110
KAUSHAL THAKKAR
15BCE125

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER ENGINEERING
Ahmedabad 382481

PERSONAL ASSISTANT FOR STOCK MARKET PREDICTION

MINI PROJECT - III

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Engineering

By

SHAIVAL RAJAN SHAH
15BCE110
KAUSHAL THAKKAR
15BCE125

Guided By
DR. K.P AGRAWAL
DEPARTMENT OF COMPUTER ENGINEERING

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the project entitled "**PERSONAL ASSISTANT FOR STOCK MARKET PREDICTION**" submitted by **SHAIVAL RAJAN SHAH (15BCE110), KAUSHAL THAKKAR (15BCE125)**, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Dr. K.P Agrawal
Assistant professor
Department of Computer Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Sanjay Garg
Dept. of Computer Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

On the very outset of this report, we take this opportunity to express our profound gratitude and heartfelt obligation towards our guide; **Dr. K.P. Agrawal**, who helped us in this endeavor and took keen interest on our project topic "**PERSONAL ASSISTANT FOR STOCK MARKET PREDICTION**". Without his active guidance, constant support, and encouragement we would not able to complete this project. We extend our gratitude to Nirma University for giving us this Opportunity. At last but not the least we give special thanks to our friends who helped us in this project.

ABSTRACT

Prediction of the stock market trend is challenging task as there is lots of uncertainty in the stock market. In this report we have mentioned various algorithms for predicting the price of the stock. For that we have taken 10 technical parameters into consideration. We have used two algorithms one is Support Vector Machine and other is Artificial Neural Network. In the Support Vector Machine, we have used two types of kernel function. First is polynomial kernel and other is Redial Basis Function. In the Artificial Neural Network, we have used 3 hidden layers and Sigmoid function in the output layer. The accuracy of both the model is evaluated. For prediction we have used 10 years of historical data of the yahoo stock.

CONTENTS

LIST OF FIGURES

Chapter 1: Introduction

1.1 **Introduction to the project**

Predicting stock price trend and its movement has been viewed as a standout amongst the most difficult utilizations of time arrangement expectation. Despite the fact that there has been numerous research which manage the issues of anticipating stock price trend, most exact discoveries are related with the developed financial markets. But it is difficult to predict the trend or price of the stock because of the uncertainty in the stock market. There are two types of analysis, Fundamental analysis and technical analysis. In fundamental analysis, performance of the company, economic factors and political factors are considered. In technical factors, previous n days closing price, highest price, lowest price etc. are considered. We can predict the trend of stock or price of the stock using technical analysis. Fundamental analysis is hard to measure and hard to implement in computer language. Technical analysis does not measure the intrinsic security value of the stock, but it uses technical stock charts to predict the trend of the stock.

In initial stage of the study of the stock market prediction, classical methods were used. But as stock market is a non-stationary time series of data. It was not so effective. So non-linear machine learning techniques like Artificial neural networks (ANN) and Support Vector Machine(SVM) are used widely. In this project we have used both the techniques to predict the trend of the stock and measured the accuracy of both the techniques.

1.2 **Literature Survey**

Nowadays, there have been lots of studies going on to predict the trend and movement of the stock market. In academic tremendous work has been done on this subject. In this section we see reviews of the previous studies on Stock Market Prediction.

Stock market trend prediction is done in Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques (by Jigar Patel, Sahil Shah, Priyank Thakkar , K Kotecha)(2016), they have taken Simple 10 Day moving Average ,Weighted 10 Day Moving Average, Momentum, Stochastic K%, Stochastic D%, Relative strength Index, Moving Average convergence divergence, Larry William's R%, A/D (Accumulation/Distribution) Oscillator, CCI (Commodity Channel Index) as features and used SVM and ANN. And in SVM using Polynomial function they have got 84% accuracy and using RBF kernel function they have got 80% accuracy. In average they got 78.71 % accuracy in SVM and 74.94% accuracy using Artificial Neural Networks.

In Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange (by Yakup Kara, Melek Acar Boyacioglu, Ömer Kaan Baykan) they used Istanbul Stock Exchange to predict the trend of the stock price. In that they have used ANN and polynomial SVM. Using ANN they have got 75.74 % accuracy and using polynomial SVM 71.52 %. In the polynomial SVM the degree of polynomial is 3 and value of c is 100.

1.3 **Scope of this work**

Stock market investment is challenging tasks. There is higher probability of losing money but also gain lots of money from it. So there should be one model which gives the gist of trends in stock market. So using machine learning techniques, we can predict the trend of the stock market and invest in it according to that.

Chapter 2: Proposed Methodology

2.1 **Introduction of**

In our project we have used to SVM and ANN as a classifier for learning the trend of stocks of next day. Both are compared on the bases of accuracy. We are working on yahoo stocks which has data of 5 years. We have calculated 11 features. We are using tensorflow and keras library for machine learning models. Artificial neural network created is based on keras. We are training my model in such a way that the trend of the next day is taken as target value for current day. So our model will predict the trend for next day. We are focusing more on calculating the and finding features which affects the stock prices. Right now i have only calculated 11 features using moving average for 10 days. We have selected these features on the basis of literature survey of research papers. We have taken yahoo stocks for prediction.

2.2 **Features or Technical Indicators**

Features calculated are mentioned below.

- Simple 10 days moving average
- Weighted 10 days moving average
- Momentum
- Stochastic K%
- Stochastic D%
- Relative strength index (RSI)
- Moving average convergence divergence (MACD)
- Larry William's R%
- A/D (Accumulation/Distribution) oscillator
- CCI (Commodity channel index)
- Month index

Name of indicators	Formulas
Simple $n(10 here)$-day Moving Average	$\frac{C_t+C_{t-1}+\ldots+C_{t-9}}{n}$
Weighted $n(10 here)$-day Moving Average	$\frac{(10)C_t+(9)C_{t-1}+\ldots+C_{t-9}}{n+(n-1)+\ldots+1}$
Momentum	$C_t - C_{t-9}$
Stochastic $K\%$	$\frac{C_t-LL_{t-(n-1)}}{HH_{t-(n-1)}-LL_{t-(n-1)}} \times 100$
Stochastic $D\%$	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10}\%$
Relative Strength Index (RSI)	$100 - \frac{100}{1+\left(\sum_{i=0}^{n-1} UP_{t-i}/n\right)/\left(\sum_{i=0}^{n-1} DW_{t-i}/n\right)}$
Moving Average Convergence Divergence (MACD)	$MACD(n)_{t-1} + \frac{2}{n+1} \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} \times 100$
A/D (Accumulation/Distribution) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t is the low price and H_t the high price at time t, $DIFF_t = EMA(12)_t - EMA(26)_t$, EMA is exponential moving average, $EMA(k)_t = EMA(k)_{t-1} + \alpha \times (C_t - EMA(k)_{t-1})$, α is a smoothing factor which is equal to $\frac{2}{k+1}$, k is the time period of k-day exponential moving average, LL_t and HH_t implies lowest low and highest high in the last t days, respectively. $M_t = \frac{H_t+L_t+C_t}{3}$, $SM_t = \frac{\left(\sum_{i=1}^{n} M_{t-i+1}\right)}{n}$, $D_t = \frac{\left(\sum_{i=1}^{n} |M_{t-i+1} - SM_t|\right)}{n}$, UP_t means upward price change while DW_t is the downward price change at time t.

Fig 1. Features with formulas

2.3 **Flow of our project**

1. Start
2. Read the dataset
3. Define features and label
4. Encode the dependent variable
5. Divide the dataset into two parts for training and testing
6. TensorFlow data structure for holding features, labels etc.
7. Implement the model
8. Train the model
9. Reduce MSE
10. Make prediction on the test data
11. End

Chapter 3: Artificial Neural Network

3.1 **Description**

Artificial neural networks are one of the primary instruments utilized as a part of machine learning. As the "neural" some portion of their name recommends, they are cerebrum enlivened frameworks which are expected to duplicate the way that we people learn. Neural networks comprise of information and yield layers, and (much of the time) a hidden layer comprising of units that change the contribution to something that the output layer can utilize. They are brilliant instruments for discovering designs which are unreasonably unpredictable or various for a human software engineer to concentrate and instruct the machine to perceive.

While neural networks (likewise called "perceptron") have been around since the 1940s, it is just over the most recent a very long while where they have turned into a noteworthy piece of artificial intelligence. This is because of the entry of a strategy called "backpropagation," which enables networks to change their concealed layers of neurons in circumstances where the result doesn't coordinate what the maker is seeking after — like a network intended to recognize dog, which misidentifies a cat, for instance.

Another vital progress has been the landing of deep learning neural networks, in which distinctive layers of a multilayer network extricate diverse features until the point when it can recognize what it is searching for.

3.2 **Working**

For an essential thought of how a deep learning neural network learns, imagine a manufacturing plant line. After the crude materials (the informational collection) are input, they are then passed down the transport line, with each subsequent stop or layer separating an alternate arrangement of abnormal state highlights. If the network is expected to recognize an object, the brightness of pixel may be analyzed by first layer.

The following layer could then identify any edges in the picture, based on lines of comparative pixels. After this, another layer may recognize surfaces and shapes, etc. When the fourth or fifth layer is achieved, the deep learning net will have made complex component indicators. It can make sense of that specific picture components, (for example, a couple of eyes, a nose, and a mouth) are regularly discovered together.

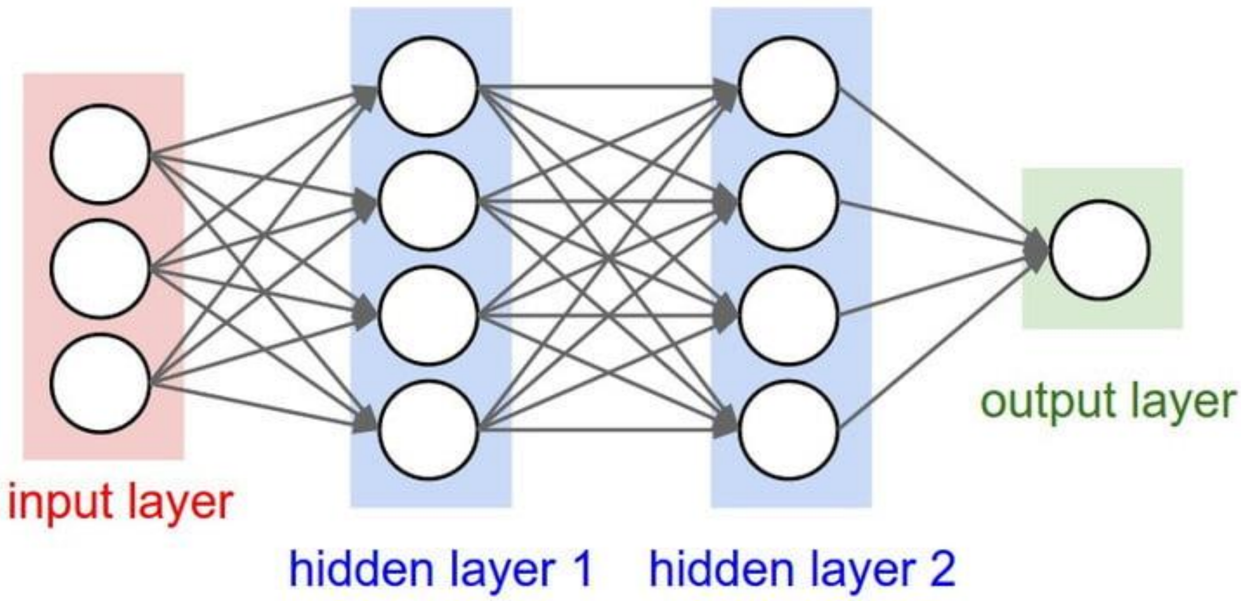

Fig 2. Artificial Neural Network

When this is done, the analysts who have prepared the network can give labels to the output, and afterward utilize backpropagation to redress any oversights which have been made. Sooner or later, the network can complete its own characterization assignments without requiring people to help unfailingly.

3.3 **Use of ANN in our project**

It has 3 layers.
2 layers uses relu activation function and last layer uses sigmoid.
Backpropagation using GradientDescentOptimizer is used for reducing error.
Main reason behind this is relu function creates more sparse data when W*X + B < 0 and other reason is it reduces chances of the gradient to vanish. Using relu function faster learning is done.
taking sigmoid or relu activation function in output layer i.e. last layer gives almost same accuracy, so we have taken sigmoid function to obtain nonlinear model. Also, we have normalized the dataset.
The accuracy got in ANN model is 71.73%

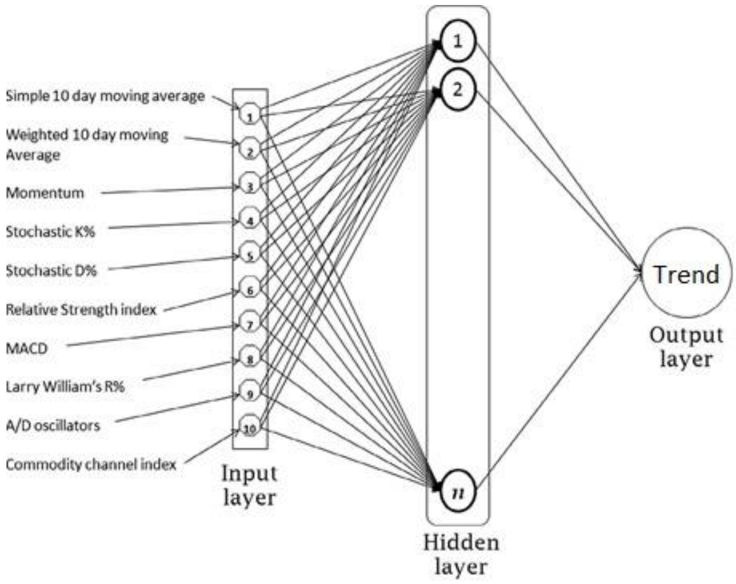

Fig 3. ANN model

3.3.1 **Backpropagation**

Backpropagation, another way to say "backward propagation of errors," is a calculation for supervised learning of artificial neural networks utilizing gradient descent. Given an artificial neural network and a error function, the strategy ascertains the gradient of the error function regarding the neural network's weights. It is a speculation of the delta manage for perceptrons to multilayer feedforward neural networks.

The "backwards" some portion of the name originates from the way that computation of the gradient continues backwards through the network, with the gradient of the last layer of weights being ascertained first and the gradient of the principal layer of weights being figured last. Fractional calculations of the gradient from one layer are reused in the calculation of the gradient for the past layer. This backwards stream of the mistake data takes into account productive calculation of the gradient at each layer versus the innocent approach of computing the gradient of each layer independently.

Backpropagation's prevalence has encountered a current resurgence given the boundless appropriation of deep neural networks for picture acknowledgment and discourse acknowledgment. It is viewed as an effective calculation, and current executions exploit particular GPUs to additionally enhance execution.

DEFINITION

1) **Dataset** consisting of input-output pairs $(\vec{x_i}, \vec{y_i})$, where $\vec{x_i}$ is the input and $\vec{y_i}$ is the desired output of the network on input $\vec{x_i}$. The set of input-output pairs of size N is denoted $X = \left\{(\vec{x_1}, \vec{y_1}), \dots, (\vec{x_N}, \vec{y_N})\right\}$

2) A **feedforward neural network**, as formally defined in the article concerning feedforward neural networks, whose parameters are collectively denoted θ. In backpropagation, the parameters of primary interest are w_{ij}^k, the weight between node j in layer l_k and node i in layer l_{k-1}, and b_i^k, the bias for node i in layer l_k. There are no connections between nodes in the same layer and layers are fully connected.

3) An **error function**, $E(X, \theta)$, which defines the error between the desired output $\vec{y_i}$ and the calculated output $\hat{\vec{y_i}}$ of the neural network on input $\vec{x_i}$ for a set of input-output pairs $(\vec{x_i}, \vec{y_i}) \in X$ and a particular value of the parameters θ.

Training a neural network with gradient descent requires the calculation of the gradient of the error function $E(X, \theta)$ with respect to the weights w_{ij}^k and biases b_i^k. Then, according to the learning rate α, each iteration of gradient descent updates the weights and biases (collectively denoted θ) according to

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta},$$

Chapter 4: Support Vector Machine

4.1 **Description**

Support Vector Machine(SVM) is a supervised learning model that is introduced by Vapnik. It is used for classification as well as regression. For regression Support Vector Regressor(SVR) is used. The main objective of the SVM model is to separate data points as far as possible from a hyperplane. Hyperplane is a plane that divides the data points according to their labels. In addition to linear classification, SVM can classify the non-linear data using kernel trick that maps the data into high dimensional space and transoms it into linear separable data. In general term support Vector Machine defines a hyperplane in high dimension which can be used to classify or dived the data points. The main kernel functions are polynomial kernel, Gaussian kernel, Radial Basis Function(RBF), Linear kernel and Sigmoid kernel.

4.2 **Working**

For binary classification, suppose x_i be the input vectors and y_i be the corresponding output labels. Here for binary classification y_i ε {1,-1}. SVM maps the input vector into more dimensional space ϕ(x_i) and make a hyperplane with maximized distance between two classes. Mapping is done by a kernel function K (x_i, x_j). And the final classifier is given in the equation below.

$$f(x) = sgn\left(\sum_{i=1}^{N} y_i \alpha_i \cdot K(x, x_i) + b\right)$$

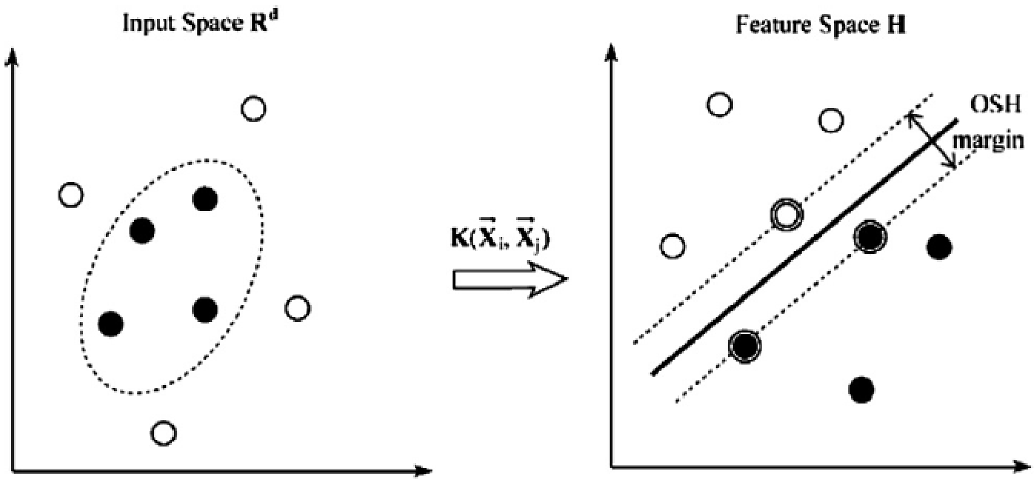

Fig 4. Support vector machine

$$\text{Maximize} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \cdot y_i y_j \cdot K(x_i, x_j)$$

$$\text{subject to } 0 \leqslant \alpha_i \leqslant c$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad i = 1, 2, \ldots, N$$

Where c is a regularization perimeter, which suggest the trad off between distance of hyperplane from the classes and misclassification error. Larger value c, smaller margin hyperplane and less misclassification and smaller the value of c, larger margin hyper plane but they may misclassify the bounty points.
Some of the kernel functions are
Polynomial Function: $K(x_i, x_j) = (x_i * x_j + 1)^d$
Radial Basis Function: $K(x_i, x_j) = \exp(-\Upsilon ||(x_i - x_j)||^2)$
Here d is degree of the polynomial function and ϒ is a constant.

4.2.1 **Radial Basis Function**

Radial Basis Function measures hoe far is the input vector(x) from the hyperplane vector (μ).

$$\phi_j(\boldsymbol{x}) = exp\{-\frac{||\boldsymbol{x} - \boldsymbol{\mu}_j||^2}{2\sigma_j^2}\}$$

Here x is the input vector and μ and σ represent the centre and spread of dataset points. To adjust the μ and σ below formula is used.

$$\mu_{Class_k} = \frac{1}{|Class_k|} \sum_{x^{(n)} \in Class_k} x^{(n)}$$

$$\sigma^2_{Class_k} = \frac{1}{|Class_k|} \sum_{x^{(n)} \in Class_k} ||x^{(n)} - \mu_k||^2$$

4.3 **Use of SVM in our project**

NO.	**Kernel Function**	**d**	**C**	**Avg. Accuracy**
1	Polynomial	2	100	52.79%

2	Polynomial	3	100	52.79%
3	Radial Basis	-	100	51.52%

Chapter 5: Summary and Conclusion

5.1 Summary

In this project we have implemented two different models for predicting stock market price of yahoo stock for next day. SVM and ANN is compared, and it is found that ANN is good choice over SVM in terms of accuracy and SVM is faster than ANN. As stock market fluctuates with numerous factors exact prediction of trend is next to impossible.

5.2 Conclusion

The accuracy is the major issue in stock market prediction because stock prices can go up and down with numerous unexpected activities. So only technical indicators are not sufficient to predict exact trend. Stock prices fluctuates by political news, company performance, CEO statements. And various other factors which affects the price. For this future work can be done on sentimental analysis which classifies these factors has very good, good, neutral, bad, worse which can be given weight and helps to predict accurate trend. Also target time is important for prediction as there is similar trend in particular period in a year. So as mentioned there is lot of scope of stock prediction in future and lots of work has to be done in this field.

Comparison:

No.	Algorithm used	Avg. Accuracy
1	Support Vector Machine	52.79%
2	Artificial Neural Networks	71.73%

References

- Predicting stock market index using fusion of machine learning techniques Jigar Patel, Sahil Shah, Priyank Thakkar , K Kotecha
- Aldin, M. M., Dehnavr, H. D., & Entezari, S. (2012). Evaluating the employment of technical indicators in predicting stock price index variations using artificial neural networks (case study: Tehran stock exchange). International Journal of Business & Management, 7.
- Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. Knowledge-Based Systems, 35, 245–258.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity.
- Journal of Econometrics, 31, 307–327.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.
- Cheng, C., Xu, W., & Wang, J. (2012). A comparison of ensemble methods in financial market prediction. In 2012 Fifth international joint conference on computational sciences and optimization (CSO) (pp. 755–759). IEEE.
- Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines.
- Hadavandi, E., Ghanbari, A., & Abbasian-Naghneh, S. (2010a). Developing an evolutionary neural network model for stock index forecasting. In Advanced intelligent computing theories and applications (pp. 407–415). Springer.
- Hadavandi, E., Shavandi, H., & Ghanbari, A. (2010b). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. Knowledge-Based Systems, 23, 800–808.
- Han, J., Kamber, M., & Pei, J. (2006). Data mining: Concepts and techniques. Morgan Kaufmann.
- Hsieh, D. A. (1991). Chaos and nonlinear dynamics: Application to financial markets. The Journal of Finance, 46, 1839–1877.
- Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid SOFM–SVR with a filter-based feature selection for stock market forecasting. Expert Systems with Applications, 36, 1529–1539.

Appendix A – list of useful websites

https://medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877

https://www.quantinsti.com/blog/machine-learning-trading-predict-stock-prices-regression/

https://towardsdatascience.com/stock-prediction-in-python-b66555171a2