Stock Trend Prediction using Machine Learning Techniques.

Nirzar Panvelkar
Computer Engineering
Don Bosco College of Engineering
Goa,India
nirzarpanvelkar38@gmail.com

Siddesh Kamat
Computer Engineering
Don Bosco College of Engineering
Goa,India
siddeshkamat40@gmail.com

Kashyap Chodankar
Computer Engineering
Don Bosco College of Engineering
Goa,India
chodankarkashyap@gmail.com

Akhilesh Saraf
Computer Engineering
Don Bosco College of Engineering
Goa,India
sarafakhilesh30@gmail.com

Abstract—Finance is one of the main pillars on which an economy grows. Having a precise understanding of how the financial market works corroborates greater monetary returns. Stock Markets are established for the general public to own a piece of a company and state an ownership whereas the company gains funds from the public which are used for the expansion of the business and return profits to the investor based on the amount of investment. However choosing which company's stock to buy is the most challenging task. Exhaustive research and study about the company, their way of working, history, their place in the market, their holdings, projects, etc is required to get an idea about their present performance and future. However no one can predict how a company will perform in the future but an estimation can be achieved by studying various parameters. Machine Learning and Deep Learning techniques have proven their mettle in analysing and prediction data with state of the art results. Though without any intention to predict the exact future of a company, these techniques can be extremely helpful in getting an estimation which bolster our choice rather than just trusting our intuition.

Keywords— Finance, LSTM, KNN, Stock Market

I. INTRODUCTION

The stock market is a club of buyers and sellers of stocks signifying ownership of a particular business or a company. A stock can be classified on various stages, and the trading of stocks is done accordingly. Some stocks can be listed in multiple stock exchanges pasturing across countries to make the stock bewitching to investors internationally. Investment in the stock market has yielded implausible returns in bounded period of time but there are also grave risks. However a successful prediction of a stock's future price could strengthen the force of return. Initially, the future price could be estimated with fundamental analysis which follows a top down approach [5]. It scans the global economy, then the country analysis followed by the sector performance and then the company evaluation. With the advent of technological advancements, it is possible to analyse the trends of the market with various statistical tools which would otherwise be stringy. Now one of the most important factors while considering to invest in a company's stocks is the company's Price-To-Earnings Ratio. The dependency of a stock price on various factors is unstable and it is never possible to predict the exact price of the stock. But with the utilization of technologies like Machine
Learning and Deep Learning, it is possible to develop an algorithm to minimise this window of error and get the most accurate output possible with the help of as much data at our disposal [12]. Machine Learning is mathematical technique of finding patterns in the data provided and applying what is learnt on a new set of data and making predictions. It is an art of making the machine learn new patterns without explicitly programming the machine to do so. The basic process in Machine Learning from data gathering to obtaining predictions can be summarised as follows: initially, the user analyses the data and selects the input format which is preferred. Then, the user has to decide which machine learning algorithm will best suit the use case. There are various mathematical algorithms best suited for a particular type of problem and the input. Then the user makes an analytical model based on the chosen mathematical algorithm. Later, the user trains the model that he prepared using the input data chosen. Finally, the model trained is tested on other data for finding patterns, making predictions, to generate scores and other findings [1,11,14]. Machine Learning can be used to improve the working of the traditional models due to

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

its fast processing and pattern analysing proficiency [3,6-10].

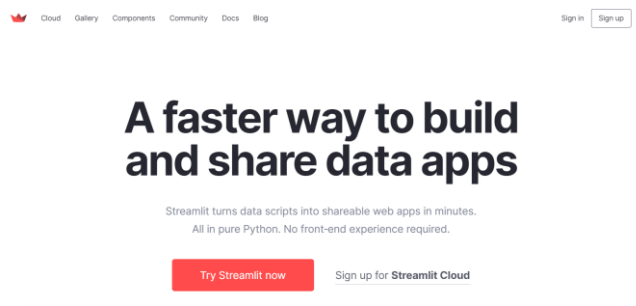
Stock trends vary on various factors be it financial, social or even due to natural calamities. They directly impact the stock price and all these patters need to be learnt by the predicting model accurately. LSTM are used to boost this memory. KNN is best suited for sequential data as they have a recurrent architecture where the output of first iteration again goes into the model again as input.

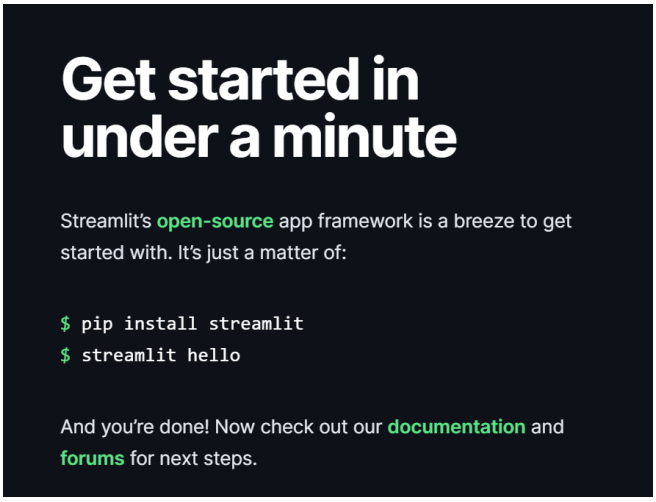
II. User Interface

A. STREAMLIT

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc. With Streamlit, no callbacks are needed since widgets are treated as variables. Data caching simplifies and speeds up computation pipelines. Streamlit watches for changes on updates of the linked Git repository and the application will be deployed automatically in the shared link.





B. DATA USED

We used the raw data from yahoo Finance, Yahoo Finance is a media platform that provides financial news, data about stock quotes, press releases, and financial reports. And all the data provided by Yahoo Finance is free. Yahoo Finance API is the API that Yahoo provides to fetch financial information.

Yahoo deprecated their Finance API in 2017. So you can see many websites talking about alternatives for Yahoo Finance API. However, the python library yfinance offers a temporary fix to the problem by scraping the data from Yahoo! Finance and returning the data in the DataFrame format. So you can still use Yahoo Finance to get free stock market data

The Yahoo Finance API provides access to the information about:

- finance summaries like earnings, balance sheet.
- stocks historical prices.
- stock actions (including splits and dividends).

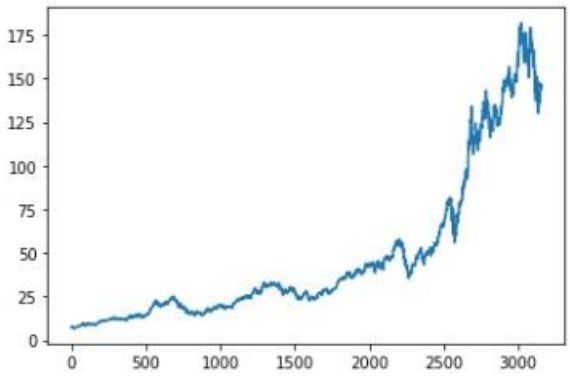
All this information is from the Yahoo Finance official website.

```
# To install yfinance before you use it.
> pip install yfinance
```

Date	Open	High	Low	Close*	Adj Close**	Volume
Jul 14, 2022	16,018.85	16,070.85	15,858.20	15,938.65	15,938.65	-
Jul 13, 2022	16,128.20	16,140.00	15,950.15	15,966.65	15,966.65	233,300
Jul 12, 2022	16,126.20	16,158.75	16,031.15	16,058.30	16,058.30	208,600
Jul 11, 2022	16,136.15	16,248.55	16,115.50	16,216.00	16,216.00	255,900
Jul 08, 2022	16,273.65	16,275.50	16,157.90	16,220.60	16,220.60	281,100
Jul 07, 2022	16,113.75	16,150.50	16,045.95	16,132.90	16,132.90	264,600
Jul 06, 2022	15,818.20	16,011.35	15,800.90	15,989.80	15,989.80	288,400
Jul 05, 2022	15,909.15	16,025.75	15,785.45	15,810.85	15,810.85	254,200
Jul 04, 2022	15,710.50	15,852.35	15,661.80	15,835.35	15,835.35	304,300
Jul 01, 2022	15,703.70	15,793.95	15,511.05	15,752.05	15,752.05	364,100
Jun 30, 2022	15,774.50	15,890.00	15,728.85	15,780.25	15,780.25	306,000

The following graph shows us the trend in Stock price of Apple, using the data from yahoo finance.

```
start = '2010-01-01'
end = date.today()
df=data.DataReader('AAPL','yahoo', start, end)
```



C. Fitting the ML Models for prediction

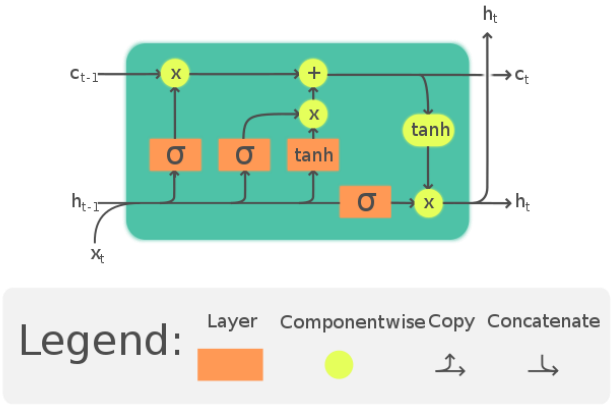
The dataset is then divided into training and testing sets after being cleaned and resampled. Despite the fact that there are a number of ways for splitting the data, we chose to utilise sklearn's train test split() because it divides the data into two random partitions with a ratio of 0.70:0.30 and is timeeffective when working with such a large dataset. We selected to use the specialised classification approaches of LSTM, XgBoost, KNN, and Prophet to predict the target result in the testing set.

D. Methods or Models used.

Long-Short Terms Memories Networks (LSTM):

Niu [20] stated in their paper that the RNN is the best method to use for the time-series data due to its ability to capture long and short terms dependencies in time-series. According to Lei [21] this is a policy-based algorithm trading model. LSTM is a special type of RNN, Wang [14] introduce the concept of LSTM networks to learn the sequential pattern. LSTM is incorporated into the Recursive Neural Networks' hidden layers due to its ability to memorise the data. This is a special Neural Networks which can store and access a bigger range of sequential contextual input information and perform very well at handling vanishing gradient issues. The cell is comprised of input gate, forget gate and output gate, when the input data enter the LSTM networks, the cell structure determined which information can stay and which can be forgotten in the forget gate according to specific rules.

The problems of vanishing gradient when the weights assigned were significantly less and the inaccuracy of long term dependencies were solved with the help of Long Short Term Memory. These are a special type of RNN where they can easily handle long term dependencies as a result of the large memory. Similar to RNN, they too have a chain structure but the repeating loop mechanism is complicated and has a number of gates to handle data in a more efficient way. They have an input gate, a forget gate and an output gate [2]. When new information is added into a RNN model, it applies an activation function and modifies it completely rather than considering it as an integral part in predicting something in the distant future. There is no tagging of the output as 'important' or 'not important' for referring in the future. The LSTM on the other hand does not directly modify the input using a function but uses mathematical transformations. There are cell states between which the input is modified when data goes from one cell state to the other. Hence, the model can remember distinctly what the data was at a particular cell state thus determining what to remember and what to not.



LSTM architecture. xt is current input; ht-1 is previous state and ht is current output state.

E. XgBoost

- XGBoost is a supervised learning algorithm based on ensemble trees. It aims at optimising a cost objective function composed of a loss function (d) and a regularization term (β):
- Ω(θ) = Xn i=1 d(yi, ybi) | {z } Loss + XK k=1 β(fk) | {z } regularization ,
- where ybi is the predictive value, n the number of instances in the training set, K is the number of trees to be generated and fk is a tree from the ensemble trees. The regularization term is defined as:
- β(ft) = γT + 1 2 " α XT j=1 | cj | +λ XT j=1 c 2 j
- where γ is the minimum split loss reduction, λ is a regularization term on the weight and c is the weight associated to each leaf. Let ft(xi) = cq(xi) , where q is in [1,T], where T is the number of leafs. A greedy approach is performed to select the split that increases the most the gain. The detailed procedures and equation's derivations are given in Appendix A. Table III outlines the ten XGBoost key parameters, ranges and default values of each parameter.
- XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting

framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

F. KNN

The K-Nearest-Neighbors (KNN) is a nonparametric classification algorithm, i.e. it does not make any presumptions on the elementary dataset. It is known for it's simplicity and effectiveness. It is a supervised learning algorithm. A labeled training dataset is provided where the data points are categorized into various classes, so that the class of the unlabeled data can be predicted. In Classification, different characteristics determine the class to which the unlabeled data belongs. KNN is mostly used as a classifier. It is used to classify data based on closest or neighboring training examples in a given region. This method is used for its simplicity of execution and low computation time. For continuous data, it uses the Euclidean distance to calculate its nearest neighbors. For a new input, the K nearest neighbors are calculated and the majority among the neighboring data decides the classification for the new input. Even though this classifier is simple, the value of 'K' plays an important role in classifying the unlabeled data.

There are many ways to decide the values for 'K', but we can simply run the classifier multiple times with different values to see which value gives the most effective result. The computation cost is slightly high because all the calculations are made when the training data is being classified, not when it is encountered in the dataset. It is a lazy learning algorithm as not much is done when the dataset is being trained except storing the training data and memorizing the dataset instead.

It does not perform generalization on the training dataset. So the entire fundamental dataset being trained is required when in the testing stage. In regression, KNN predicts continuous values. This value is the average of the values of its K - nearest neighbors.

KNN is used in datasets where data is separated into different clusters so that the class of the new input can be determined. KNN is more significant for a study where there is no previous knowledge about the data being used.

A. WORKING

k-NN is a classification algorithm. Mainly there are two steps in classification: 1. Learning Step: Using the training data a classifier is constructed. 2. Assessment of the classifier. According to the nearest neighbor technique, the new unlabeled data is classified by determining which classes its neighbors belong to. KNN algorithm utilizes this concept in its calculation. In the case of KNN algorithm, a particular value of K is fixed which helps us in classifying the unknown tuple. When a new unlabeled tuple is encountered in the dataset, KNN performs two operations. First, it analyzes the K points closest to the new data point, i.e. the K nearest neighbors. Second, using the neighbors' classes, KNN determines as to which class the new data should be classified into. Fig. 1 shows a simple K-NN structure.

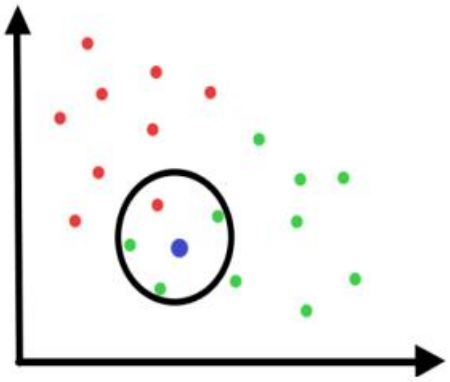


Fig. 1. A simple KNN

After we gather K-Nearest Neighbours, we simply take the majority of them to predict the class of the training example. The factors that affect the performance of KNN are the value of K, the Euclidean distance and the normalisation of the parameters. To understand the detailed working of the algorithm, the steps are as follows:
Given the training dataset :
{ (x(1), y(1)) , (x(2), y(2)), …… , (x(m), y(m))
}
Step1: Store the training set
Step2: For each new unlabeled data,
Calculate Euclidean distance with all training data points using the formula:
$\sqrt{\sum}(xi - yi)^2$

Find the k- nearest neighbours
Assign class containing the maximum number of nearest neighbours.
After storing the training, set all parameters must be normalized, so that the calculations become easier. The result of the classification is sensitive to the value of 'K'.The input variable 'K' decides the number of neighbours that must be considered. The value of 'K' effects the algorithm as using the 'K' value we can build the boundaries of each class. The best value of K is chosen by first examining the data. Larger values of K are more precise as they reduce the net noise but this is not

guaranteed. A good value of K can also be determined using cross-validation.
If K=1, then the data is simply allocated to the class of its nearest neighbour. At K=1, the error rate is consistently zero for the training data. This happens because the nearest point to any training data point is itself. Hence the best results are obtained if the value of K=1. But with K=1, the boundaries are over-fitted. In the case of very small values of 'k' the algorithm is too sensitive to noise.

G. FB- Prophet

Time series analysis of data is useful to get meaningful statistics and other properties of data in the business environment. The time series forecasting model has an important usage in the forecasting model where time plays an important role. This forecasting model has a great impact on determining future sales and managing businesses. It is also important because many predictions involve time-related components that need to be handled carefully to do the prediction when the actual result is unknown. To determine the root cause of a certain event it is required to know the pattern of related data and their time. There are four major components-. • Level- It is the base value used in the time-series data. • Trend-It is shown as a curve that may increase or decrease depending on time. • Seasonability- This is indicated as a cycle or pattern over time. • Noise- It shows the variation in the observed data. Facebook has introduced an open-source forecasting tool FB prophet available to use in python and R programming languages as a library.The FB prophet is developed to meet the forecasting need from the business point of view. It has the following characteristics- • Time series data observed on an hourly, daily, and monthly basis for a year or more. • It takes care of holiday or break intervals that are known in advance. • It takes care of trends, outlier detection, missing data, etc. At its core, the Prophet works on an additive regression model (research.fb.com) with the following trends- • Modular regressive or liner curve for the growing trend. • A Fourier series based seasonal component • A seasonal component every week. • The user suggested a list of break intervals or holidays.

Fig. 1 shows the basic Prophet workflow (Forecasting at Scale, Sean J. Taylor and Benjamin Letham, 2017), it has the sweetest part where the surface problem is being automated and analysts have to inspect the forecasts.

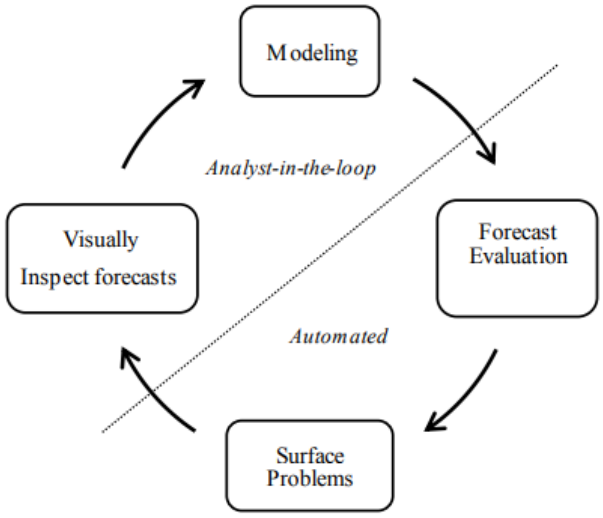
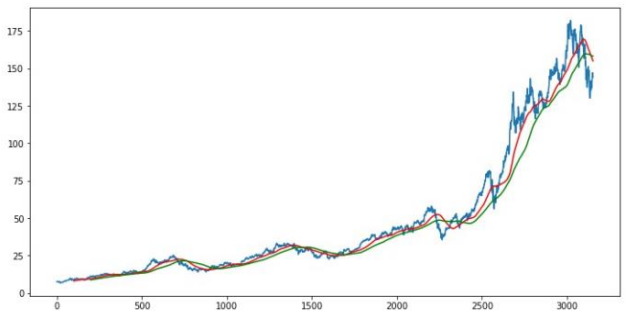


Fig. 1: Prophet workflow

The FB Prophet is based on the curve fitting technique in the Bayesian model. It has easily understandable parameters and also it doesn't require much time-series data to do prediction. The technique is most suitable when the timeseries data has strong seasonal attributes as influencing factors. It also takes care of planned breaks or holidays in the continuous data. FB Prophet deals better in case of missing data, variation in trends, and outlier detection. In real-world scenarios such as sales prediction, such variations need to be addressed. It also has easily usable and interpretable libraries.

H. Comparison table

One way to assess how well a regression model fits a dataset is to calculate the root mean square error, which tells us the average distance between the predicted values from the model and the actual values in the dataset.



The formula to find the root mean square error, often abbreviated RMSE, is as follows:

$$RMSE = \sqrt{\Sigma (Pi - Oi)2 / n}$$

where:

- Σ is a fancy symbol that means “sum”
- Pi is the predicted value for the ith observation in the dataset
- Oi is the observed value for the ith observation in the dataset
- n is the sample size

	RMSE
XgBoost	0.54801572
KNN	5.427252109
Prophet	1.884439647

i. Conclusion

Finance is one of the arenas where trades of billions of dollars are made globally on a daily basis. Studying such a colossal market with the help of machine learning techniques can give insights about how the financial market works. The financial data about stocks was retrieved from yahoo finance, model was trained on this data for various stocks and then compared with different models.

Models like KNN, LSTM and XgBoost and Prophet is widely known among every financial market analysts. These neural networks models were proved to have better performance and prediction accuracy. The demonstration of the output on different charts shows the prediction resembles the movement of the target stock with a very low error rate. There are a few limitations that are needed to be taken into consideration, the risk aspect were not considered among the research papers

In the future, more data need to be collected from more sources need to be collected where different models in both Machine Learning and Deep Learning are compared. Moreover, with the increase in the advancement of technologies, Deep Learning might become the most favourable method to use when it comes to this domain, future research can focus on exploring other Deep Learning methodologies that were not mentioned in this research paper.

Additionally, sentiment analysis and investor behaviour analysis need to be added in the pre-processing stage of the architecture to produce better input data for the model. Finally, risk analyse also needed to be considered as all investment come with risks, by this, the accuracy will significantly increase while risks are minimizing.

References

[1] D. P. Gandhmal and K. Kumar, "Systematic analysis and review of stock market prediction techniques," Computer Science Review, vol. 34, 2019, doi: 10.1016/j.cosrev.2019.08.001.

[2] Y. Karaca, Y.-D. Zhang, and K. Muhammad, "Characterizing Complexity and Self-Similarity Based on Fractal and Entropy Analyses for Stock Market Forecast Modelling," Expert Systems with Applications, vol. 144, 2020, doi: 10.1016/j.eswa.2019.113098.

[3] A. Dainotti, A. Pescape, and K. Claffy. Issues and future directions in traffic classification. IEEE network, 26(1), 2012

[4] D. Wettschereck and D. Thomas G., “Locally adaptive nearest neighbour algorithms,” Adv. Neural Inf. Process. Syst., pg. 184–186, 1994.

[5] Han EH.., Karypis G., Kumar V. (2001) “Text Categorization Using Weight Adjusted k-Nearest Neighbour Classification”. In: Cheung D., Williams G.J., Li Q. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2001. Lecture Notes in Computer Science, vol 2035. Springer, Berlin, Heidelberg.

[6] Shengyi Jiang,Guansong Pang,Meiling Wu,Limin Kuang, “An improved K-nearest-neighbour algorithm for text categorization”,Expert systems with Applications,Elsevier(2012)

[7] Zunic, Emir &Korjenić, Kemal &Hodžić, Kerim&Donko, and Dzenana.,“Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data”, International Journal of Computer Science and Information TechnologyInternational Journal of Computer Science & Information Technology (IJCSIT), Vol 12, No 2, PP. 23-36, April 2020.

[8] Chen Guo, Quanbo Ge, HaoyuJiang,Gang Yao and Qiang Hua, “Maximum Power Demand Prediction UsingFbprophetwith Adaptive Kalman Filtering”,IEEE Access, 2020.