*A Project Report*

*on*

**STOCK PREDICTION USING MACHINE LEARNING**

*carried out as part of the Minor Project (CS1634) Submitted by*

***Shubam Agarwal***

***169105188***

***6th semester B.Tech Computer Science and Engineering***

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

In

**Computer Science & Engineering**





**Department of Computer Science & Engineering,**

**School of Computing and IT,**

**Manipal University Jaipur,**

***April, 2019***

**CERTIFICATE**

This is to certify that the project entitled "***Stock prediction using machine learning***" is a bonafide work carried out as part of the course ***Minor project(CS1634)*** , under my guidance by ***Shubam Agarwal***, student of ***B.Tech 6th semester*** at the Department of Computer Science & Engineering , Manipal University Jaipur, during the academic semester ***2018-2019***, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, at MUJ, Jaipur.

Place:Manipal university jaipur

Date:26th April 2019 Signature of the Instructor (s)

**ABSTRACT**

In the financial sector of this world stock trading is one of the most important activities.Stock market prediction means to able to estimate the price of shares and other stock market amenities. Stock Prices are volatile and complex in nature . People invest in stock market based on their understanding of the current market and the prediction. For investing people search for different methods which can lower their risk of losing money and increase the chances of making profit.. Employing traditional methods like fundamental and technical analysis may not ensure the reliability of the prediction.

Here we compares the prediction of a stock using different Machine Learning algorithms.Here we are trying to predict the low price of the share price considering only date as the deciding factor. The programming language used to predict the stock market using machine learning is Python applied in jupyter environment. we will be training different algorithms on HDFC share price obtained from official website of Bombay stock exchange and evaluating their score. In this context this study uses a machine learning technique called Support Vector Machine (SVM) , linear regression , ridge regression , random forest , svr with rbf kernle and MLPS neural network with tanh activation and relu activation keeping in mind the training time and ease of implementation on a moderate PC.

Key Words:

Stock Market, Machine Learning, Predictions,Support Vector Machine , linear regression , ridge regression , random forest , svr with rbf kernel , MLPS neural network using tanh activation ad relu activation

**TABLE OF CONTENT**

**Page no:**

* 1. Introduction : 1

1.1Motivation

* 1. Literature Review 4
     + 1. What all have been tried
     1. Outcome of Literature Review
     2. Problem Statement
     3. Research Objectives.
  2. Methodology and Framework 4
     1. System Architecture
     2. Algorithms
     3. Design Methodologies
  3. Work Done 4
     1. Details
     2. Results and Discussion

1. Conclusion and Future 15

Work 5.1. Proposed Work plan of the project

6.References 16

**LIST OF FIGURES AND TABLES**

FIGURE 1:- Trend of HDFC stock price till january 2019

FIGURE 2.:- Fitting of linear regression model

FIGURE 3 :- Fitting of ridge regression model

FIGURE 4 :- MLP trend using tanh and relu activation

**INTRODUCTION**

Predicting the Stock Market is a dream come true for investors. Everyday billions of rupees are traded in share market, and behind every trade is an investor hoping to make profit in one way or another.Companies are shook by the rise and fall of the market. If we are able to predict the price we can create anormous wealth. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination every time it misbehaves. The 2008 financial crisis was no different, as evidenced by the flood of films and documentaries based on the crash.. Hence a better understanding of the market helps to avoid these types of mishap.

**Motivation**

Even though there is so much study about this market nobody is willing to share any strategies they might have developed for better prediction. A chief goal of this project is to add and improve academic understanding of stock market prediction so that greater understanding of how the market moves, investors will have better tools to prevent another financial crisis. The project will evaluate some existing strategiesand compare them. The investor may use his knowledge of the market combined with the predictions to take better decisions. We will focus exclusively on predicting the daily trend of a stock. The project will make no attempt to deciding how much money to allocate to each prediction. More so, the project will analyse the accuracies of these predictions.

**LITERATURE REVIEW**

**Zhang and Zhou (2004):** Here they have used kdd for financial application. They said that data mining techniques will play a major role in future for this. They tell that the data mining algorithm are Association rule mining (ARM) that shows attractive connection patterns among a large dataset by showing attribute-value conditions that occur together frequently. Classification and prediction explain and differentiate data classes or concepts, Clustering analysis segments a huge set of data into subsets or clusters. Each cluster is a collection of data objects that are similar to one another within the same cluster but not similar to objects in other clusters. Here Here they look for a patter in swquence and events following each other.

**Ting et. al (2006):** Here they used the Sequential and non-sequential association Rule mining (ARM) were used to look at between stocks and in the stock. Here they deriving the probable connection among the stocks and observed the performance of Stocks with respect to a specific stock index.

**Sudhanshu et. al (2009)** Here rthey have applied genetic algorithm with minimum complexity over the rules fetched from Apriori association rule mining and used baye’s theorem on the generated rules. Their proposed a system for generating association rule by genetic Algorithm by frequent item sets are generated using the Apriori association rule mining algorithm, and genetic algorithm item sets to generate the rules containing positive attributes, the negation of the attributes with the resultant part consists of single attribute and more than one attribute. . Here they describe that addition to the positive associations, negative associations can provide valuable information.

**Hajizadeh et. al (2010)** This provides a summary of application of data mining techniques such as decision tree, neural network, association rules, factor analysis and etc in financial markets. Also, this paper reveals further applications in addition to existing gap and less considered area. This paper has mainly concentrated on applications of the algorithms and comparison between them.

**Kerana and Rangaswamy (2011)** They present partition algorithm for mining frequent item sets using Clustering. This algorithm finds the frequent item sets by partitioning the database transactions into clusters. Clusters are formed based on the similarity between the transactions. Then it finds frequent Item sets with transactions in the clusters directly using improved Apriori algorithm which also reduces the number of scans in the database and hence improve the speed. Clustering is considered the most important unsupervised learning Problem.

**Dwivedi (2012)** Here they talk about data mining algorithms and models. To create a model, an Algorithm first analyzes a set of data and looks for specific patterns and trends. The algorithm uses the results of this analysis to define the parameters of the mining model. These parameters are then applied across the entire data set to extract actionable patterns and detailed statistics.

**OUTCOME:**

Many different methods have been tried to predict the stock price including creating hybrid algorithms , analysing the news , social media analysis for sentiments etc for stock prediction and have decent accuracy but their Here they never predict the volatility too good . Neural networks is the most common form used but faces the problem of overfitting whereas svm overcomes with this problem.

**PROBLEM STATEMENT:**  Stock market prediction using machine learning algorithms.

**RESEARCH OBJECTIVE:** To compare the different ML algorithms on stock market prediction in terms of their accuracy and fit

**Methodology :**

Here we have used different methodologies of machine learning models to predict the stock.We have used Linear regression , ridge regression , random forest , simple vector machine using rbf kernel , and MPLS .

**Algorithms:** the algorithms that have been implemented are Support Vector Machine (SVM) , linear regression , ridge regression , random forest , svr with rbf kernle and MLPS neural network with tanh activation and relu activation

**4.Work done:**

I studied the implementation of different machine learning algorithms from different sources and finalised the above mentioned algortihms for implementation considerind the time foe training and ease of implementation on a moderate pc. Then I collected the dataset of HDFC from official bse website that had stock prices , dates and many other features. I extracted the date and the low price from that dataset after reading it in jupyter notebook made few changes and implemented different algorithms and looked at their score and fit and also implemented plots for few.

**Code:**

import sklearn #import kneighborsRegressor

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Ridge

from sklearn.model\_selection import train\_test\_split

import numpy as np

import matplotlib.pyplot as plt

%matplotlib notebook

stock = pd.read\_csv("hdfc.csv" )

stock

#make date as index

stock['Date'] = pd.to\_datetime(stock.Date,format='%m/%d/%y')

stock.set\_index('Date')

plt.figure(figsize=(16,8))

plt.plot(stock['Low Price'], label='Low Price history')

x = stock['Date']

import datetime as dt

x = pd.to\_datetime(x)

x=x.map(dt.datetime.toordinal)

x

y = stock['Low Price']

y

x\_train , x\_test , y\_train , y\_test = train\_test\_split(x , y)

x\_train = x\_train.reshape(-1,1)

x\_test = x\_test.reshape(-1,1)

linreg = LinearRegression().fit(x\_train , y\_train)

train\_score = linreg.score(x\_train , y\_train)

test\_score = linreg.score(x\_test , y\_test)

print("test score on train data :{}".format(train\_score))

print("test score on test data :{}".format(test\_score))

plt.figure(figsize = (5,4))

plt.scatter(x , y ,marker = 'o' , s = 50 , alpha = 0.8)

plt.plot(x , linreg.coef\_\*x+linreg.intercept\_ , 'r-')

plt.title('stock linear regression')

plt.xlabel('date')

plt.ylabel('price')

plt.show()

linridge = Ridge(alpha=20.0).fit(x\_train,y\_train)

train\_score = linridge.score(x\_train , y\_train)

test\_score = linridge.score(x\_test , y\_test)

print("test score on train data :{}".format(train\_score))

print("test score on test data :{}".format(test\_score))

plt.figure(figsize = (5,4))

plt.scatter(x , y ,marker = 'o' , s = 50 , alpha = 0.8)

plt.plot(x , linridge.coef\_\*x+linridge.intercept\_ , 'r-')

plt.title('stock ridge regression')

plt.xlabel('date')

plt.ylabel('price')

plt.show()

from sklearn.ensemble import RandomForestRegressor

regr = RandomForestRegressor(max\_depth=2, random\_state=0,n\_estimators=100)

regr.fit(x\_train, y\_train)

train\_score = regr.score(x\_train , y\_train)

test\_score = regr.score(x\_test , y\_test)

print("test score on train data :{}".format(train\_score))

print("test score on test data :{}".format(test\_score))

'''xfit = np.linspace(0, 10, 1000)

yfit = regr.predict(xfit[:, None])

ytrue = model(xfit, sigma=0)

plt.errorbar(x, y, 0.3, fmt='o', alpha=0.5)

plt.plot(xfit, yfit, '-r');

plt.plot(xfit, ytrue, '-k', alpha=0.5);'''

from sklearn.svm import SVR

svr\_rbf = SVR(kernel='rbf', C=200, gamma=0.01, epsilon=.1)

svr\_lin = SVR(kernel='linear', C=100, gamma='auto')

svr\_poly = SVR(kernel='poly', C=100, gamma='auto', degree=3, epsilon=.1,

coef0=1)

svr\_rbf.fit(x\_train , y\_train)

train\_score = svr\_rbf.score(x\_train , y\_train)

test\_score = svr\_rbf.score(x\_test , y\_test)

print("test score on train data :{}".format(train\_score))

print("test score on test data :{}".format(test\_score))

#from sklearn.preprocessing import MinMaxScaler

'''scaling = MinMaxScaler(feature\_range=(-1,1)).fit(x\_train)

x\_train = scaling.transform(x\_train)

x\_test = scaling.transform(x\_test)'''

#svr\_lin.fit(x\_train , y\_train)

#train\_score = svr\_rbf.score(x\_train , y\_train)

#test\_score = svr\_rbf.score(x\_test , y\_test)

#print("test score on train data :{}".format(train\_score))

#print("test score on test data :{}".format(test\_score))

#svr\_poly.fit(x\_train , y\_train)

#train\_score = svr\_poly.score(x\_train , y\_train)

#test\_score = svr\_poly.score(x\_test , y\_test)

#print("test score on train data :{}".format(train\_score))

#print("test score on test data :{}".format(test\_score))

from sklearn.neural\_network import MLPRegressor

fig , subaxes = plt.subplots(2,1,figsize=(11 , 16))

x\_predict\_input = np.linspace(-3 , 3 , 50).reshape(-1 , 1)

for thisaxisrow ,thisactivation in zip(subaxes ,['tanh','relu']):

mlpreg =MLPRegressor(hidden\_layer\_sizes=[100 , 100],activation=thisactivation,alpha= 0.0001,solver = 'lbfgs').fit(x\_train , y\_train)

y\_predict\_output = mlpreg.predict(x\_predict\_input)

train\_score = mlpreg.score(x\_train , y\_train)

test\_score = mlpreg.score(x\_test , y\_test)

print("test score on train data :{}".format(train\_score))

print("test score on test data :{}".format(test\_score))

thisaxisrow.set\_xlim([-2.5 , 0.75])

thisaxisrow.plot(x\_predict\_input , y\_predict\_output ,'^',markersize = 10)

thisaxisrow.plot(x\_train , y\_train ,'o')

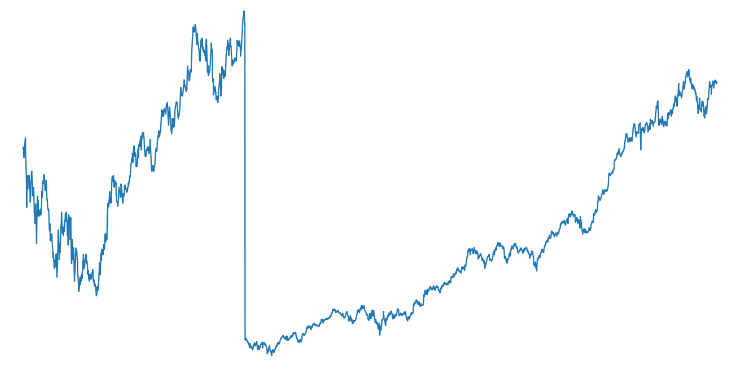
thisaxisrow.set\_xlabel('Input feature')

thisaxisrow.set\_ylabel('target feature')

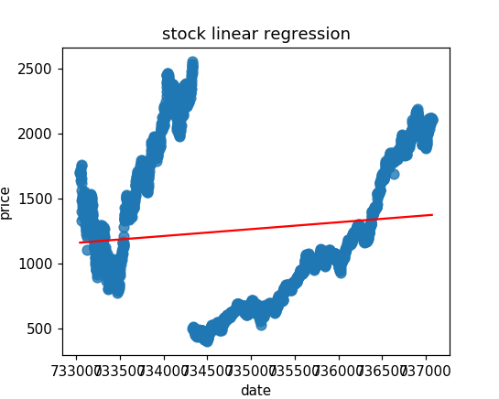
thisaxisrow.set\_title('MLP regression alpha = 0.0001 , activation ={}'.format(thisactivation))

plt.tight\_layout()

**Result:**



**FIGURE 1**

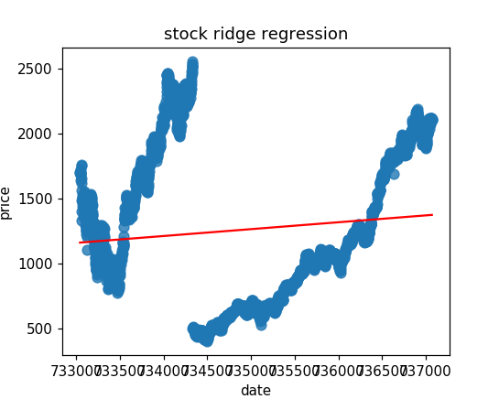


**FIGURE 2**

For Linear regression :

test score on train data :0.011723985140082771

test score on test data :0.0007118376410431004



**FIGURE 3**

**For ridge regression :**

test score on train data :0.011723985140082771

test score on test data :0.0007118376815350436

**For random forest:**

test score on train data :0.7236142591015899

test score on test data :0.7004294484572134

**For svr using rbf kernel and keepin c = 200 and gamma = 0.01:**

test score on train data :0.9955345736934454

test score on test data :0.9928495335693136

**For MLP using tanh and relu activation :**

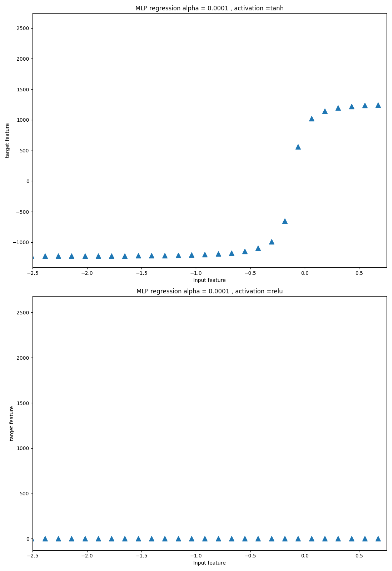
test score on train data :-2.220446049250313e-16

test score on test data :-0.005952907154652464

test score on train data :0.0007503798108795534

test score on test data :-0.005375480555421142

respectively



**FIGURE 4**

**CONCLUSION**

We conclude that stock prediction is best done using simple vector machine(svr) using rbf kernel followed by random forest.

**FUTURE**

In future I will get a generalised view of these algorithms on implementing them on large no of different stocks. Also I would consult few scholars in this field and try to come up with better algorithms for this purpose.

**References:**

1. Osman Hegazy , Omar S. Soliman 2 and Mustafa Abdul Salam3https://www.researchgate.net/publication/259240183\_A\_Machine\_Learning\_Model\_for\_Stock\_Market\_Prediction 27/03/2019

2. MARDUNNE http://markdunne.github.io/public/mark-dunne-stock-market-prediction.pdf , 16/03/2019

3. Özgür İcan1 & Taha Buğra Çelikhttps://www.researchgate.net/publication/320449394\_Stock\_Market\_Prediction\_Performance\_of\_Neural\_Networks\_A\_Literature\_Review , 10/04/2019

4. Rohith Gandhi, https://medium.com/datadriveninvestor/predicting-stock-prices-comparison-of-different-algorithms-ab2b8fd42514 , 5/04/2019

5. Rohith Gandhi https://medium.com/datadriveninvestor/predicting-stock-prices-comparison-of-different-algorithms-ab3b8fd42514 5/04/2019

6. Ezhilmaran Devarasan , prasnamma shantamma,,10/03/2019 ,https://www.researchgate.net/publication/313441684\_An\_analysis\_on\_Stock\_Market\_Prediction\_using\_Data\_Mining\_Techniques

7. Shangkun Deng, Takashi Mitsubuchi, and Akito Sakurai ,https://www.hindawi.com/journals/tswj/2014/861641/,28/02/109

8. V V Ramalingam1, A Pandian1, shivam Dwivedi1 and Jigar P. Bhatt1 https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012026 , 20/02/2019

9. Enric Junqué de Fortunya,⇑, Tom De Smedtb, David Martensa, Walter Daelemanshttps://www.clips.uantwerpen.be/~walter/papers/2014/jdmd14.pdf. 23/02/2019

10. Dwivedi,https://www.researchgate.net/publication/228664309\_Application\_of\_data\_mining\_techniques\_in\_stock\_markets\_A\_survey,01/02/2019

ss