SAN JOSÉ STATE
UNIVERSITY

PROJECT REPORT

CMPE 256

Stock Market Analysis and prediction

Instructor: Magdalini Eirinaki

Project Team

Ambika Bohra (011540269)

Sujana Jonnagadla (011423308)

Vishal Praanesh (011485175)

TABLE OF CONTENTS

1. Introduction

Forecasting of stock market is a way to predict future prices of stocks. It is a long time attractive topic for researcher and investors from its existence[2]. The Stock prices are dynamic day by day, so it is hard to decide what is the best time to buy and sell stocks. Machine Learning provides a wide range of algorithms, which has been reported to be quite effective in predicting the future stock prices.

In this project, we explored different data mining algorithms to forecast stock market prices for NSE stock market. Our goal is to compare various algorithms and evaluate models by comparing prediction accuracy. We examined a few models including Linear regression, Arima, LSTM, Random Forest and Support Vector Regression. Based on the accuracy calculated using RMSE of all the models, we predicted prices of different industries. For forecasting, we used historical data of NSE stock market and applied a few preprocessing methods to make prediction more accurate and relevant.

2. System Design & Implementation

2.1 Algorithms used

SVM(Support Vector Machine for Regression)

SVM is considered as one of the most important breakthroughs in machine learning field and can be applied in classification and regression[4]. In this project, SVR is considered to solve a regression problem as it avoids difficulties of using linear functions.

LSTM (Long Short-Term Memory)

Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture that learns about values using intervals. LSTM keeps track of the past values and use those changes to predict future values. In our project we have stock values for each day which can be treated as sequence of values. for its ability to act as memory unit, LSTM can be treated as one of the best algorithms for time-series analysis problems.

Y is present value and X is past value by one day. LSTM will link between X and Y to predict future value.

X	Y
22	35
35	48
48	52

Fig. 1: LSTM model

ARIMA (AutoRegressive Integrated Moving Average)

ARIMA model works appropriately for modeling time series with trend characteristics, random walk processes, and seasonal and nonseasonal time series[10]. It has simple structure that enables to model our time series dataset characteristics properly.

Random Forest Algorithm

Random tree is an ensemble of multiple trees. It has its own way of predicting values. We don't know whether the data is linear or not. In such cases, Random forest is effective.

Linear Regression

It is one of the most used algorithm for regression analysis. This algorithm is implemented to check how it works compared to other algorithms.

2.2 Technologies & Tools

Language and libraries : Python, SciPy, NumPy, Pandas, SciKit Learn, Keras.

Keras is required to implement LSTM model. the other libraries are required to process data and implement machine learning algorithms. Pandas made data preprocessing relatively easy.

Tool: Jupyter is convenient to use and is very fast.

2.3 System Design and Workflow

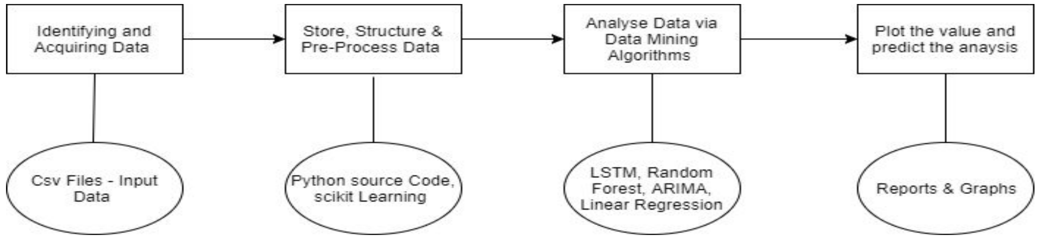

Identifying and Acquiring Data
Store, Structure & Pre-Process Data
Analyse Data via Data Mining Algorithms
Plot the value and predict the anaysis
Csv Files - Input Data
Python source Code, scikit Learning
LSTM, Random Forest, ARIMA, Linear Regression
Reports & Graphs

Fig. 2: System Design

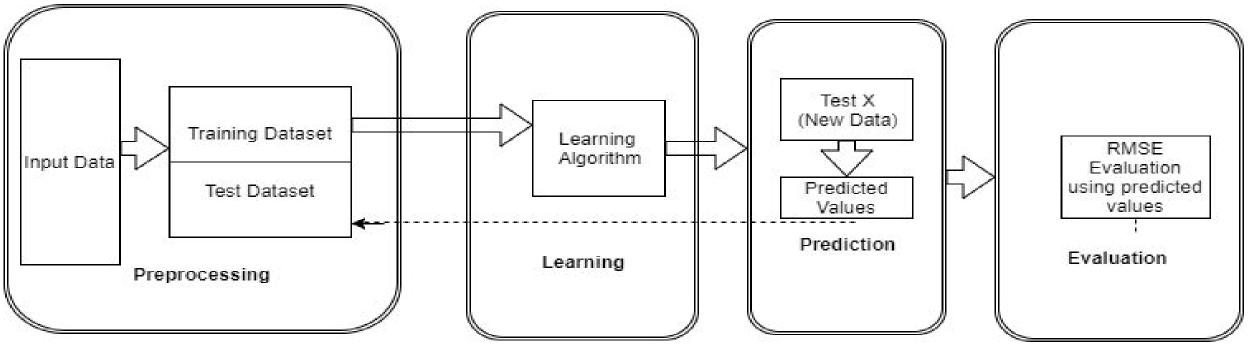


Fig. 3: Workflow

The input data is preprocessed by cleaning of data and splitting them into proper sets of training and test. This is in turn is fed to the learning algorithms for the main phase of analysis. Based on the output from the algorithm, the values are predicted and new data has been generated. Generated data are been used for the evaluation of the predicted values to find the accuracy of the algorithms efficiency.

3. Experiments / Proof of concept evaluation

3.1: Dataset Detail

In the project, we chose the National Stock exchange collected from [3]**.** This dataset includes India stocks and our index covers a diverse set of sectors featuring many indian companies. Our aim was to focus on making general and unbiased model, which works on every type of scenario irrespective of company or financial sector. It helps to validate our predictive algorithm and provide more accurate stock prediction.

Our dataset includes eight features such as company Index, Date, Time, Open, Close, High, Low values and Volume of trading (prices are in INR). The dataset covers 440 companies every minute since 2015. We took this dataset as it's size is quite large (~2gb) and it can be used to evaluate several companies using our algorithm. With the primary dataset prepared, we applied preprocessing methods to carry out individual experiments[1].

3.2: Data preprocessing

Pre-processing refers to the transformations applied to your data before feeding it to the algorithm. Selecting and pre-processing the data are crucial steps in any modeling effort, particularly for generalizing a new predictive model. Our dataset has some limitations such as it contains invalid values, null values and missing records etc. We applied following techniques to preprocess our data to make accurate prediction.

1. Data cleaning

   In real world, data tend to be incomplete and inconsistent. Therefore, the purpose of data cleaning is to fill in missing values and correct inconsistencies in the data. Index, Date, time closing prices of NSE dataset are used as input. There were some missing values due to public holidays. We removed null values and invalid indexes. There were few irrelevant columns in the dataset which were not used as input. So we eliminated those columns to reduce the complexity of our prediction model.

2. Data Transformation

   As our dataset contains minute-wise stock prices and we needed daily basis prices to fit in our model, so we grouped the data on daily basis prices and took mean of all the rows.. Also we applied min-max scaling for a few algorithms to get more accurate prediction.

Code	Date	Open	Close	Low	High	Volume
3IINFOTECH	2014-12-11	7.437	7.446	7.427	7.435	2538.135

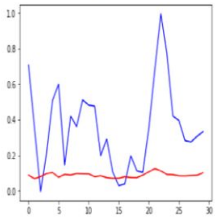
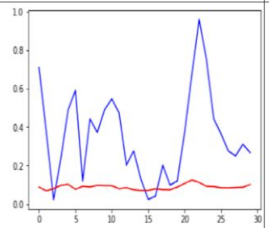
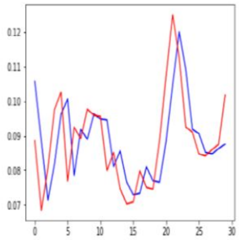
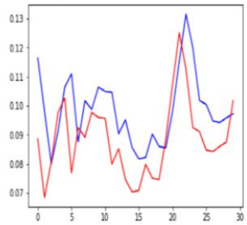
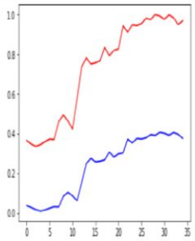
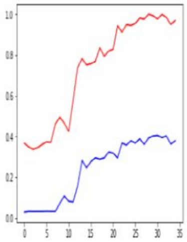
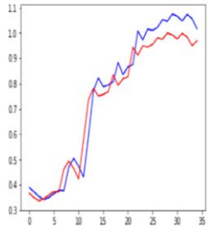
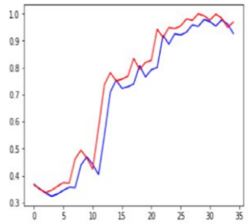
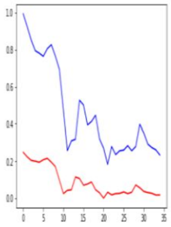
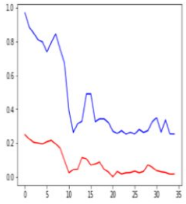
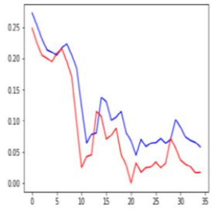
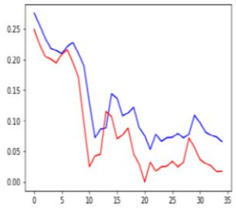
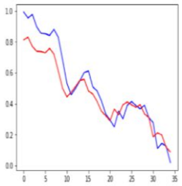
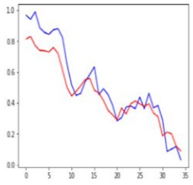
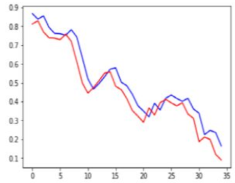
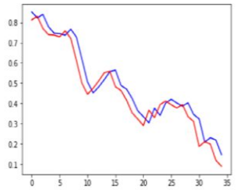
Table 1: Sample pre-processed Dataset

3.3: Methodologies

In this project, we have made a time-series analysis and it doesn’t need n-fold cross validation methodology since it’s sequential data. We split our dataset in train and test data. Top 80 percent of data will be Train data and the remaining will be test.

3.4: Comparison

Below are the few companies with graphs plotted for predicted values(Red) versus actual values **(**Blue**)** for different algorithms. We can see that LSTM and Arima performs better compared to random forest and linear regression

	Linear regression	Random forest	Ltsm (5th day)	Ltsm (1st past day)
3IINFOTECH				
8KMILES				
ABAN				
BOSCHLTD				

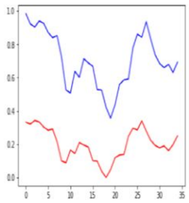
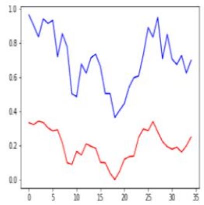
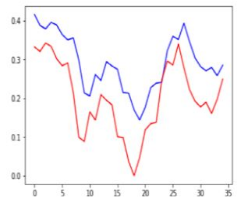
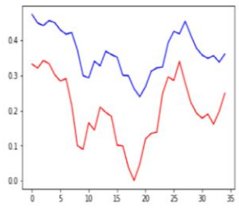
NTPC				

Fig 3: Actual closing price index and its predicted value from LR, RF, LSTM models

Arima model (monthly basis):

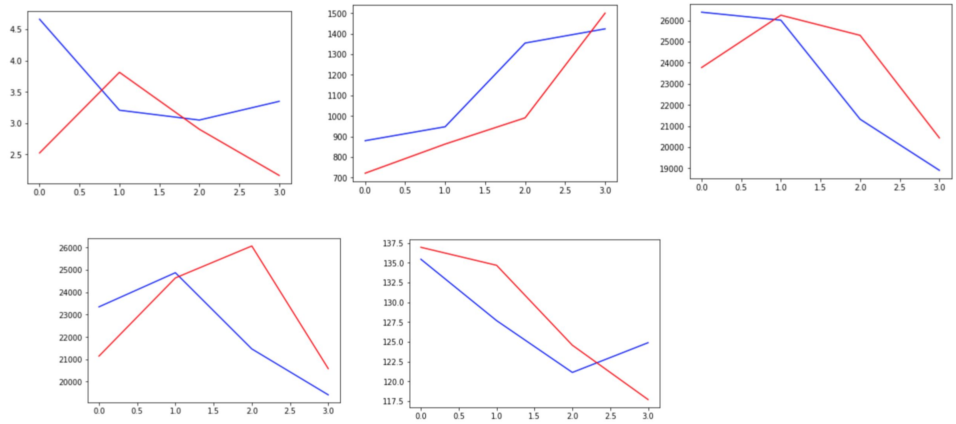


Fig 4: Graph Comparison for five companies (Left to right) Infotech, 8kmiles, Aban, Bosch Ltd, NTPC

3.5: Evaluation

The accuracy of prediction is referred to as “goodness of fit”. In this project, most popular and statistical accuracy measure RMSE is used for comparison of different algorithms on same dataset, which is defined as:

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2/N]^{1/2}$$

Below is the table of evaluation of all used algorithms:

Model	3IINFOTECH	8KMILES	ABAN
Linear Regression	0.334	0.502	0.415
Support Vector Regression	NA	NA	NA
LSTM (relating present and past fifth day)	0.011	0.064	0.043
LSTM (relating present and past day)	0.013	0.054	0.039
Arima	1.263	206.344	23.707
Random Forest	0.345	0.502	0.402

Table 2: RMSE comparison for three companies

4. Discussion & Conclusions

4.1: Decisions made

- We decided to make analysis using close values of the stock on a particular day or month and predict the closing values for future.
- Decided to work on different data mining algorithms which are Linear regression, Recurrent neural network, Support Vector machines, Random Forest, Arima model with different approaches.

4.2: Difficulties faced

- Data set is in Gb and takes some time to load and process.
- Preprocessing took some time to extract the values in required format and had difficulty initially on understanding how to forecast.
- Deciding the features to be considered for regression model.

4.3: Things that worked and didn’t work well

- The models that worked well are Long-short term memory RNN and Arima model.
- SVM and linear regression didn’t give accurate results.
- SVM took long time to process our large dataset.s

4.4: Conclusion

In the project, We proposed the use of different algorithms to predict the future stock prices of almost twenty companies. Although comparison is shown for only five companies (randomly selected) in the report due to space constraint, the behaviour can be known for any company by using the same code. Long short term memory algorithm worked best in case of forecasting. In future, we will extend the project for other effective methods that might result a better performance. Our algorithms can be used to maximize profit of investors but it has to be improved for real time conditions.

Project Plan / Task Distribution (1/2 page)

a. Research about algorithms and methods to solve the problem: Ambika, Sujana and Vishal
b. Preprocessing the data : Ambika and Sujana
c. Algorithms taken up and done by Sujana: LSTM, Random forest and linear regression
d. Algorithms taken up and done by Ambika: Arima, LSTM and SVM
e. Algorithms taken up and done by Vishal: linear regression
f. Presentation: Ambika, Sujana and Vishal
g. Report : Ambika, Sujana and Vishal

5. References

- ➢ http://markdunne.github.io/public/mark-dunne-stock-market-prediction.pdf
- ➢ http://citeseerx.ist.p su.edu/viewdoc/download?doi=10.1.1.278.6139&rep=rep1&type=pdf
- ➢ https://www.kaggle.com/ramamet4/nse-company-stocks
- ➢ https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
- ➢ https://ec.europa.eu/eurostat/sa-elearning/arima-model
- ➢ Cortes, C., Vapnik, V.: Support-vector Networks. In: Machine Learning, vol. 20(3), pp. 273– 297. Springer, Heidelberg (1995)
- ➢ **Source code link:** https://github.com/ambikabohra/Stock-Market-Prediction