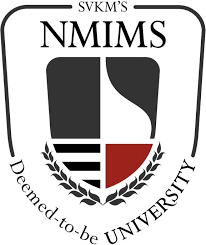
**PROJECT INTERIM**

**REPORT**

SVKM’S NMIMS Mukesh Patel School Of

Technology Management and Engineering



**Submitted to: Submitted By:**

Dr Radhakrishnan Ram bola Shivam Sekra(N266)

HOD (Computer Science) Abhishek Singh(N269)

(Project Mentor) Prakhya Singh(N270)

A REPORT ON

**Stock price prediction-Stacked LSTM(RNN)**

Presented to

Head of Department

DR. Radha Krishna Rambola

By

Shivam Sekra(N266)

Abhishek Singh(N269)

(Prakhya Singh(N270)

A Report Submitted in partial fulfilment of the requirement of 5 years Integrated MBA (Tech) Program of Mukesh Patel School of Technology Management & Engineering, NMIMS

# Completion Certificate

Has completed training & project as a part of Technical Internship as mentioned below and the Report is also submitted.

1. Project Title: Stock price prediction-Stacked LSTM(RNN)
2. Date of Joining: 4-May-2020
3. Date of Completion: 30-June-2020

In partial fulfilment of XII Semester Technical Internship for MBA(Tech) program of Mukesh Patel School of Technology Management & Engineering, Narsee Monjee Institute of Management Studies (NMIMS)

(Deemed-to-be University), Mumbai.

……………………………………….

Faculty Mentor

Date:

Place:

Institution Seal:

# Acknowledgement

I would like to thank my institution SVKM’s NMIMS Mukesh Patel School of Technology Management and Engineering, Dr R. S. Gaud, Director (SVKM’s NMIMS, Shirpur Campus), Dr. Nikhilesh Kumar Sharma, Director (Engineering Program, MPSTME NMIMS Shirpur Campus), Dr. Narayan Chandak, (MPSTME NMIMS, Shirpur Campus), Associate Dean Dr. Radhakrishna Rambola, Head of Department (MPSTME NMIMS, Shirpur Campus) for introducing me to this Project, which gave me the great opportunity to expose myself to the real world besides giving me a platform to have hands-on experience to learn new technologies and apply them.

My faculty mentor Dr. Radhakrishna Rambola, Associate Dean (MPSTME NMIMS, Shirpur Campus) has given me constant support, always motivates me to step forward and besides being a source of inspiration through his words of encouragement. His continuous supervision and much needed guidance has really helped me in conceptualizing my project and executing the same.

I place a deep sense of gratitude towards my family members and my friends who have been a constant source of inspiration to make this project executable. Many people, especially our classmates and team members itself, have made valuable suggestions on this proposal which gave us an inspiration to improve our project. We thank all the people for their help directly and indirectly to complete our project within the prescribed time.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, to attain desired career objectives.

Date: 30-June-2020

**Table of Content**

|  |  |
| --- | --- |
| **Content** | **Page No.** |
| Abstract | 6 |
| Introduction | 6 |
| Project Aim | 7 |
| Problem statement | 8 |
| H/w and S/w required | 9 |
| Project Methodology | 10 |
| Design and Implementation  (Dataset,Visualization,Pre-processing) | 11 |
| Modeling | 14 |
| Comparison Analysis | 19 |
| Hyper-parameter Tuning | 20 |
| Testing and Final Result Analysis | 23 |
| Conclusion | 24 |
| References | 25 |

**ABSTRACT**

The prevailing theories is that stock prices are totally random and unpredictable but that raises the question why top firms like Morgan Stanley and Citigroup hire quantitative analysts to build predictive models. As these days they're more likely to see rows of machine learning experts quietly sitting in front of computer screens. Infact about 70% of all orders on Wall Street are now placed by software, we're now living in the age of the algorithm.

This project will help and try to predict stock price and trends, that will ultimately help to lowering the market risk with help of machine learning and deep learning algorithms like linear regression, KNN, and LSTM(long short term memory) with hyper parameter tuning to get best out of the algorithm.

**INTRODUCTION**

The ﬂuctuation of stock market is violent and there are many complicated ﬁnancial indicators. However, the advancement in technology, provides an opportunity to gain steady fortune from stock market and also can help experts to ﬁnd out the most informative indicators to make better prediction. The prediction of the market value is of paramount importance to help in maximizing the proﬁt of stock option purchase while keeping the risk low.

The next section of the paper will be methodology where we will explain about each process in detail. After that we will have pictorial representations of the analysis that we have made and we will also reason about the results achieved. Finally, we will deﬁne the scope of the project. We will talk about how to extend the paper to achieve more better results.

**PROJECT AIM:**

The main aim of this project is that it will help and try to predict stock price and trends, that will ultimately help to lowering the market risk with help of comparison analysis of different ML and DL algorithms to create a successful model for stock prediction.

**MOTIVATION:**

Stock price prediction is a classic and important problem. With a successful model for stock prediction, we can gain insight about market behaviour over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, machine learning will be an efficient method to solve this problem. The motivated idea is that, if we know all information about past stock trading (of all specific traders), the price is predictable. Thus, if we can build a model to predict prices, we can expect to improve the current prediction a lot.

**PROBLEM STATEMENT:**

The main work of the project is to examine a number of different forecasting techniques to predict future stock returns based on past returns to successfully predict stock price changes and will **compare different ML and DL models that can be used in prediction to find the best algorithm for prediction.**

**GOALS:**

* Explore stock prices.
* Implement basic model of linear regression.
* Implement K-NN regression model.
* Implement LSTM(RNN) model.
* Compare the results and submit the project.

**HARDWARE AND SOFTWARE REQUIRED:**

**Hardware Requirement:**

• i3 Processor Based Computer or higher

• Memory: 2 GB or higher

• Hard Drive: 50 GB

• Monitor

• Internet Connection

• GPU(nvidia for cuda library)

**Software Requirement:**

• Windows 7 or higher

• Google Chrome Browser

•Python(anaconda env)

•Keras , tensorflow(v2.2.0) libraries

• CUda (GPU use for running codes faster)

**PROJECT METHODOLOGY**

This section will give you the detailed analysis of each process involved in the project. Each sub section is mapped to one of the stages in the project.

• **Data Collection**: This stage includes collection of required data from different sources and integrating it.

• **Data pre-processing**: This includes Pre-processing techniques like Data Cleaning, scaling, removing unwanted features, handling null values, splitting train and test data etc.

• **Data Visualization**: Gaining insights from data to help us understand data better.

• **Model creation**: This is stage which includes training different models and algorithms with comparing their performance with help of visuals and metrics like RMSE (root mean square error).

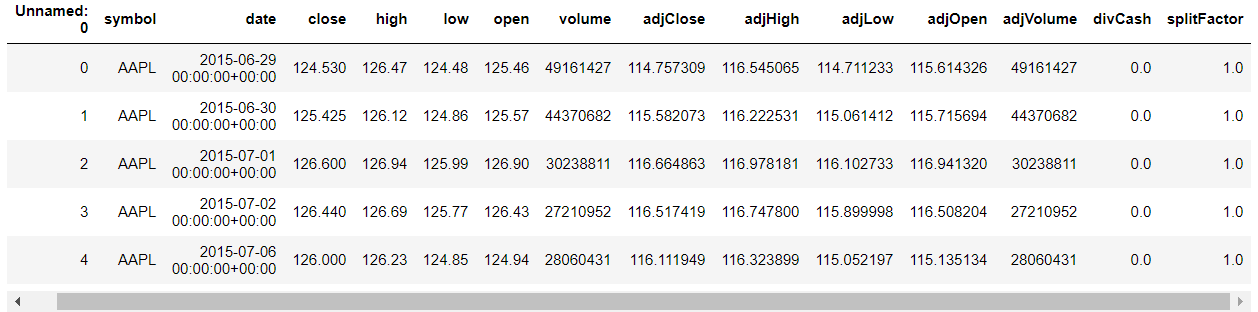
•**Hyper parameter tuning and finalising**: Tuning algorithms with hyper parameter and getting best results out of algorithms.

Finally, we get the best algorithm which successfully gives us best prediction.

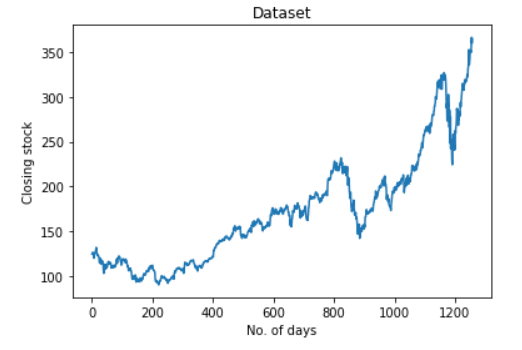
**DESIGN AND IMPLEMENTATION**

**Data Exploration(Dataset):**

* The data has been collected form website <https://www.tiingo.com/> using an API key provided by website to retrieve the data.
* From pandas package pandas\_datareader has been used to connect api with the website to retieve data.
* **Apple** company stock data of 1258 days is been used in the project.
* As the input data has many features (columns) , we are extracting only ‘close’ column .
* Dataframe:



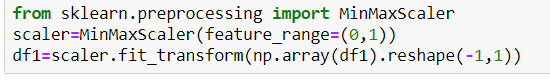
**Data visualization**: ‘APPLE’ data: for this we have used matplotlib library to plot the graph.



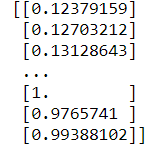
**Scaling (Minmax Scaler):** In this approach, the data is scaled to a fixed range - usually 0 to 1. The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers.

* This scaling is done as the algorithms we are using are sensitive to the outliers and large deviations can create un-predictable results with our models.

**Code:**



Data after scaling:



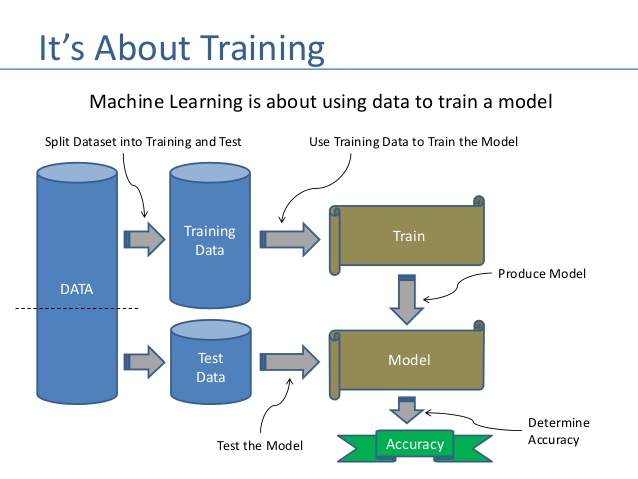
Note: we didn’t observe any null values or any abnormal data like negative data.

**Splitting Data (train, test and validation set):**

* There are a lot of ways for Training-Testing method like Cross Validation, Some Random Seeds, but both of these methods are more effective when there’s a linear relation between Data. Since here we are using time series, we will be directly splitting data into three parts that is train part and test part and validation part.
* Train set (X\_test, y\_test): 60% of full dataset. Used to train the models.
* Validation set(X\_cv,y\_cv): 20% of full dataset. Used to tune hyper-parameters and compare the results.
* Test set(X\_test,Y\_test):20% of full dataset. As the developed model should not be tested on a previously seen data. So, accuracy is checked through predicting using X\_test and comparing it with y\_test.

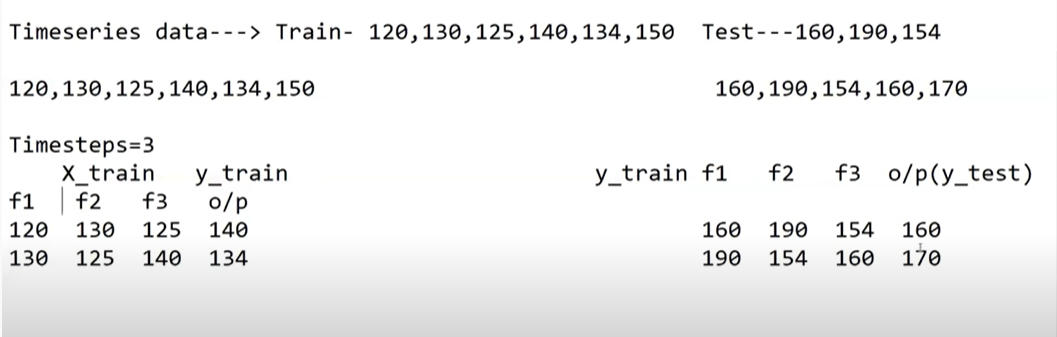
How training-testing works?

* The **training data** is used to make sure the **machine** recognizes patterns in the **data**, the cross-validation **data** is used to ensure better accuracy and efficiency of the algorithm used to **train** the **machine**, and the **test data** is used to see how well the **machine** can predict new answers based on its **training**.



**Pre-Processing:**

* Pre-processing is the converting of a Data Set into Independent and Dependent Data using Time step.
* "**Timestep**" (N)is the time interval or number of days the algorithm will take to predict the next value. Initially N= 9;

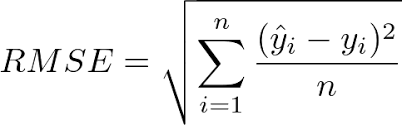


**MODELING (Model Training):**

* Modeling: training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.
* Technology used:
* TensorFlow(keras): As background, Keras is a high-level Python neural networks library that runs on top of either TensorFlow (ML and DL library to provide interface for keras) or Theano.
* As per keras we are using sequential model (A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.) with dropout layer.
* Linear regression has been imported from sklearn.linear\_model . Also, KNN (k-nearest neighbor regression) has been used.
* -Loss function used is RMSE (root mean squared error)

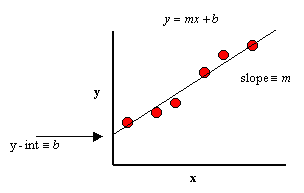
**RMSE (Root mean squared error):**

The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model’s predicted values. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.



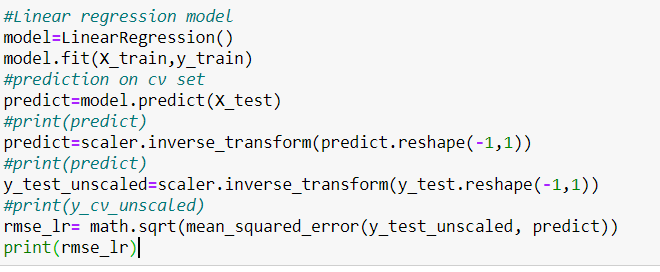
**MODEL-1 (Linear regression):**

* **Linear regression** models are **used to** show or predict the relationship between two variables or factors.
* The **equation** used in algo. has the form Y= a + bX, where Y is the dependent variable (that's the variable that goes on the Y axis), X is the independent variable (i.e. it is plotted on the X axis), b is the slope of the line and a is the y-intercept.



* Use:

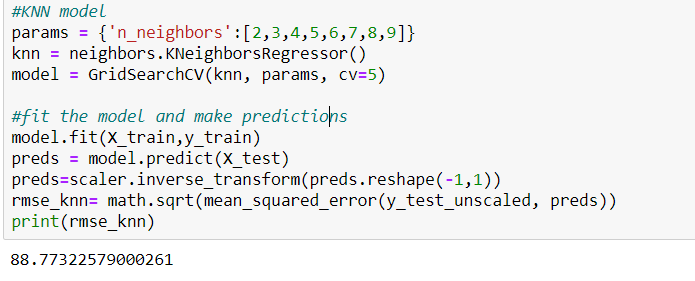
from sklearn.linear\_model import LinearRegression



**MODEL-2 (K-NN regressor):**

* **KNN** algorithm can be used for both classification and regression problems. The **KNN** algorithm **uses** 'feature similarity' to **predict** the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.
* Use:

from sklearn import neighbors

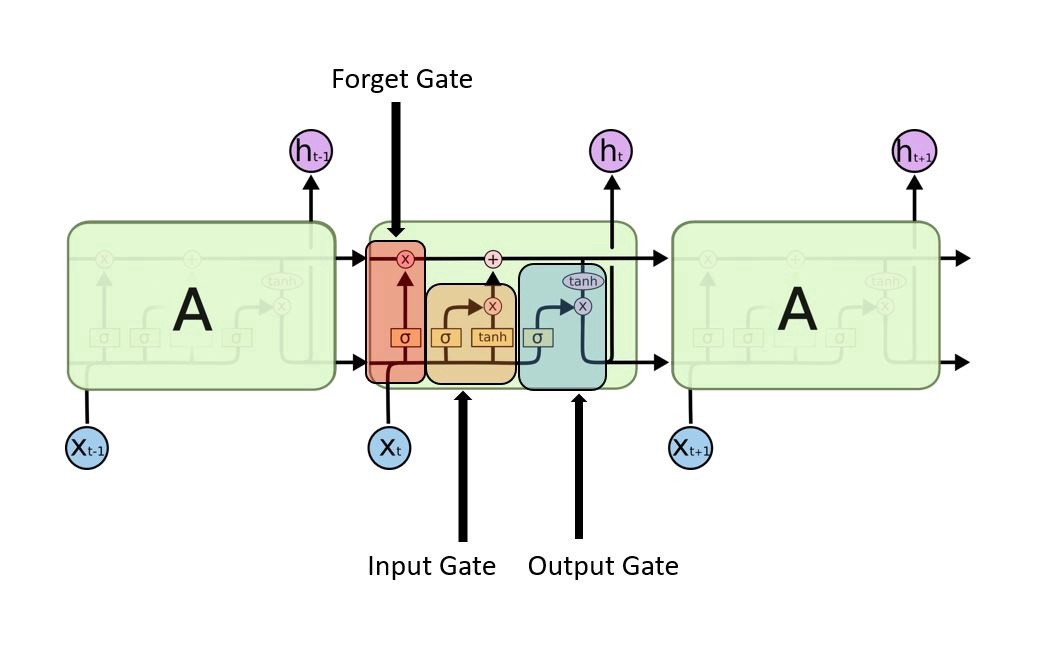


**MODEL-3(RNN-LSTM**)

* LSTM -Advanced version of RNN that is Stacked LSTM (Long short-term memory) which is used in time series data (sequential data) where time is important feature in training model.
* We are using LSTM as RNN has a disadvantage of vanishing and exploding gradient problem so we choose LSTM RNN which overcomes it.

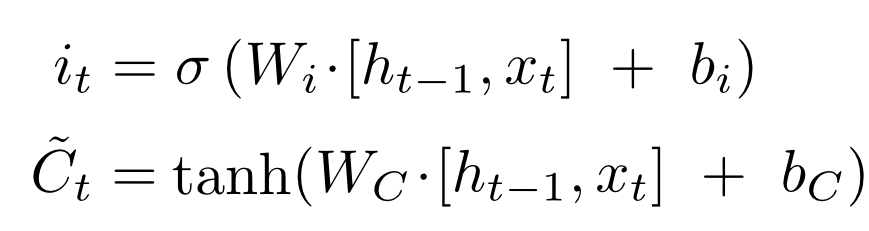
**What is Long Short-Term Memory (LSTM)?**

* Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present:

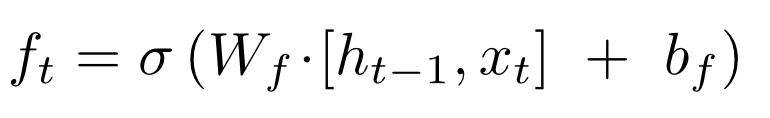


**LSTM gates**

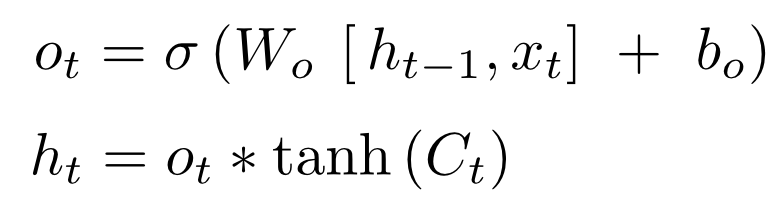
**1.Input gate** — discover which value from input should be used to modify the memory. **Sigmoid** function decides which values to let through **0,1.**and **tanh**function gives weightage to the values which are passed deciding their level of importance ranging from**-1** to **1.**



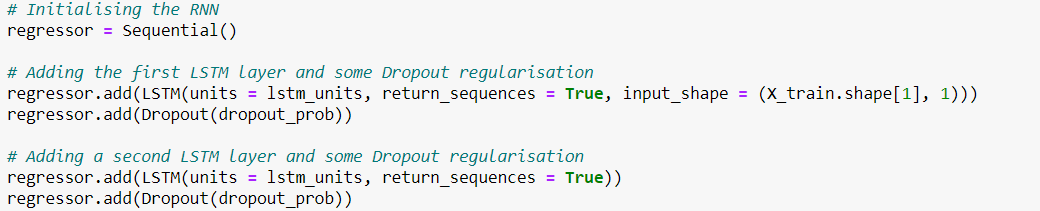
**2. Forget gate**— discover what details to be discarded from the block. It is decided by the **sigmoid function.**it looks at the previous state(**ht-1**) and the content input (**Xt**) and outputs a number between **0(***omit this*) and **1(***keep this***)** for each number in the cell state **Ct−1**.



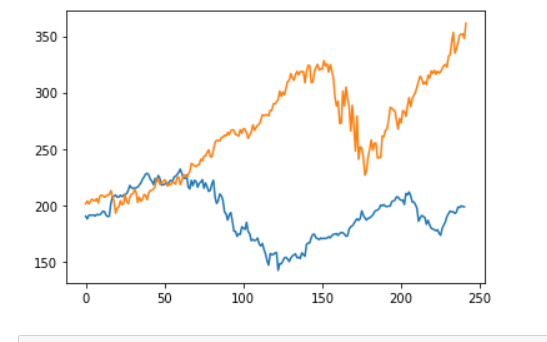
**3. Output gate** — the input and the memory of the block is used to decide the output. **Sigmoid** function decides which values to let through **0,1.**and **tanh**function gives weightage to the values which are passed deciding their level of importance ranging from**-1** to **1**and multiplied with output of **Sigmoid.**



Use:

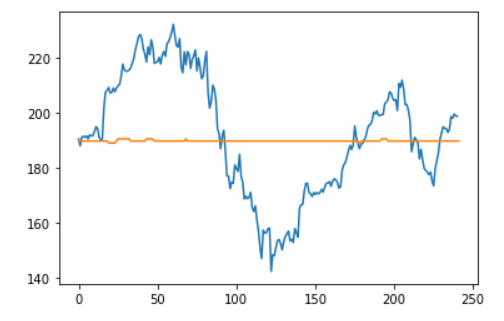


**COMPARISON ANALYSIS**

**Comparing results from three models**

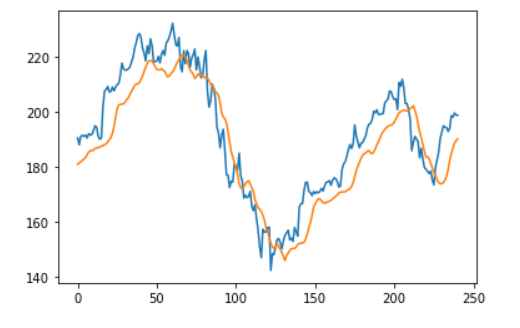
KNN-Regressor

RMSE =88.732



Linear Regression

RMSE =66.7921



As we observe that LSTM Out-perform with lowest RMSE Score. SO, now we tune hyperparameters of LSTM-RNN to get better results.

LSTM-RNN

RMSE =52.977

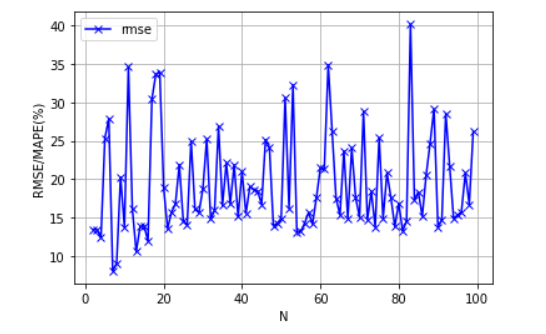
**HYPER-PARAMETER TUNING**

**Hyper-parameters:**

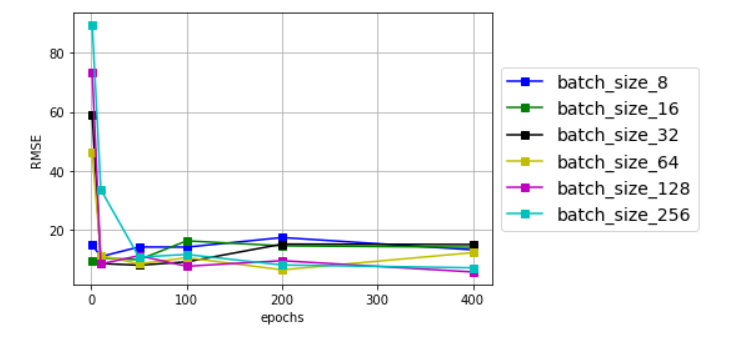
* **LSTM-units:** referring to the dimensionality of the hidden state and dimensionality of the output state (they must be equal). A LSTM comprises an entire layer.
* **Dropout-probability**: Dropout is a technique where randomly selected neurons are ignored during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.
* **N:** N is the number of Time-steps used to predict the next value.
* **Optimizer**: Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses. Optimizers help to get results faster.
* **Epochs**: One epoch is when an entire dataset is passed both forward and backward through the neural network only once.
* **Batch size**: refers to the number of training examples utilized in one iteration. Can`t be greater than training data.

-Optimum value of hyper-parameters is obtained in this phase of modeling using cv\_data (cross-validation data) (dataset was divided into-train set, test set and validation set) to tune the model loss function.

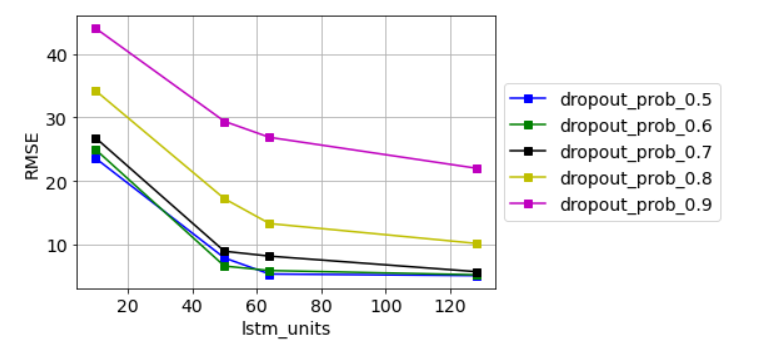
**N:**



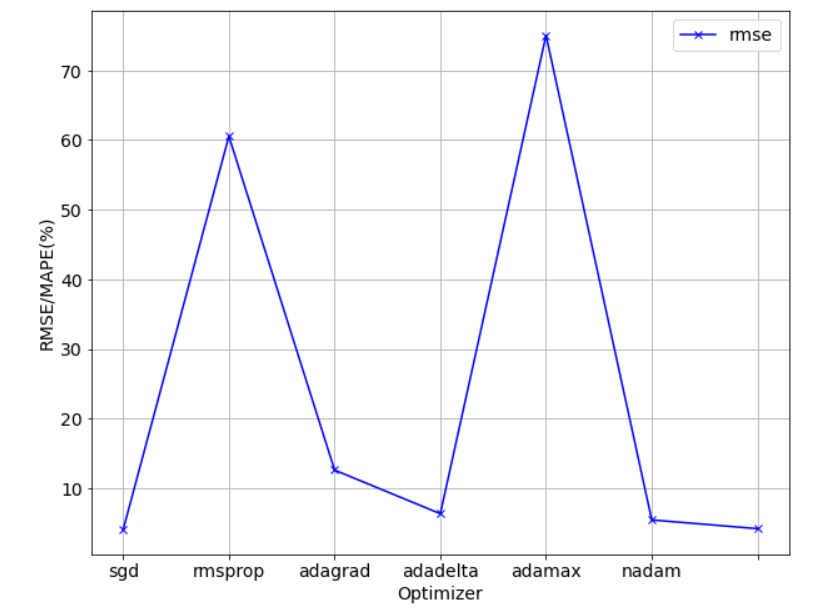
**Batch Size and Epochs:**

****

**Dropout Probability and LSTM units**



**Optimizer:**

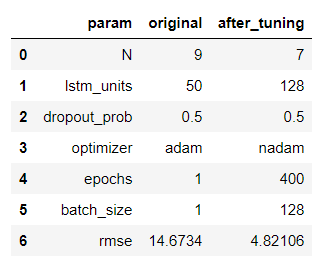
****

**TESTING AND RESULT ANALYSIS**

**Final Model**

* -Final model is being trained on train set and test set using optimum hyper-parameters obtained after tuning phase.
* -RESULT: After comparing three models we came to this conclusion, that LSTM-RNN is the best algorithm to predict the stock prices.
* RMSE (LSTM-after tuning) =6.9591
* RMSE (LSTM-before tuning) =52.977

**Optimized Hyper-parameters**:

****

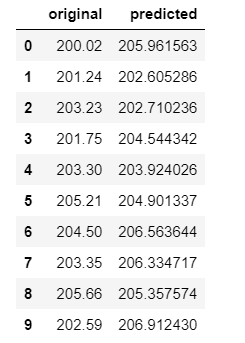
**CONCLUSION**

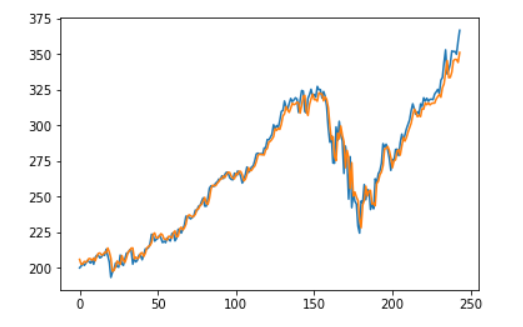
LSTM (RNN) outperforms among all the 3 algorithms we used for comparison with a good RMSE score as LSTM is a one of a type of Neural Networks in which sequential data (Time-series Data) is handled giving good accuracy and prediction scores as we observed in this project.

So, we conclude that we can`t predict exact stock prices but we can achieve some good accuracy and for that Deep Neural network (LSTM) model is best suited. Also, similar kind of prediction like sales Prediction and forecasting can also be successfully done using LSTM models which would in-turn give better results than other simple regression algorithms which are available in Machine learning.

: Original

: Predicted

****

****

Closing Stock

No. of days

**REFERENCES**

[1] H. N. I. H. M. K. A. S. Khalid AL khatib, "Stock Price Prediction Using Nearest Neighbor (kNN) Algorithm".

[2] A. N. Refenes, A. Zapranis and G. Francis, "Stock performance modeling using neural networks: a comparative study with regression models," Neural Networks, vol. 7, no. 2, pp. 375-388, 1994.

[3] C.-F. Tsai and Y.-C. Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches," Decision Support Systems, vol. 50, no. 1, pp. 258-269, 2010.

[4] "Prediction with LSTM," [Online]. Available: <https://lilianweng.github.io/lillog/2017/07/08/predict-stock-prices-using-RNN-part-1.html>.

[5] W. Bao, J. Yue and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and longshort term memory," PLoS ONE, vol. 12, no. 7, 2017.

[6] "Understanding LSTM networks," [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

[7] "Recurrent neural Network - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/Recurrent_neural_network>.

[8] " Sample of a feed-forward neural network," [Online]. Available: https://www.researchgate.net/figure/Sample-of-a-feed-forward-neuralnetwork\_fig1\_234055177.