Endsem: Predictions of US-INR exchange rate and stock prices of various companies using Data Analytics

S, Karthik
ME18B149 - IDDD Data Science
Indian Institute of Technology Madras
Chennai, India
me18b149@smail.iitm.ac.in

***Abstract*—This document gives an overall data analysis of the stock prices of various companies and also the US-INR exchange rates for a long enough period of time and also discusses models for predicting the future prices/rates. Various data analytic techniques such as regression and recurrent neural networks were used to analyze and predict the data.**

***Index Terms*—Stock prediction, Machine Learning, Deep Learning, LSTM, RNN, Data Science**

I. Introduction

Data Science is a popular subject nowadays. Everyone is all about data. What it can do and how can it help. Many times data is represented as numbers and these numbers can represent many different things. These numbers could be the amount of sales, inventory, consumers, and last but definitely not least — cash.

This specific paper discusses about prediction of stock prices of various companies and also the US-INR exchange rates using various techniques along with some insights from the data. This will be helpful in making decisions related to investments in the finance sector.

II. Data science in the finance sector

This brings us to financial data or more specifically the stock market. Stocks, commodities, securities, and such are all very similar when it comes to trading. We buy, we sell, we hold. Setting aside the difficulties of predicting the stock market's behavior, let's examine some data science tools that might be useful in trying to make such predictions.

For starters, data scientists use a lot of algorithms to help gauge what the stock market may or may not do. Algorithmic trading identifies when buying or selling a stock is ideal – such as buying after a stock only after it has decreased in value by a certain percentage in a certain timeframe, like 2.5 percent in a four-hour period.

Another way that data science can be used in the stock market is to use models. This involves exploring data related to past stock market behaviors and using that to forecast what might happen in the future.

Usually, data scientists use time-series models for this, such as the price of a stock that's ordered by a set amount of time, like hourly, daily, or monthly. By evaluating how a stock's price has performed over the course of the last week, traders can predict what might occur with the stock's price in the upcoming week.

III. Training, Validation and Testing

Data scientists might also use what's called training to try to predict what the stock market will do.

Training involves using certain data to teach machine learning what predictions to make on past data. So, a data set would be split in two, typically with 80 percent of the data being used for training and 20 percent used for testing.

The 80 percent of the data that's used for training would be comprised of past data on a particular stock, such as the historical trend of a stock's price over the last year. Then, machine learning would use that information to predict what the price of the stock might do over the course of the next month, six months, year, and so on.

To validate what the machine learning has predicted, data scientists would compare the predictions with the testing set of data.

For example, if machine learning is trained on 12 months of a stock's price data, the data from the first ten months would be used for training and the data from the final two months would be used for testing. Then, using what happened in the first ten months, machine learning would predict what would happen in the final two months. The model's predictions would then be compared to what actually happened.

The goal in doing so is to see how accurate the model is in predicting stock market behavior. The goal here would be to reduce the error between the real data and the predictions in order to create a reliable model for predicting stock market behavior.

IV. The Problem

A. Overview and objectives

The problem is an open ended problem. Several datasets are given which provide the history of stock prices of various companies and also the US-INR exchange rate over the years. Predicting stock prices and the exchange rate can be the most important application where data analytics can come into the fore. In this paper, a detailed analysis of the given data along with various models to predict data will be done.

B. *Reading the data*

We will be using python throughout this assignment. The datasets that contain the stock price of HCL Technologies, Infosys, Cognizant, State Bank of India, HDFC Bank, ICICI Bank, and also USD-INR exchange rate are given. All the datasets have 4 common columns among them as follows:

1) Open: The opening price is the price at which a stock-/currency first trades upon the opening of an exchange on a trading day.

2) Close: The closing price is a stock's or currency's trading price at the end of a trading day. This makes it the most recent price of stock until the next trading day.

3) High: The highest closing price of the stock/currency for the particular day.

4) Low: The low is the lowest price of stock/currency for a particular day.

Since we are creating a model, we would be splitting the training dataset into train and validation sets with a fraction of 20% for validation.

V. Exploratory Data Analysis and Data cleaning

Our first task is to analyze the data and form visible conclusions and trends from the same with the help of plots. Let us first plot all the stock price variables(Open, Close, Low, High) for Cognizant company. We can see from the plot (Fig 1) that all the 4 variables are not very different in terms of trends.

Since the main objective is to predict the closing price using history of closing process, the only variable of interest is the Closing price in this dataset apart from the date and the volume which are inputs. Hence, we will remove all the other variables from our analysis. Also, for analyzing variables such as opening price, high price and low price, we can follow a similar analysis as the paper describes with just the closing variable replaced by the variable of interest.

The plot comparing different companies' closing stock prices and the exchange rate is shown in Fig 2. Since the absolute scores are different, to compare the relative trends, the prices are normalized and plotted in Fig 3. Also the stock volume is plotted in Fig 4.

Since there are some entries with closing price as NaN, we will remove those entries from our analysis. They are not imputed as the number of removed entries are small and it wont affect the analysis in any way.

A. *Moving Average*

Moving average is used to filter out the noisy data. In this case, it removes the sudden changes thereby, making the overall trend in the stock price movement very clear. Since, the span of the moving average filter is an important factor in the analysis, 3 different values (10, 50, 200 days) of the same were tried out and plotted for each of the companies. Fig 5 to Fig 11 shows the moving average filter applied to all

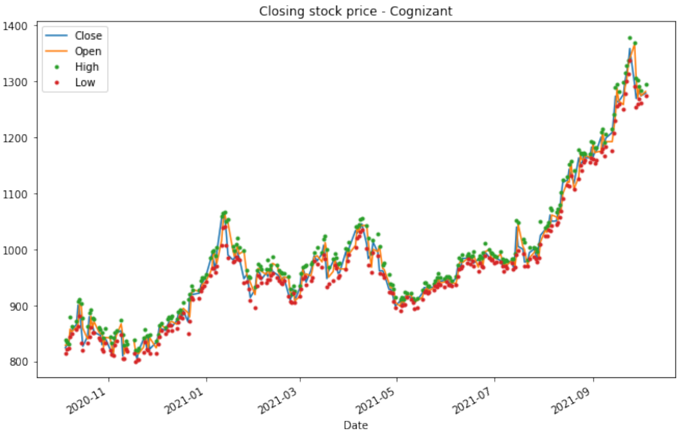


Fig. 1. Open, Close, High and Low prices consolidated plot - Cognizant

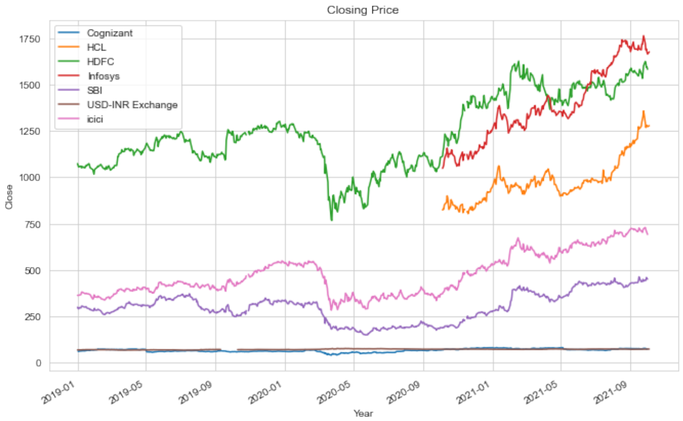


Fig. 2. Comparison of stock prices across various companies, unnormalized

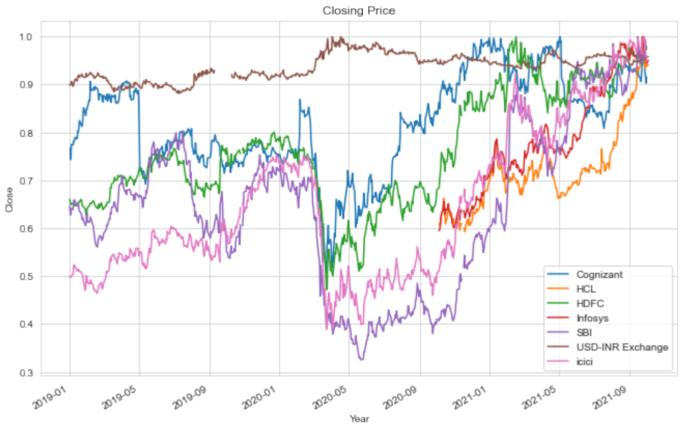


Fig. 3. Comparison of stock prices across various companies, normalized

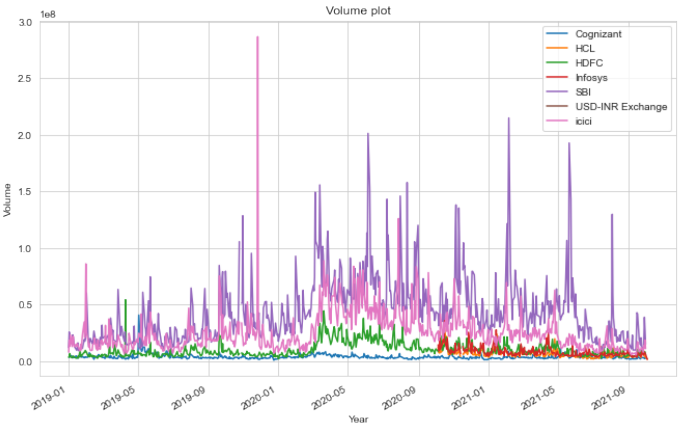


Fig. 4. Comparison of stock volume across various companies

TABLE I
NAN VALUES IN THE DATASET

Company Name	Number of entries
Cognizant	0
HCL	1
HDFC	2
Infosys	2
SBI	1
ICICI	21
USD-INR exchange	2

the companies and the exchange rate datasets. It is clear from the plots that lesser the number of days, more accurate the plot will be to the true trajectory. The purpose of this filter is to remove unwanted noise and disturbances which will give a clear trend that will help in analysis and visualization.

From the above plots, the following conclusions can be made

- The variables Open, Close, Low, High do not vary drastically in terms of trend. They follow more or less the same trend with minor errors.

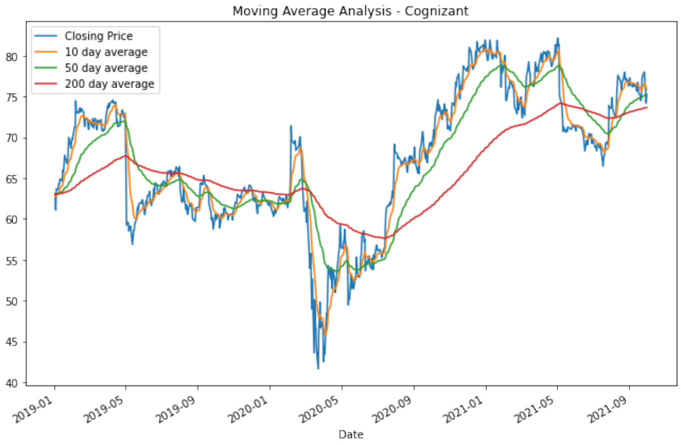


Fig. 5. Moving average plot - Cognizant

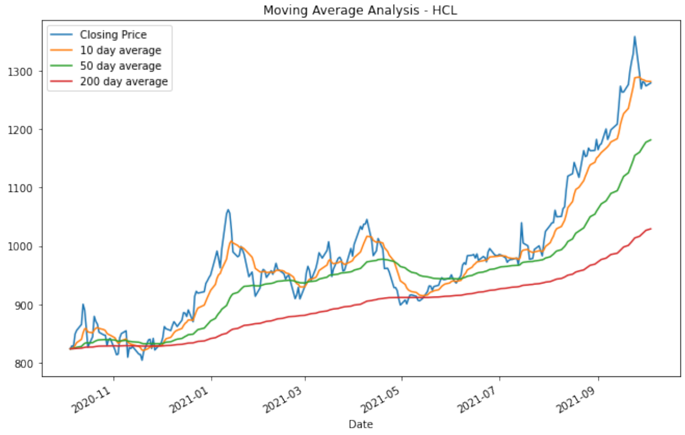


Fig. 6. Moving average plot - HCL

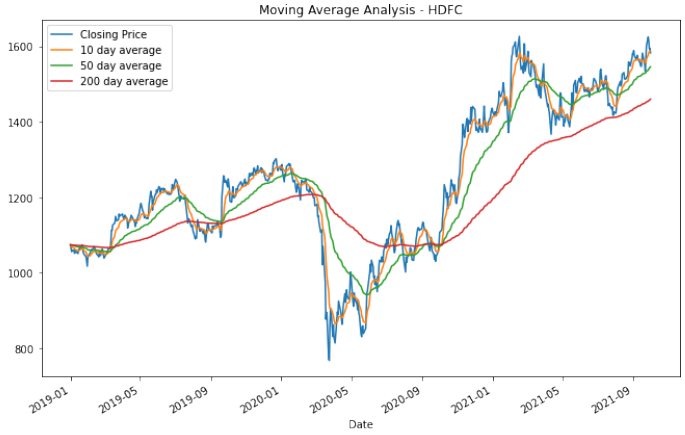


Fig. 7. Moving average plot - HDFC

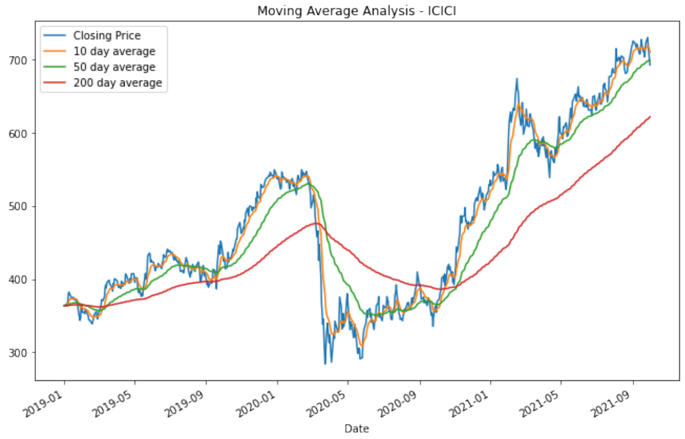


Fig. 8. Moving average plot - ICICI

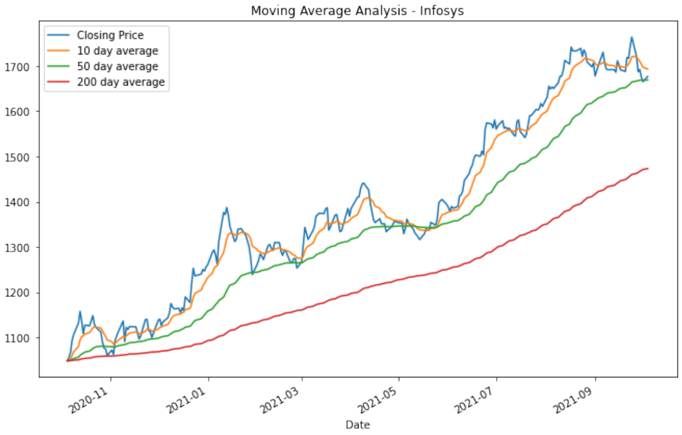


Fig. 9. Moving average plot - Infosys

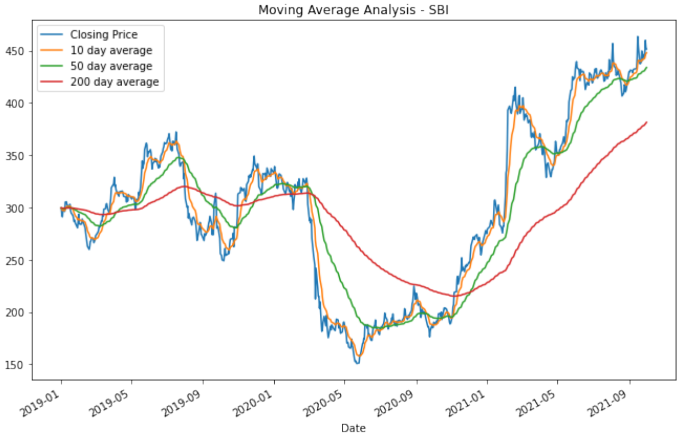


Fig. 10. Moving average plot - SBI

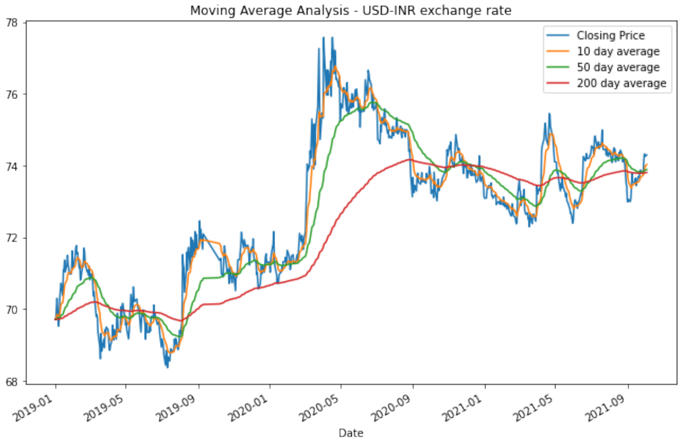
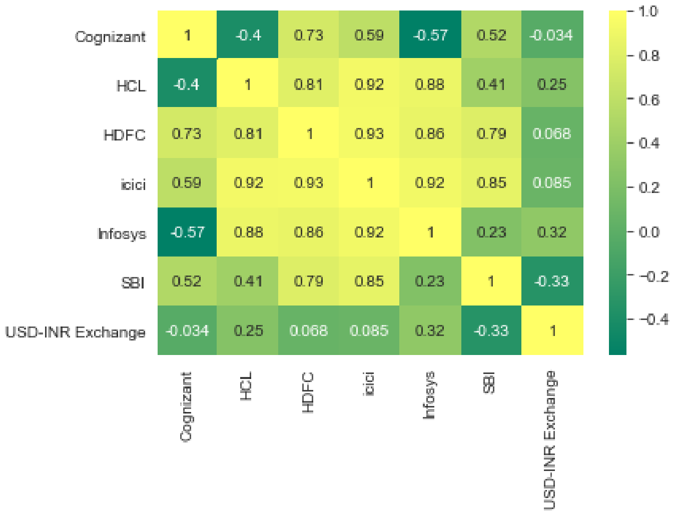


Fig. 11. Moving average plot - USD-INR exchange rate



	Cognizant	HCL	HDFC	icici	Infosys	SBI	USD-INR Exchange
Cognizant	1	-0.4	0.73	0.59	-0.57	0.52	-0.034
HCL	-0.4	1	0.81	0.92	0.88	0.41	0.25
HDFC	0.73	0.81	1	0.93	0.86	0.79	0.068
icici	0.59	0.92	0.93	1	0.92	0.85	0.085
Infosys	-0.57	0.88	0.86	0.92	1	0.23	0.32
SBI	0.52	0.41	0.79	0.85	0.23	1	-0.33
USD-INR Exchange	-0.034	0.25	0.068	0.085	0.32	-0.33	1

Fig. 12. Correlation matrix

- From Fig 2, a point of overtake can be seen between May and September 2021 where the stock price of Infosys goes past that of HDFC, which was the then leading company among this list.
- From Fig 3, we can see that the trends of major increase and decrease occur at the roughly the same time for all the companies. That is, all the companies have a major decrease in their stock prices around May 2020, which is obviously due to the **COVID-19** pandemic which shut down various commodities.

B. *Correlations*

The correlation matrix is plotted in Fig 12 and the pairplot in Fig 13 in order to analyze if one company affects the others as it happens in real life. From this, we can conclude that ICICI's stock prices are highly positively correlated with Infosys, HDFC and HCL. One of the reasons is because HDFC and ICICI are banking sector companies so they will be similar in growth. Also, even though SBI is in a similar sector, SBI is a public company which is a not-for-profit. Hence, the correlation with ICICI is not as high as Infosys and HCL as all the latter companies are private and aiming for profits. Also, the negative correlation between Infosys and Cognizant can be explained by the fact that they are competitors and so if one's price goes up, it'll have a direct effect on it's competitors.

VI. Machine Learning Models

Since the variable to be predicted is a continuous variable, it is natural that regression techniques will be used in this case. In all these models, 80% of the data will be used for training and 20% for validation. Since this is a stock prediction problem, it is best for analysis if the 1st 80% of the dates are used for training and the last 20% be used for validation as it symbloizes future prediction of stock prices using historical data. 3 different models are discussed here (Linear Regression, k-Nearest Neighbours Regressor and

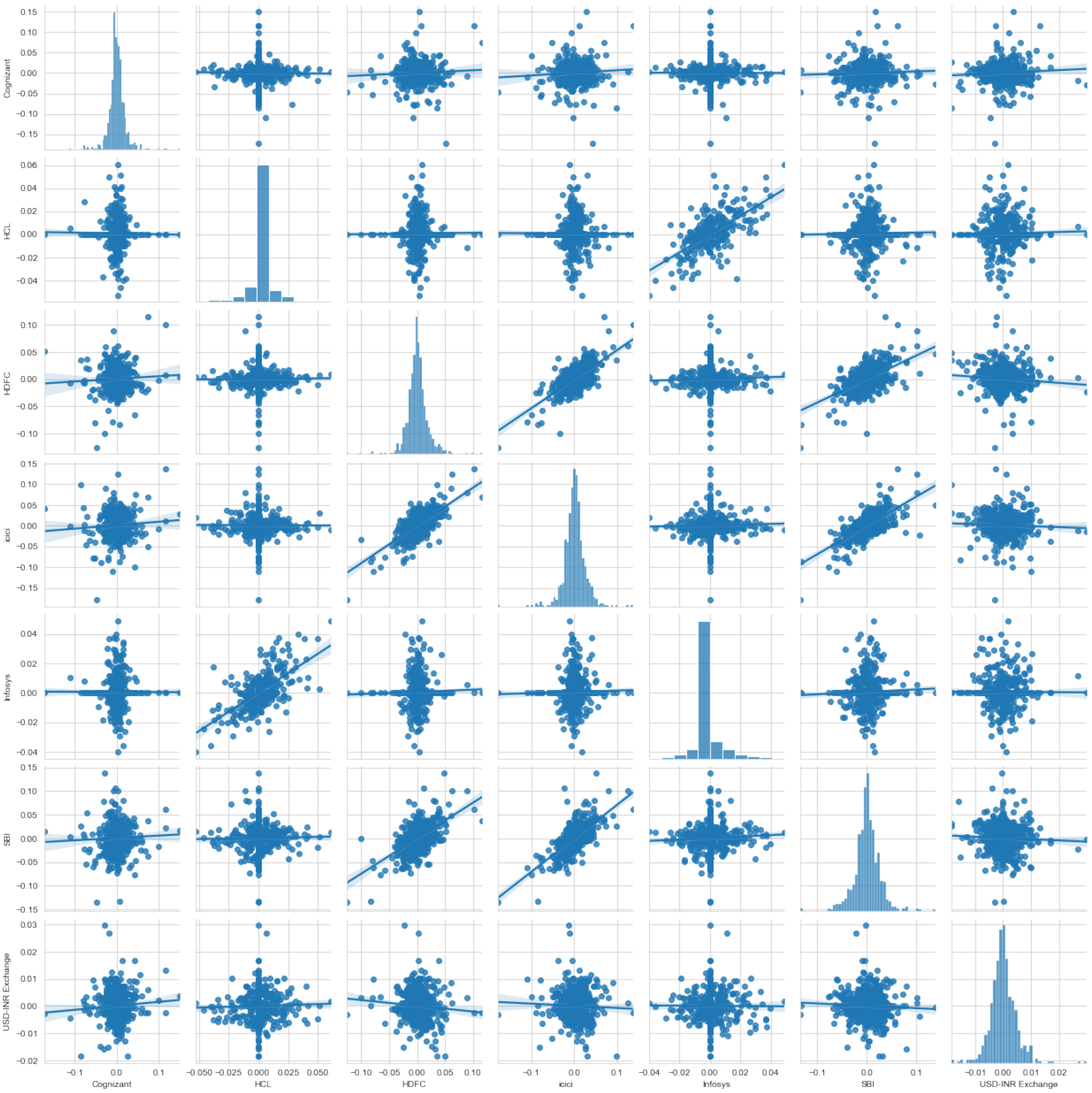


Fig. 13. Pair plot

Long-Short-Term-Memory(LSTM) Networks). The validation RMS error is used as the metric to determine the accuracy and closeness of the models.

A. Linear Regression

A linear regression model was created for each of the datasets and was trained using the training data. The inputs were the date, month, year. The target variable is the closing price on that day. This model was created for all the companies and the predictions of the validation data were plotted against the true values.

From Fig 14 to Fig 20, we can observe that Linear regressor is not a very good model for stock prediction. It captures the variations nicely in HDFC, but is way off the true values in every other plot apart from Cognizant.

This is not something very unexpected as we know that stock market fluctuations are very dynamic and it will definitely won't be covered by a linear model. Some degree of

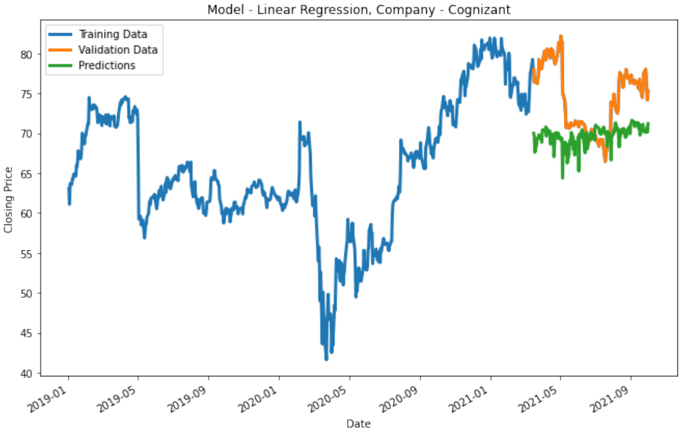


Fig. 14. Linear Regression Predictions - Cognizant

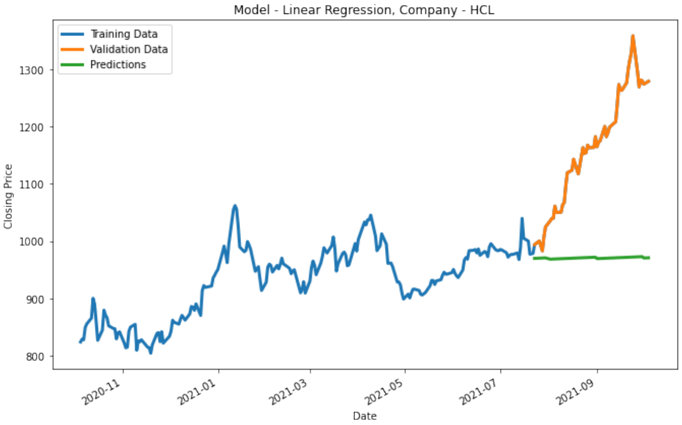


Fig. 15. Linear Regression Predictions - HCL

non-linearlity is necessary.

B. kNN Regressor

The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set. Fig 21 to Fig 27 gives the plots of kNN regression predictions for all the companies and the US-INR exchange rate. It is not much different than the linear regressor. **A possible reason for this is that Linear Regression and kNN work on interpolation whereas stock prediction ia more of an extrapolation task.** Hence, the accuracy is very less in this case.

VII. Deep Learning Models - LSTM RNN

A. Introduction

Since the machine learning methods are not giving satisfactory results, it is necessary to use something advanced for stock prediction as this deals with Time-Series data. A classic option is a deep learning algorithm like LSTM which is ideal for time series data like stock prices.

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM

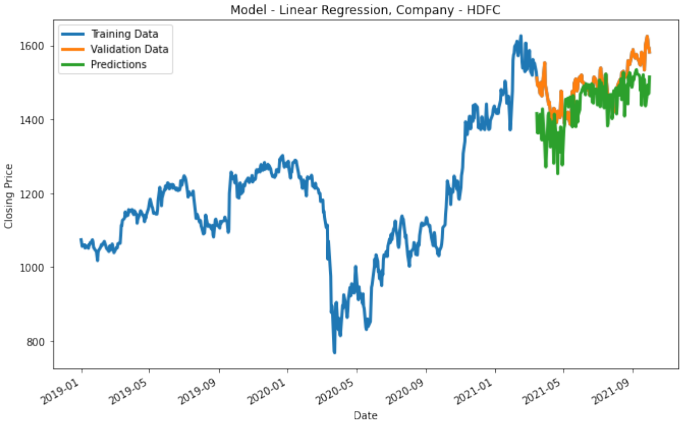


Fig. 16. Linear Regression Predictions - HDFC

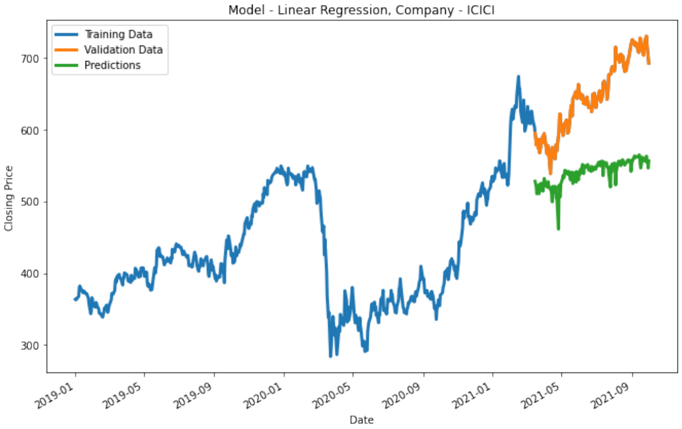


Fig. 17. Linear Regression Predictions - ICICI

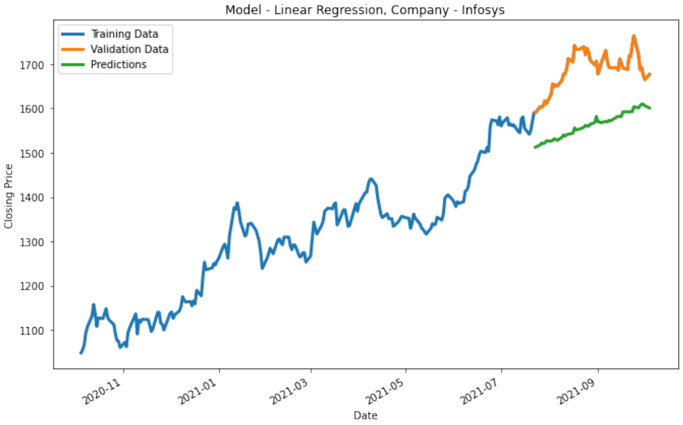


Fig. 18. Linear Regression Predictions - Infosys

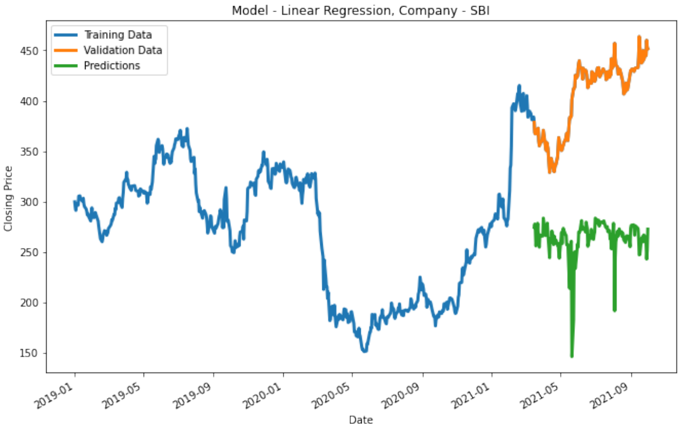


Fig. 19. Linear Regression Predictions - SBI

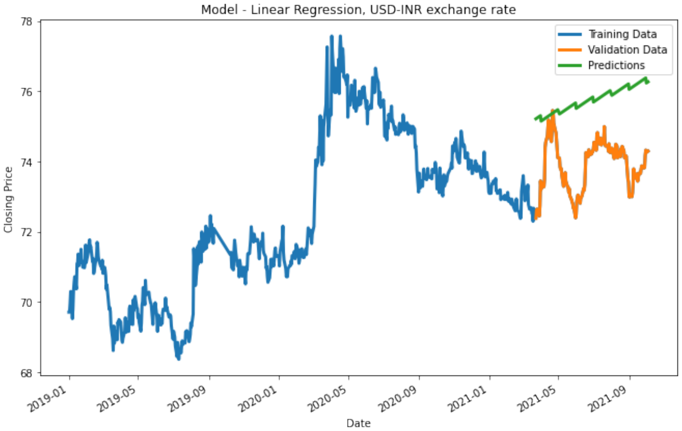


Fig. 20. Linear Regression Predictions - USD-INR rate

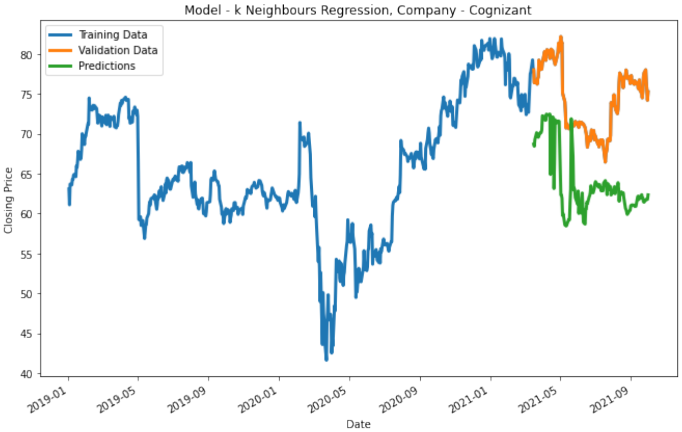


Fig. 21. k Nearest Regression Predictions - Cognizant

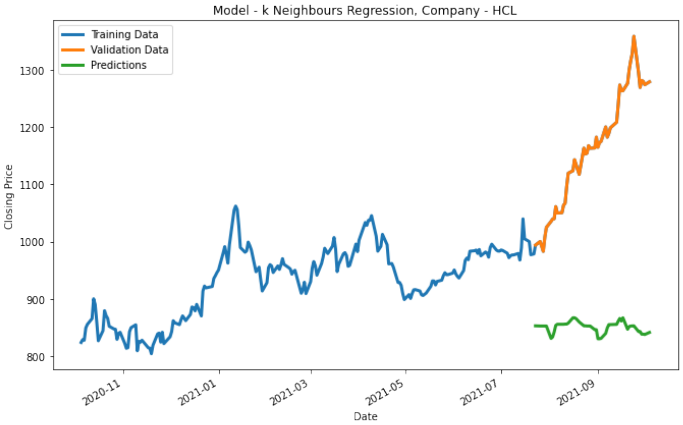


Fig. 22. k Nearest Regression Predictions - HCL

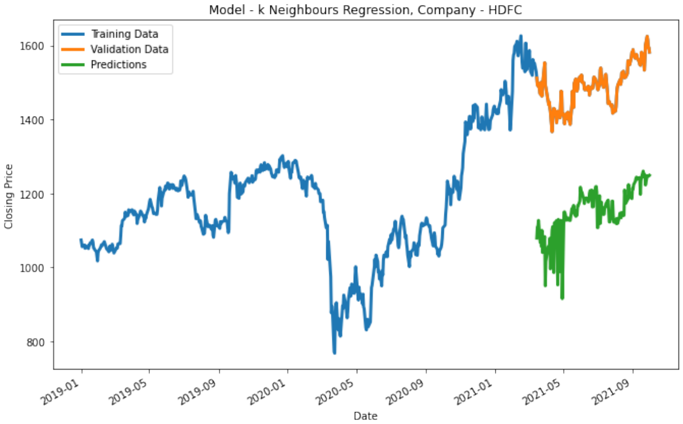


Fig. 23. k Nearest Regression Predictions - HDFC

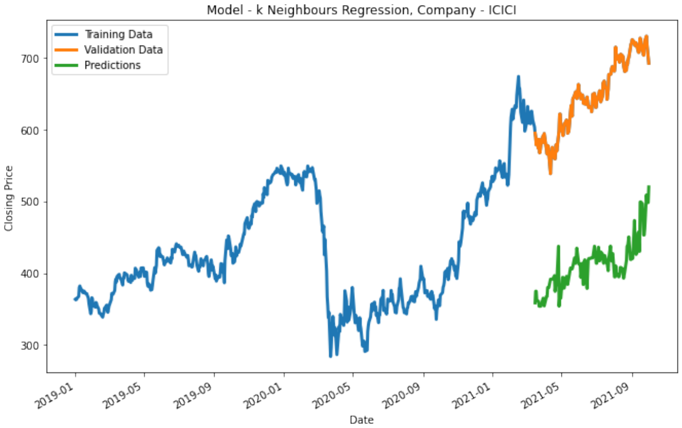


Fig. 24. k Nearest Regression Predictions - ICICI

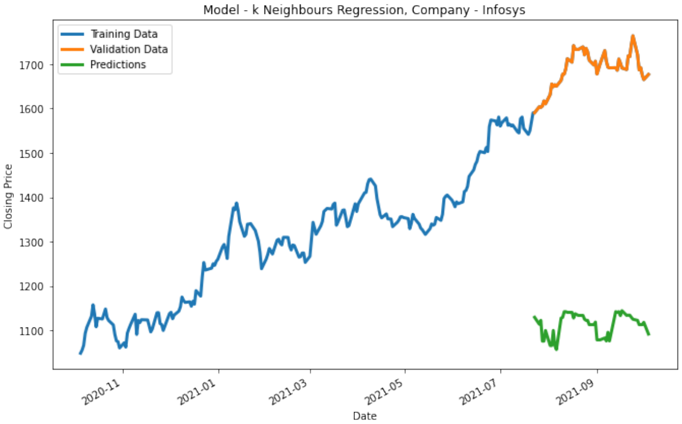


Fig. 25. k Nearest Regression Predictions - Infosys

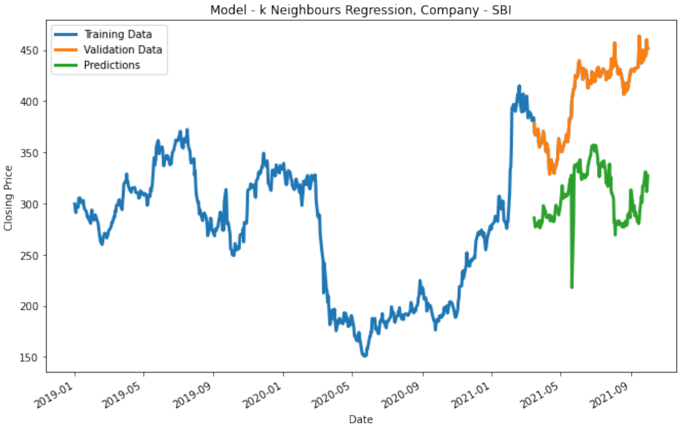


Fig. 26. k Nearest Regression Predictions - SBI

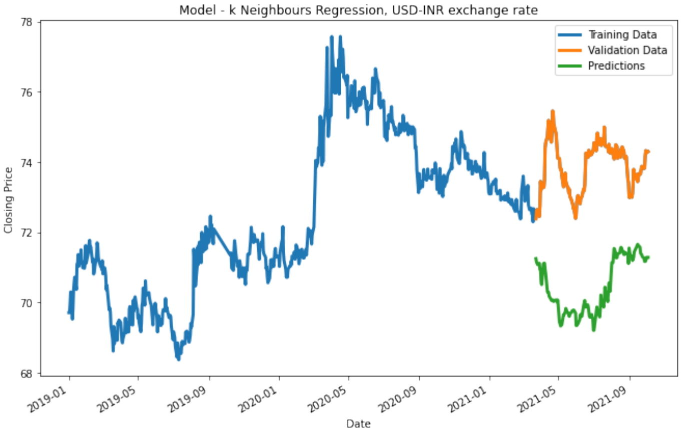


Fig. 27. k Nearest Regression Predictions - USD-INR rate

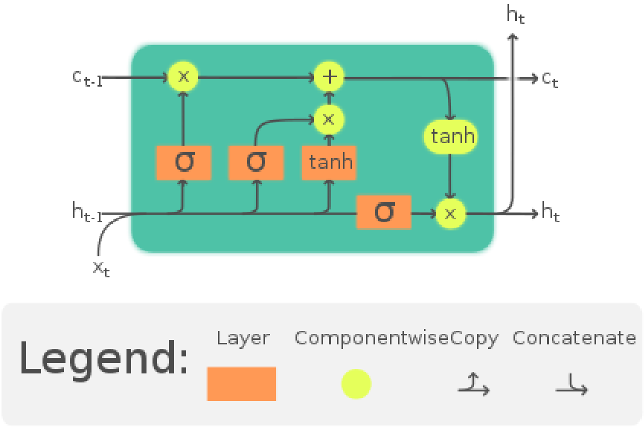


Fig. 28. LSTM - Model Structure

has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

B. Model Structure and Mathematics

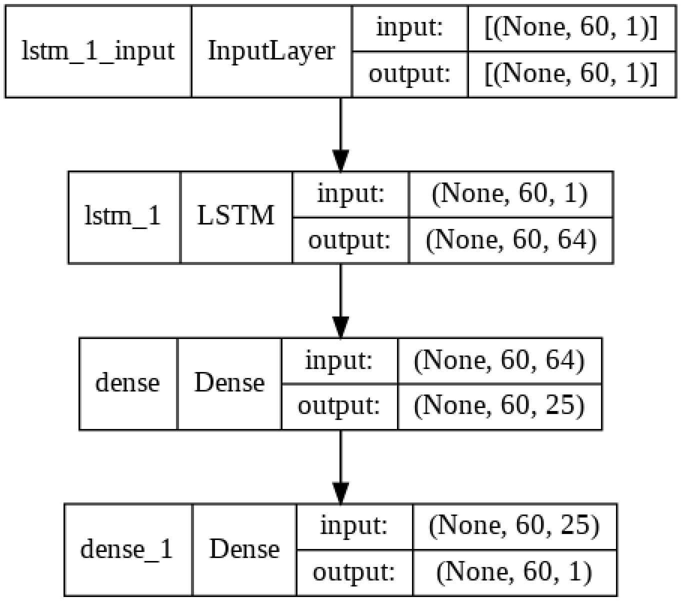
A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

The compact forms of the equations for the forward pass of an LSTM cell with a forget gate are:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\circ$ denotes the Hadamard product (element-wise product). The subscript t indexes the time step. The variables and attributes are given as follows:

- $\vec{x_t} \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in {(0,1)}^h$: forget gate's activation vector
- $i_t \in {(0,1)}^h$: input/update gate's activation vector



lstm_1_input	InputLayer	input:	[(None, 60, 1)]
		output:	[(None, 60, 1)]
lstm_1	LSTM	input:	(None, 60, 1)
		output:	(None, 60, 64)
dense	Dense	input:	(None, 60, 64)
		output:	(None, 60, 25)
dense_1	Dense	input:	(None, 60, 25)
		output:	(None, 60, 1)

Fig. 29. LSTM - Model Architecture

- $o_t \in (0,1)^h$: output gate's activation vector
- $h_t \in (-1,1)^h$; hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in (-1,1)^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h\times d}, U \in \mathbb{R}^{h\times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training
- σ_g : sigmoid function
- σ_c : hyperbolic tangent function

where the superscripts d and h refer to the number of input features and number of hidden units, respectively.

C. LSTM Model Creation and Predictions

Since this topic is out of the scope for this course, a simple LSTM with 1 hidden layer with 64 memory blocks is taken for this case. In this case the model layout for all the companies is shown in Fig 29.

LSTM models like above were created for all the datasets and trained using their respective training data. All these predictions are plotted from Fig 30 to Fig 36. From these plots, we can clearly see that the presence of a hidden layer greatly takes the model very close to the true values. This is why Deep Learning Models are preferred for Time-Series data than standard ML models.

However, even in this, HCL seems to be far from ideal (Fig 31. This is due to the fact that the number of training sampled in the HCL dataset(150) is much less than the others(500). This can be improved either by getting more samples or by making the LSTM more advanced. Since the latter is not in the scope of this course, it is chosen to leave it at this point.

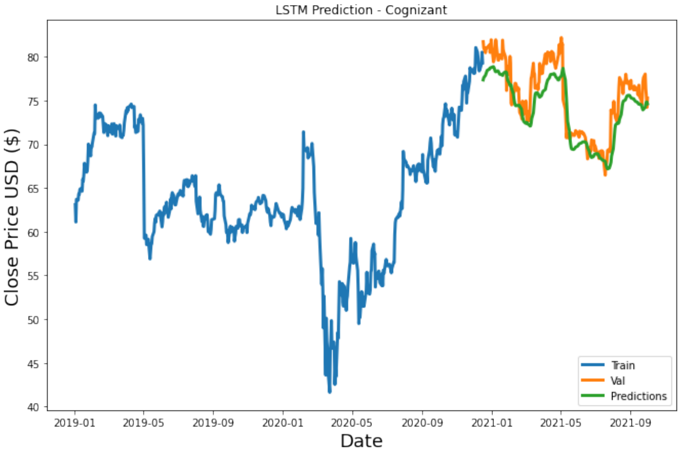


Fig. 30. LSTM Predictions - Cognizant

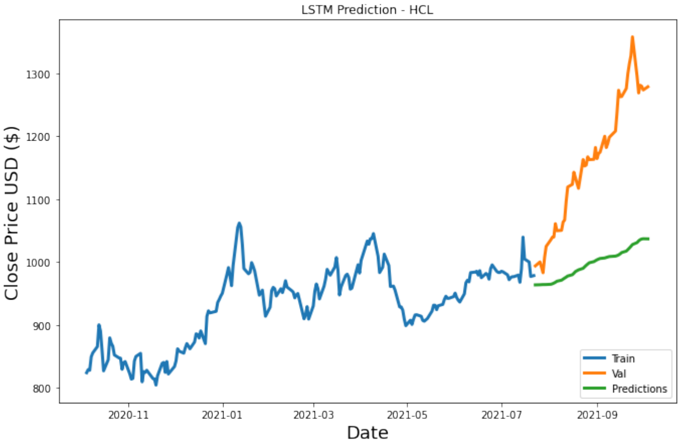


Fig. 31. LSTM Predictions - HCL

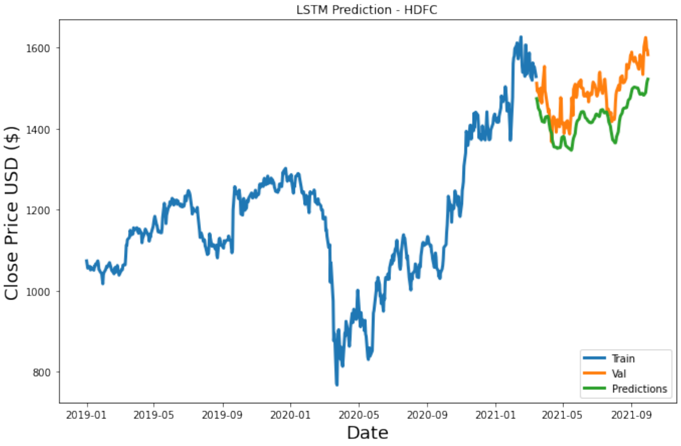


Fig. 32. LSTM Predictions - HDFC

TABLE II
RMS ERRORS

Model	Cognizant	HCL	HDFC	ICICI	Infosys	SBI	USD-INR rate
Linear Regression	6.18	215.87	75.29	114.37	128.74	148.52	2.02
kNN	9.67	327.22	341.93	238.94	575.49	104.81	3.58
LSTM	3.21	141.96	55.22	15.48	88.20	13.56	0.77

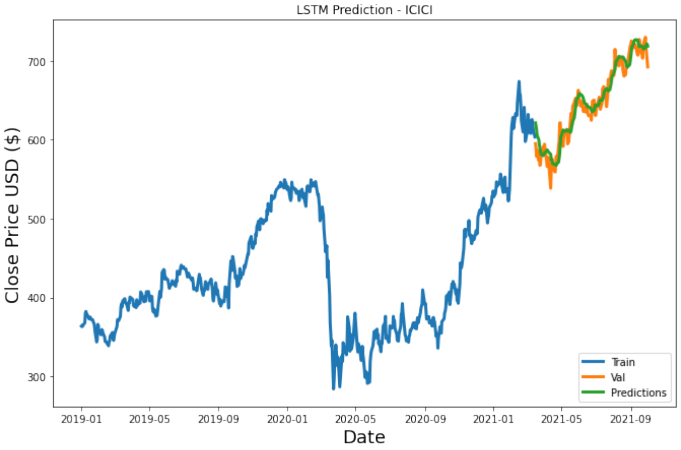


Fig. 33. LSTM Predictions - ICICI

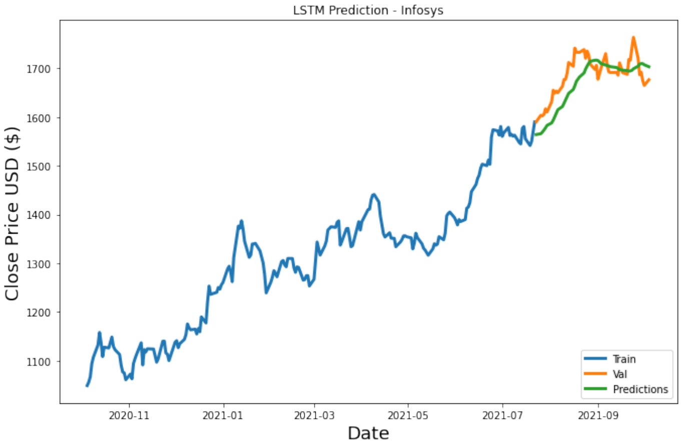


Fig. 34. LSTM Predictions - Infosys

VIII. Comparison between models

A detailed table of the RMS errors of all the models with all the datasets is shown in Table II. From the table, and all the plots, it is clear the LSTM is the best possible model among the 3 by a large margin. Among, kNN and Linear Regression, it is observed that Linear Regression gives a lesser error than kNN. This might be because kNN will be searching for classes in the data which means it will be more appropriate for an interpolation problem and hence, it does not give satsfactory results for Time-Series data such as this.

Therefore, LSTM is chosen to be the final model.

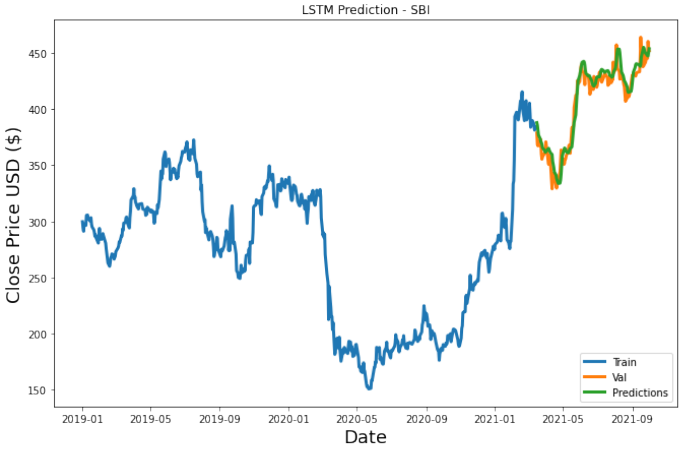


Fig. 35. LSTM Predictions - SBI

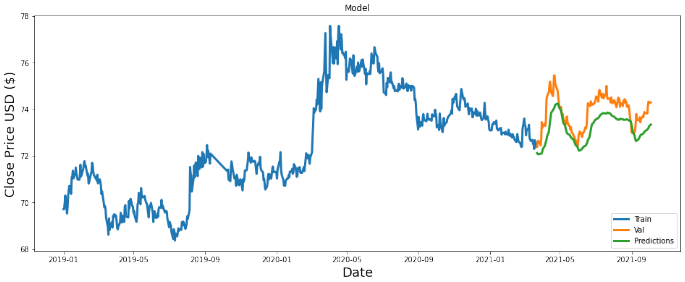


Fig. 36. LSTM Predictions - USD-INR rate

IX. Conclusion

The datasets containing stock prices of various companies and US-INR exchange rate was analyzed in great detail. Various conclusions were made based on trends observed by using techniques such as moving average method. Machine Learning and Deep Learning models were created and trained for these datasets and were validated using future data, hence, successfully creating a model for future stock prediction based on the history of closing prices. Further scope of this paper can be to investigate in more Deep Learning techniques for creating models. Also, the hyperparameters of the current LSTM model can be more effectively tuned to fit each dataset's requirements.

References

[1] Dr. Kaushik Mitra, Dr. Ramkrishna Pasumarthy, Data Analytics Laboratory Slides
[2] LSTM. Wikipedia