CAPSTONE PROJECT **ROHIT SINHA**

MACHINE LEARNING NANODEGREE **1st May 2018**

DEFINITION

Project Overview

The capstone project is to build a stock price indicator using Python. The investment and trading industry makes use of different tools to predict stock prices (Fig 1). Technical analysis is one of the means to evaluate stock prices using past prices and volume. Time series forecasting using ARIMA can also be used to predict future stock prices. The project evaluates machine learning techniques to predict daily stock price movement using time series modelling and features created from technical indicators.

In the first part of the project we evaluate time series forecasting for stock prices to lay the ground work for more sophisticated machine learning models. We explore stationarity, autocorrelation and differencing of the time series data. Using these concepts we create a naïve model to predict stock prices based on ARIMA modelling.

The second part of the project involves creating features from the raw time series data of stock prices. We introduce new features in the form of technical indicators, which will be used to predict the stock price.

The final part of the project involves creating a trading strategy on the basis of new features created. We create a machine learning model to predict the stock price movement and create a Long-Short strategy on the basis of our predictions. We then compare our results to a Long only strategy.

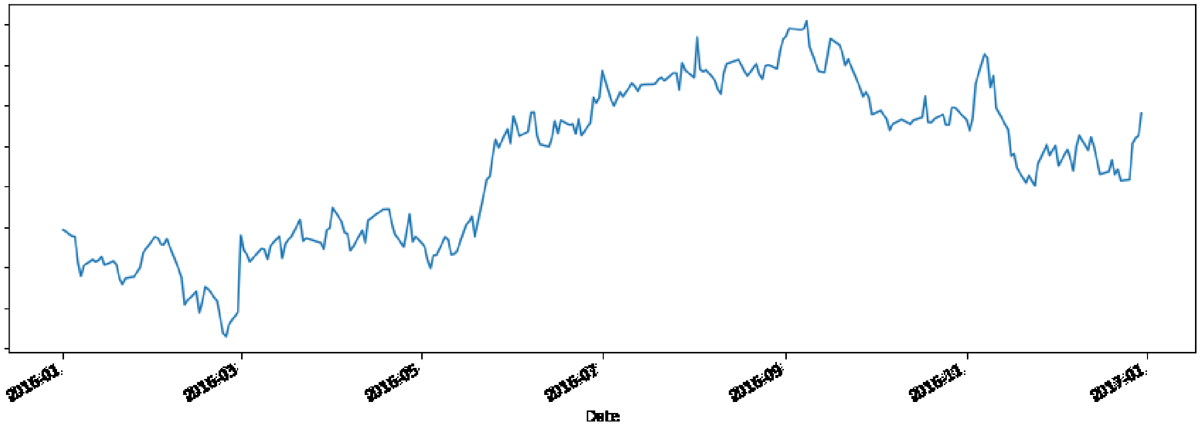


Fig 1. STOCK PRICE MOVEMENT

Problem Statement

The goal is to create a profitable stock investing strategy using Machine Learning. The tasks involved are the following:

- Download the stock prices
- Evaluate the stationarity of the data
- Create a naïve stock predictor using ARIMA modelling
- Create a stock predicting model using Logistic Regression and Neural Network
- Create a Long-Short strategy using the Logistic Regression and Neural Network models
- Compare the Long-Short strategy performance vs a Long only strategy
- Compare the predictive power of the model over short, medium and long term horizons

Metrics

- We evaluate stationarity of the time series using Dickey-Fuller test
- We evaluate autocorrelation using ACF, PACF plots
- For the naïve ARIMA model we evaluate our models using RMSE (root mean squared error)
- For the Logistic Regression and Neural Network model, we evaluate our strategy using the Annualized Returns and Volatility of our Long Short strategy vs the Long only Strategy
- We have also evaluated the Accuracy of the model but Returns and Volatility serve as a better measure of performance

ANALYSIS

Data Exploration

The data has been downloaded from Yahoo finance. (https://in.finance.yahoo.com/)

We have downloaded data for three stock listed in the National Stock Exchange of India (NSE):

- INFOSYS (INFY.NS)
- HDFC BANK (HDFCBANK.NS)
- ITC LIMITED (ITC.NS)

The three stocks belong to different sectors, so that we remove any sector bias in our study. The data has been downloaded from 2000 onwards.

The features available from the time series are the OPEN, HIGH, LOW and CLOSE prices. We create new technical factors using the existing features. The existing features will be highly correlated with each other (Fig 2) and hence we create new features which are not correlated (Fig 3) with each other and perform a better task at prediction.

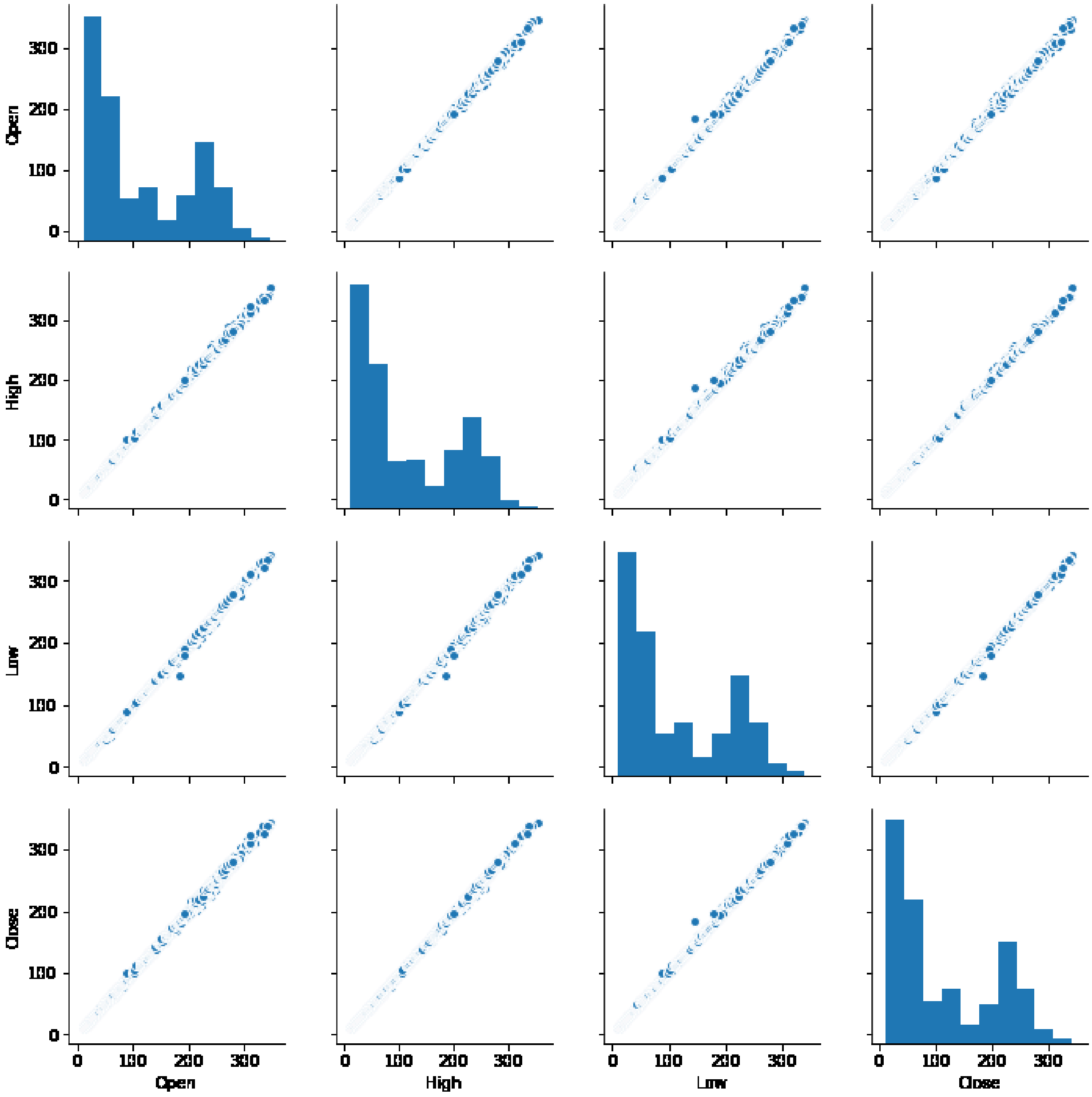


Fig 2. PAIRPLOT OF ORIGINAL FEATURES

The new technical features created are:

- High minus Low (H-L)
- Open minus Close (O-C)
- Relative Strength Indicator (RSI)
- Average Directional Index (ADX)
- Parabolic SAR
- Williams Percentage Range (Williams-R)

The correlation matrix of the above technical indicators is shown in Fig 4. The technical indicators are used to predict the next day's price movement (Up or Down).

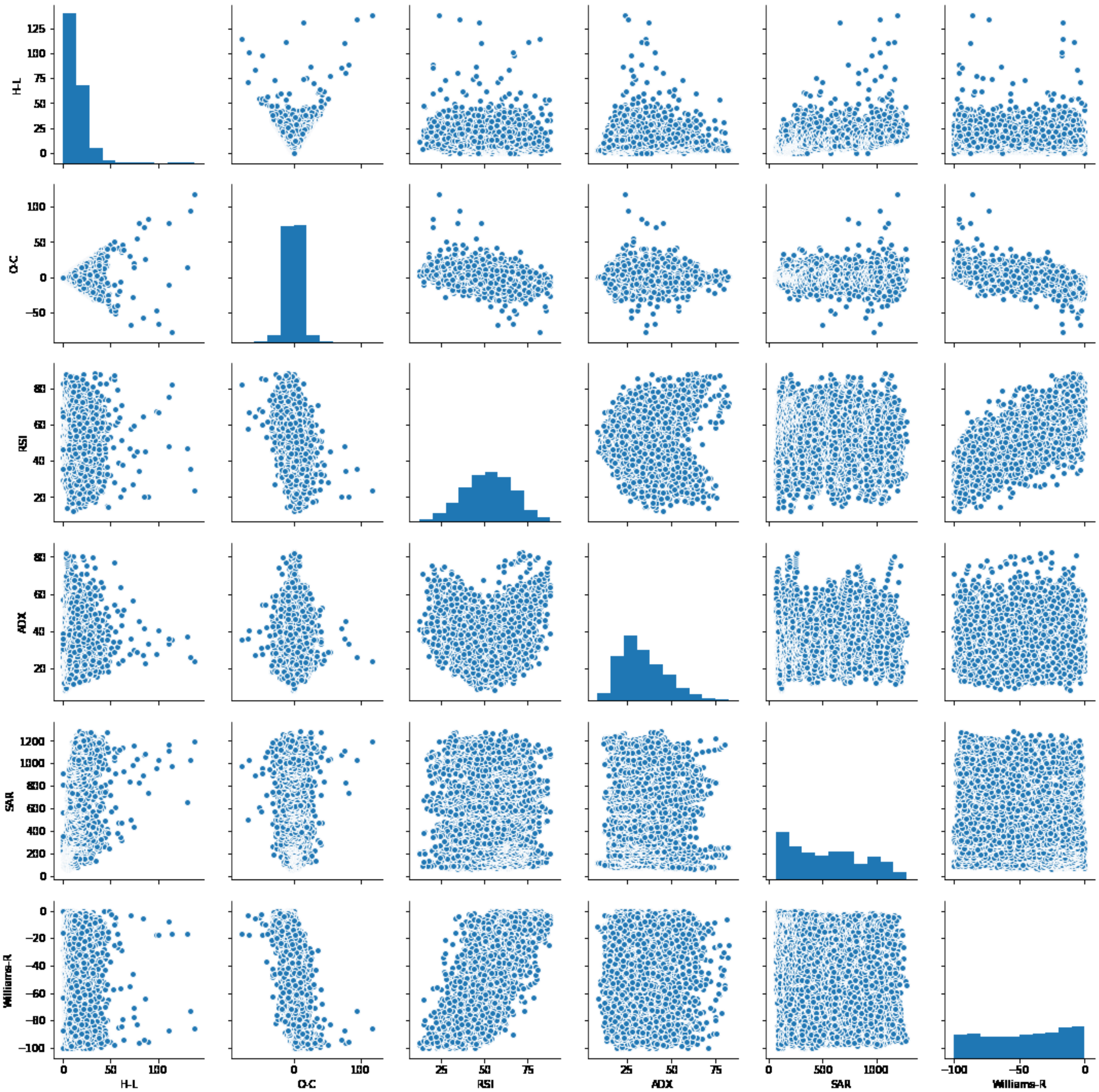


Fig 3. PAIRPLOT OF TECHNICAL INDICATORS

I would like to acknowledge the use of *TA-Lib* Python library from Quantopian to generate the technical indicators (http://www.ta-lib.org/).

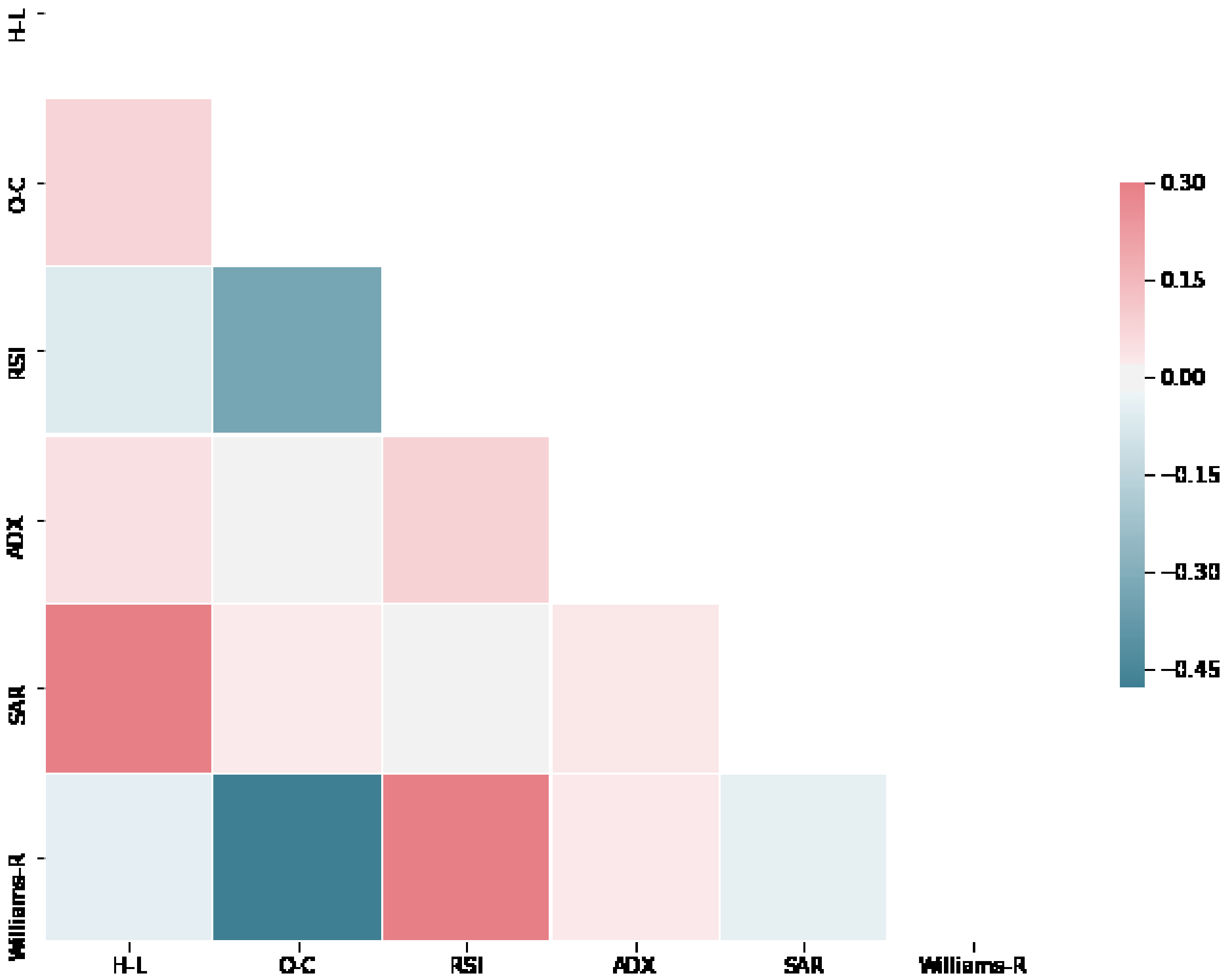


Fig 4. CORRELATION MATRIX OF TECHNICAL INDICATORS

Algorithm and Techniques

DICKY-FULLER Test is used to check for the stationarity of the data. A stationary time series is one which has a constant mean and variance and no trend and autocorrelation exists in the time series.

The actual time series of stock prices is not stationary but if we transform the data and take daily difference, the new resulting series is stationary. Prediction is correct only on stationary data because the mean and variance of the series does not vary over time. *Hence we should always forecast a change in the stock price rather than the absolute stock price due to the stationary nature of the daily difference series.*

We have plotted below the time series of daily difference of log transformed stock prices. The test statistic for DICKY-FULLER test rejects the null hypothesis and suggests that the data is stationary in all the three log transformed daily difference series as shown by the constant mean and standard deviation of the series.

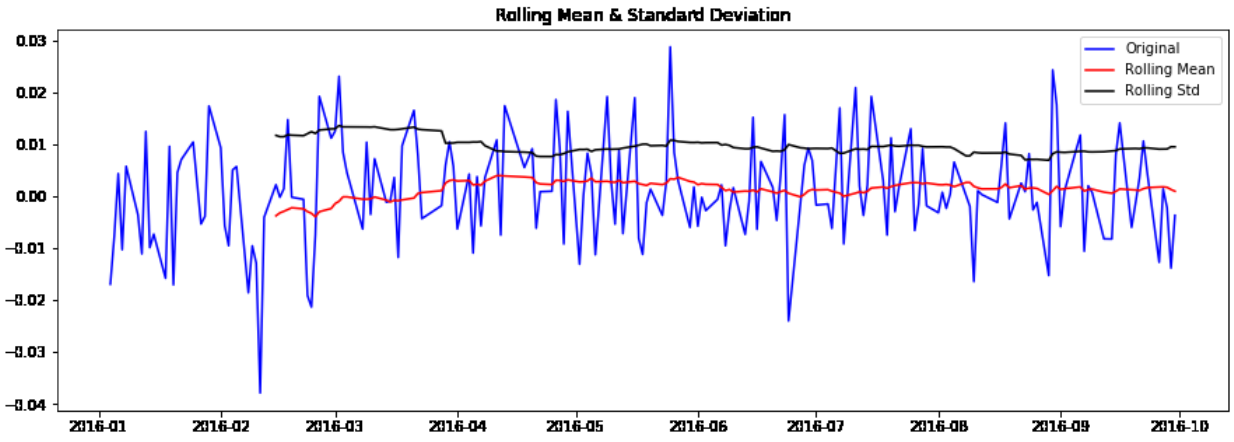


Fig 5. Stationarity of the Log Transformed Daily Difference HDFC Price

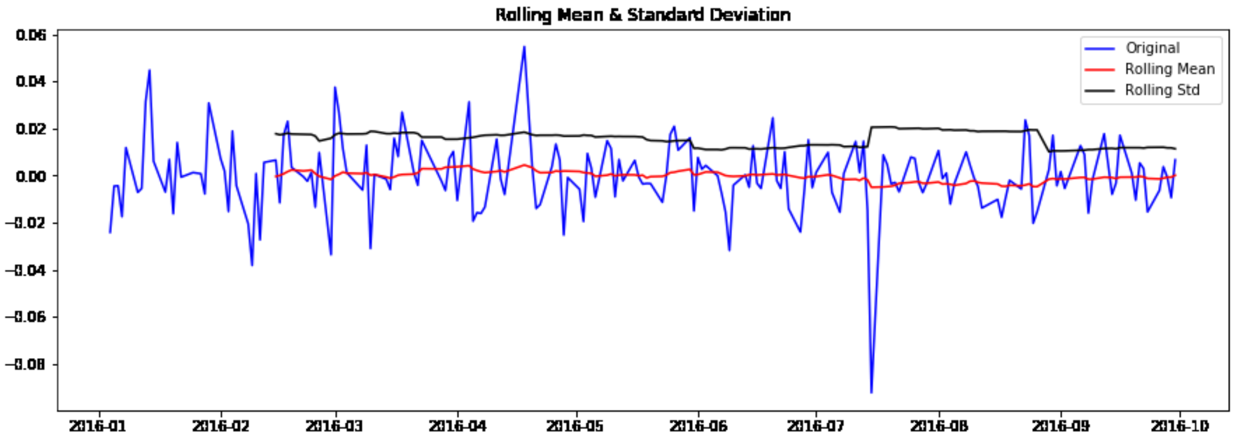


Fig 6. Stationarity of the Log Transformed Daily Difference INFOSYS Price

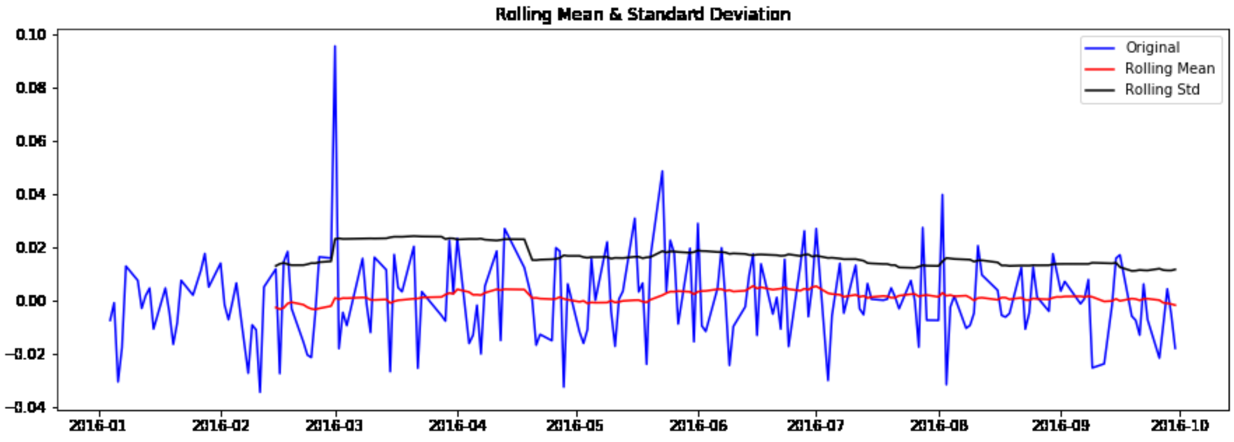


Fig 7. Stationarity of the Log Transformed Daily Difference ITC Price

The ACF and PACF plots for the time log transformed daily time series of the three stocks also suggests that no significant autocorrelation or partial autocorrelation exists for any lags. *Hence it is safe to predict the change of stock prices for any lag (T+1, T+2 etc).*

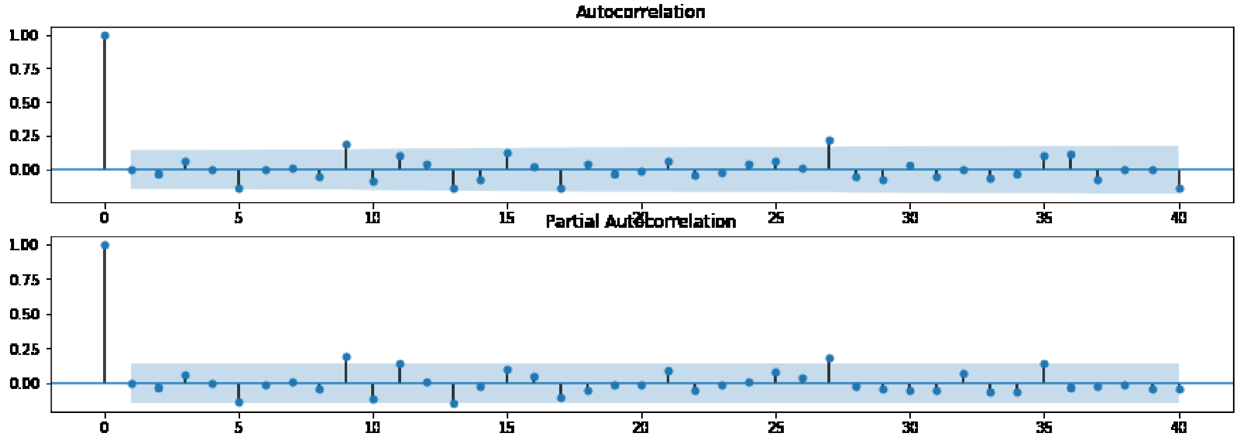


Fig 8. ACF and PACF plot of the Log Transformed Daily Difference HDFC Price

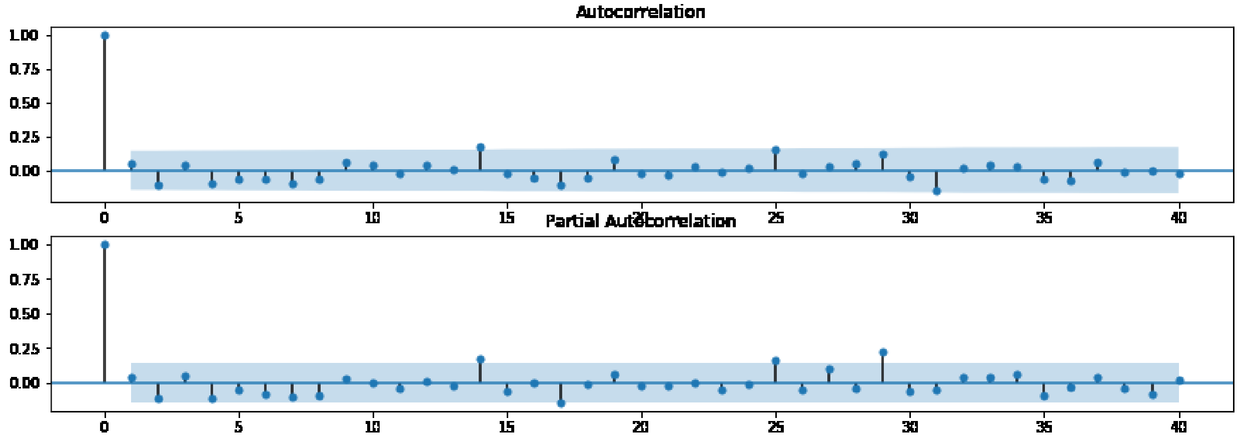


Fig 9. ACF and PACF plot of the Log Transformed Daily Difference INFOSYS Price

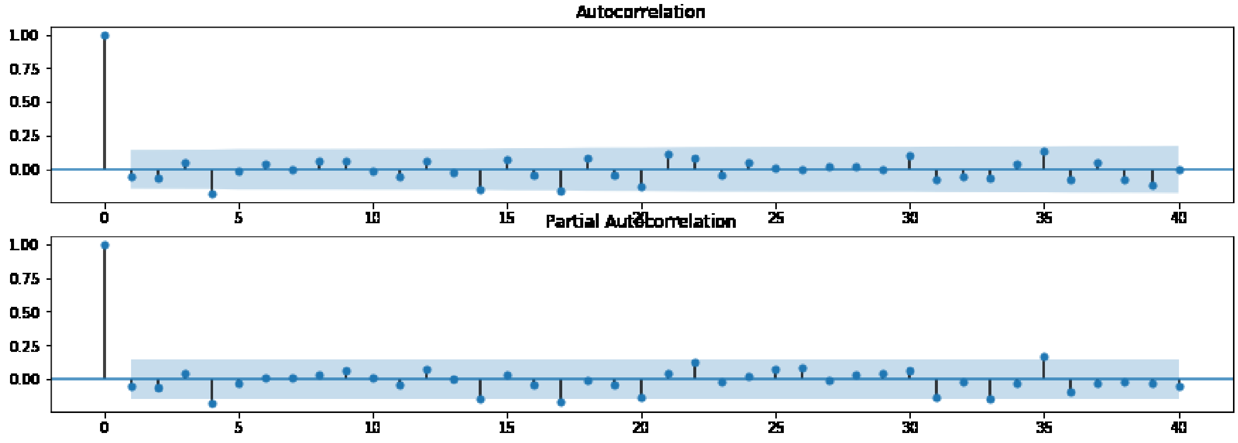


Fig 10. ACF and PACF plot of the Log Transformed Daily Difference ITC Price

Once we are sure about the stationarity of the data, we try to predict stock prices using a naïve ARIMA (2, 1, 1) model. We have only used data for 2016 to train and test the model. The out of sample results are from October 2016 to Dec 2016, while the model is trained from January 2016 to September 2016. *ARIMA model is a linear equation dependent on number of auto regressive terms, number of terms in moving average and the order of differencing. As we will see in our results below, ARIMA is used to create a very naïve model which close to a linear equation.*

The tools for time series analysis are available in the *stasmodels* package of Python.

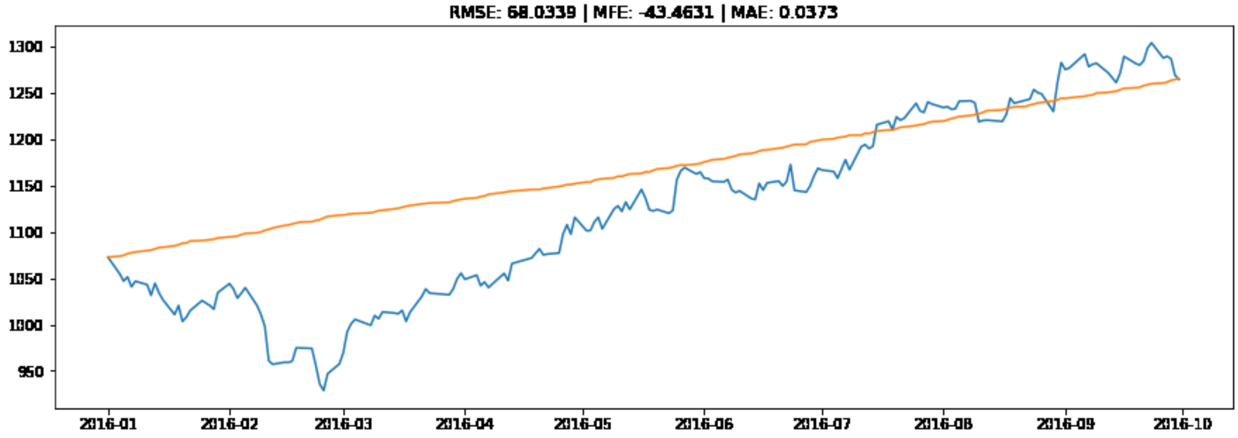


Fig 10. ARIMA Model on HDFC Training Data

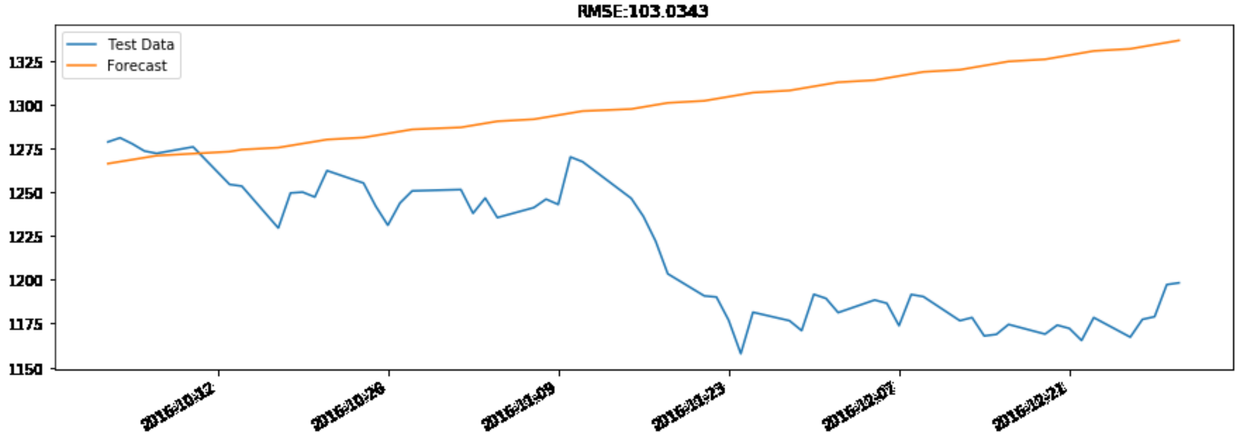


Fig 11. Predictions based on ARIMA model for HDFC

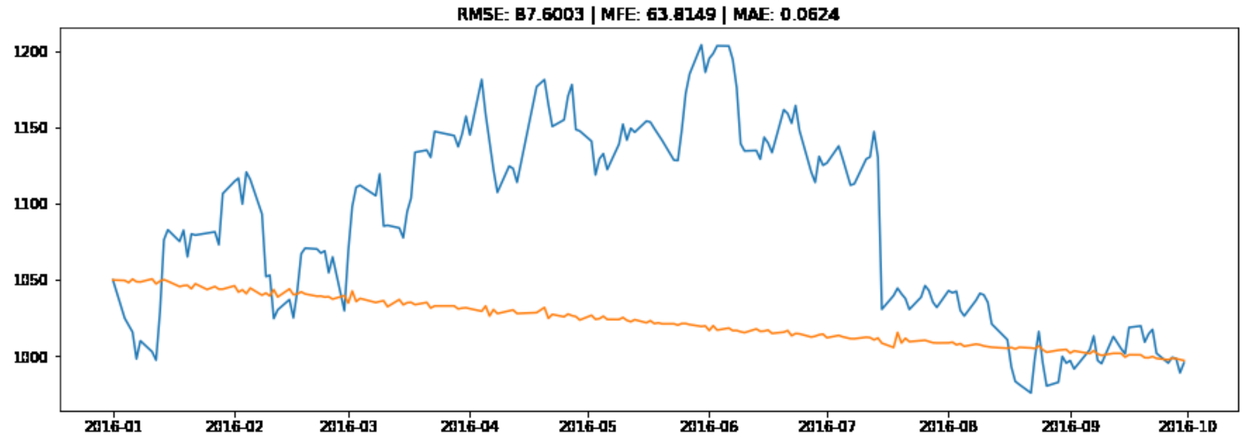


Fig 12. ARIMA Model on INFOSYS Training Data

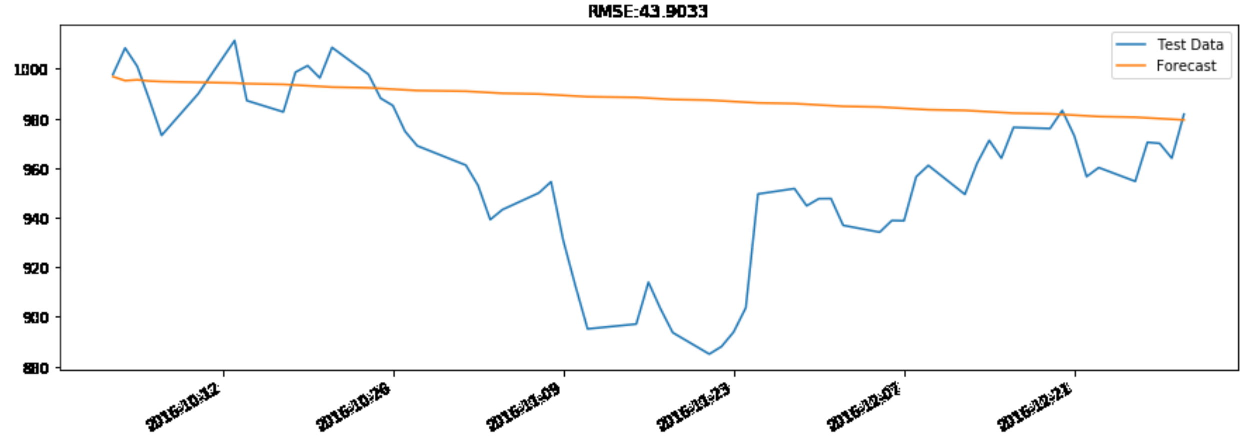


Fig 13. Predictions based on ARIMA model for INFOSYS

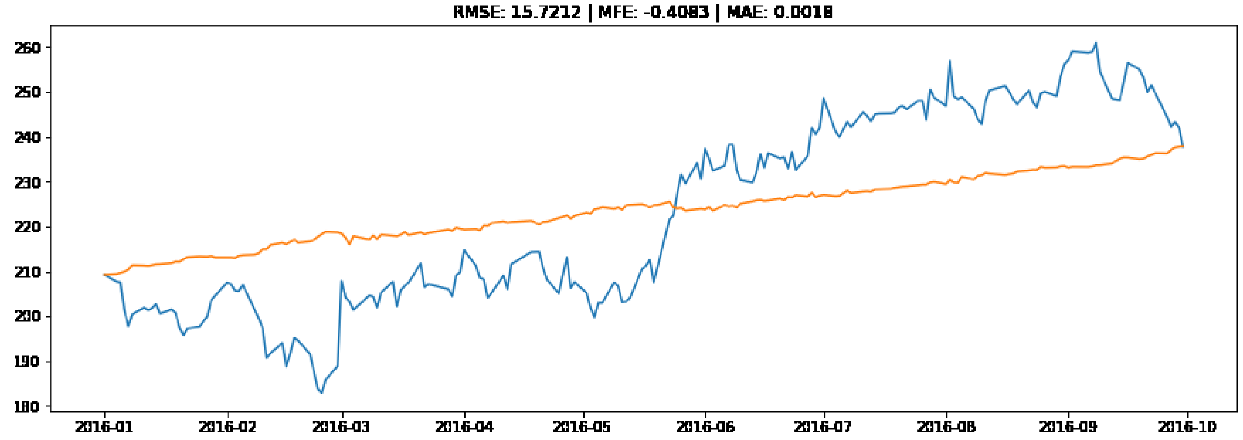


Fig 14. ARIMA Model on ITC Training Data

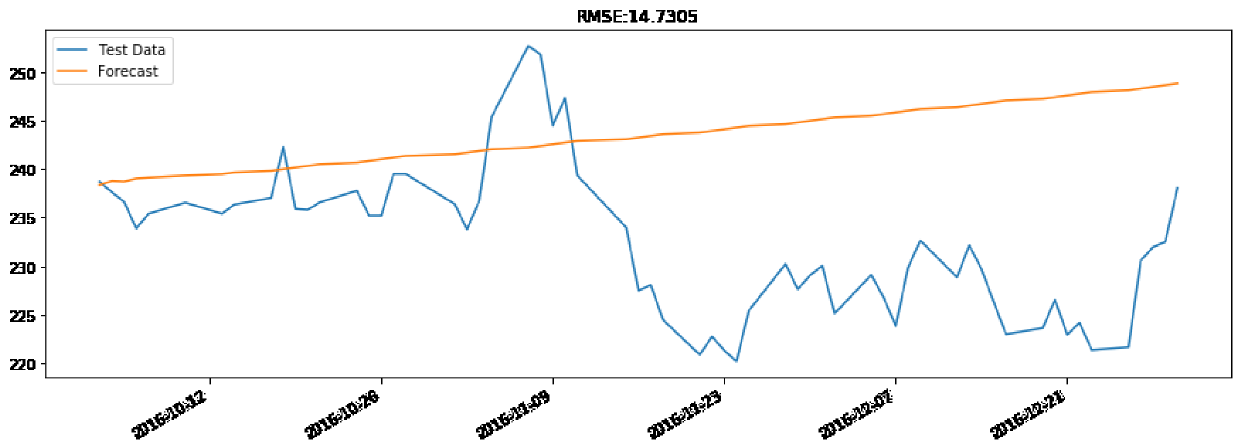


Fig 15. Predictions based on ARIMA model for ITC

The ARIMA modelling approach shows that:

- It is important to predict the change in stock price and not the level itself
- Time Series Forecasting can be used to create Naïve models which can be used to predict recent stock change but not fit for creating a sophisticated trading strategy

Keeping the above points in mind, we move from time series modelling to supervised machine learning. We have used two techniques to create a model which predict the stock change for the next day.

- Logistic Regression using *sklearn.linear_model* library
- Neural Network using *Keras* library

The independent features are the technical factors while the dependent variable is a binary variable which is 1 when Stock has gone up for the next day while 0 when the stock has gone down.

The detailed methodology for the supervised learning models is given in the next section, along with a discussion about the Long-Short strategy on the basis of the models created.

Methodology

As discussed earlier our independent variables are based on technical indicators while our dependent variable is based on our closing price. Before training the model we preprocess our dataset to make the mean of all independent variables equal to 0 and variance to 1. This ensures that there is no bias while training the model.

If next day closing price is greater than today, our dependent variable is 1 while if it is lower, our dependent variable is 0.

- Logistic Regression – It estimates the value of the model coefficients using the Sigmoid function. If the value of the output is greater than 0.5, we classify the outcome as 1 (TRUE) or else 0 (FALSE).
- Neural Network – We use *Sequential* method from the *Keras* library. We add layers to the neural network with Sigmoid function as the activation function of the output layer. If the

value of the prediction is greater than 0.5, we classify the outcome as 1 (TRUE) or else 0 (FALSE)

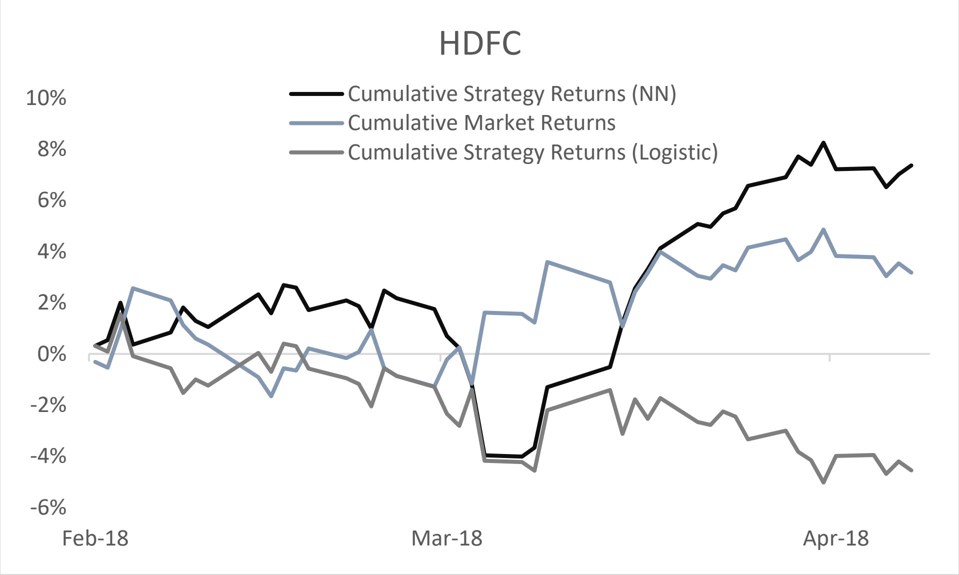
The output of the test dataset gives us the prediction to go long (1) the stock or go short the stock (0). Using the daily returns we can compute the cumulative returns of the Long-Short Strategy vs the Long only strategy. The daily returns also help us compute the annualized return and volatility for the strategies.

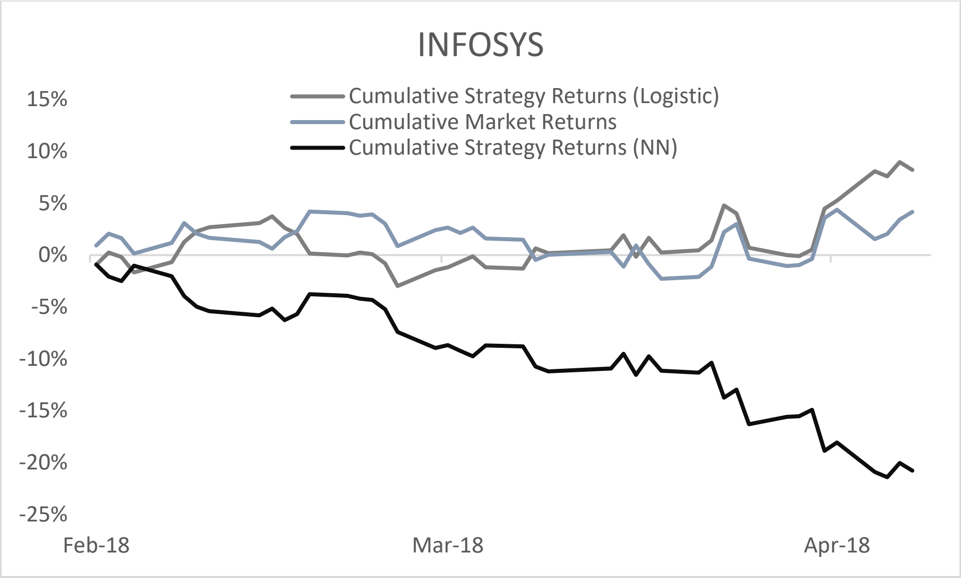
The results are compared for test sizes of 1%, 5% and 20% to confirm if the strategy is successful over short term or long term. Since it is a time series data, test data starts from the end of the training data.

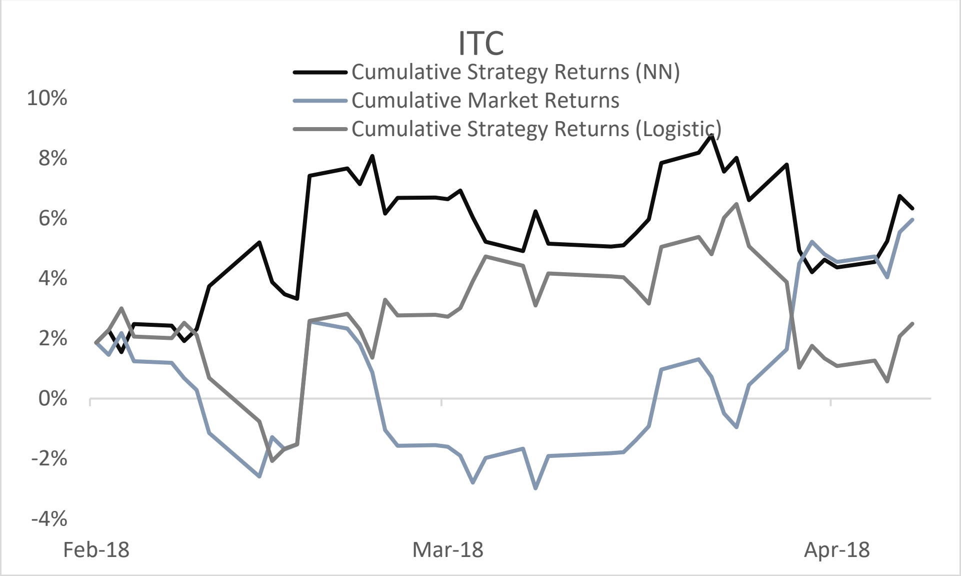
RESULTS

We compare the results of the Long Short strategy based on Neural Network and Logistic Regression vs Long Only Strategy for Test Size of 1%, 5% and 20%.

Test Size 1%

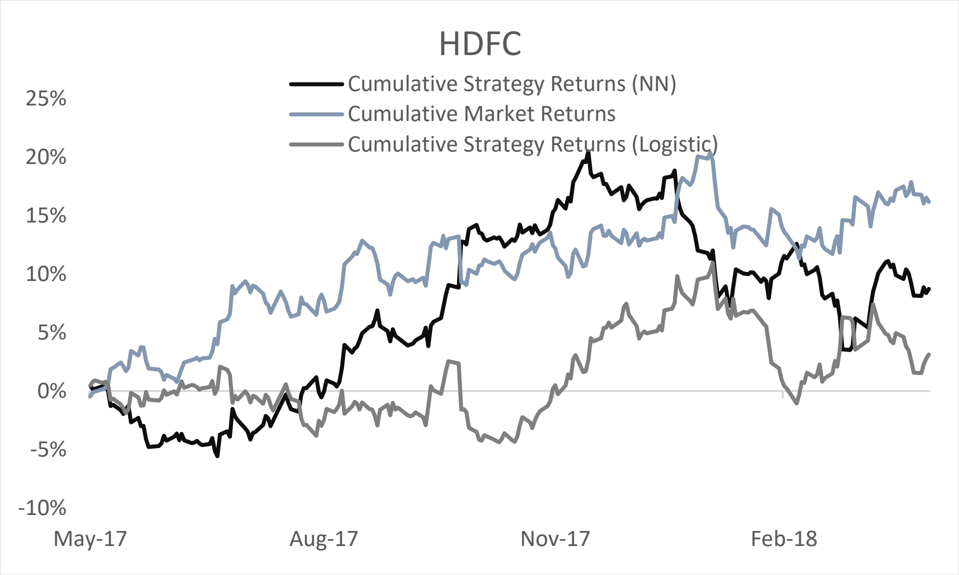


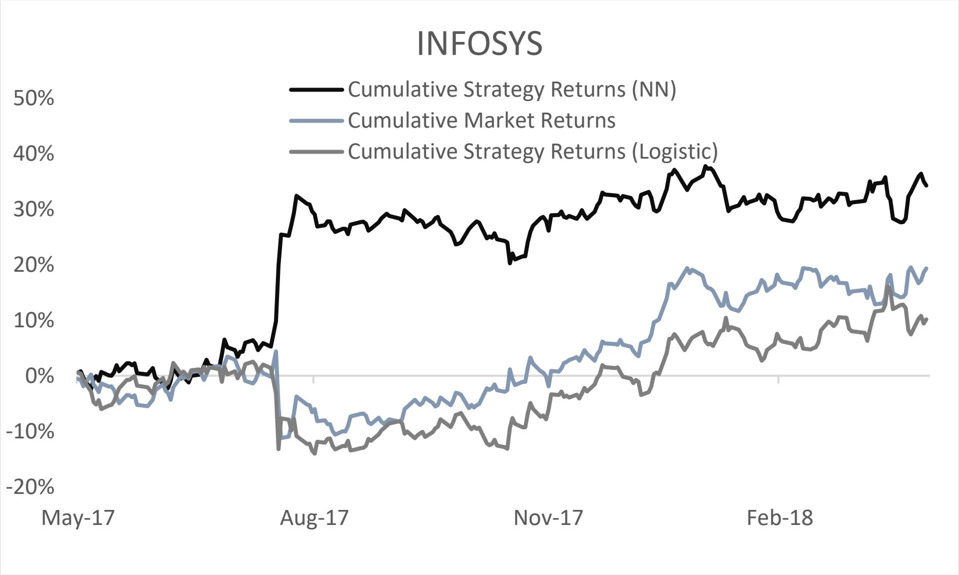


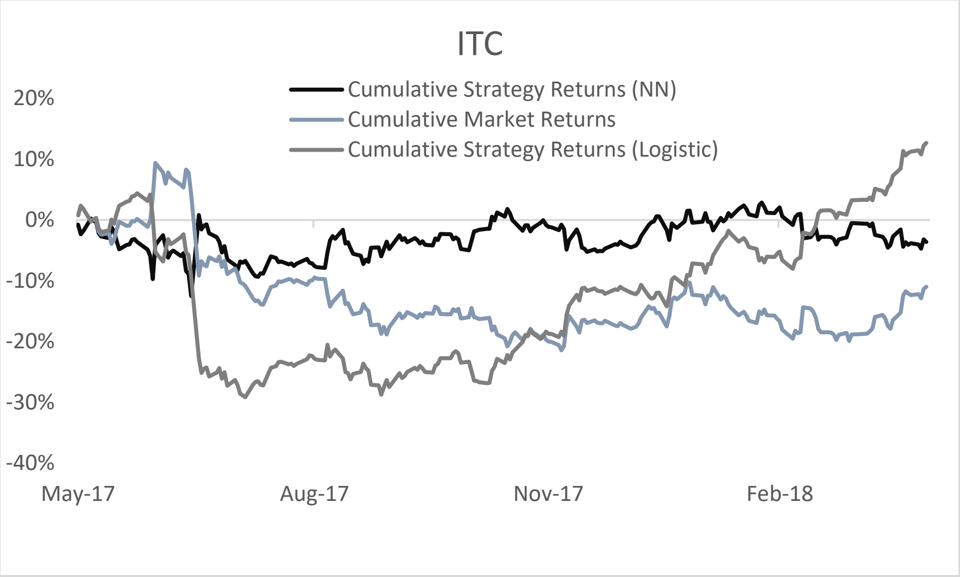


	Return vs Vol (Annualized)		
	HDFC	**INFY**	**ITC**
Return (NN)	48.9%	-72.9%	41.0%
Return (Logistic)	-23.0%	55.7%	14.7%
Return (Market)	19.2%	25.6%	38.2%
Volatility (NN)	15.3%	21.6%	18.1%
Volatility (Logistic)	15.4%	22.7%	18.2%
Volatility (Market)	15.5%	22.8%	18.1%

Test Size 5%

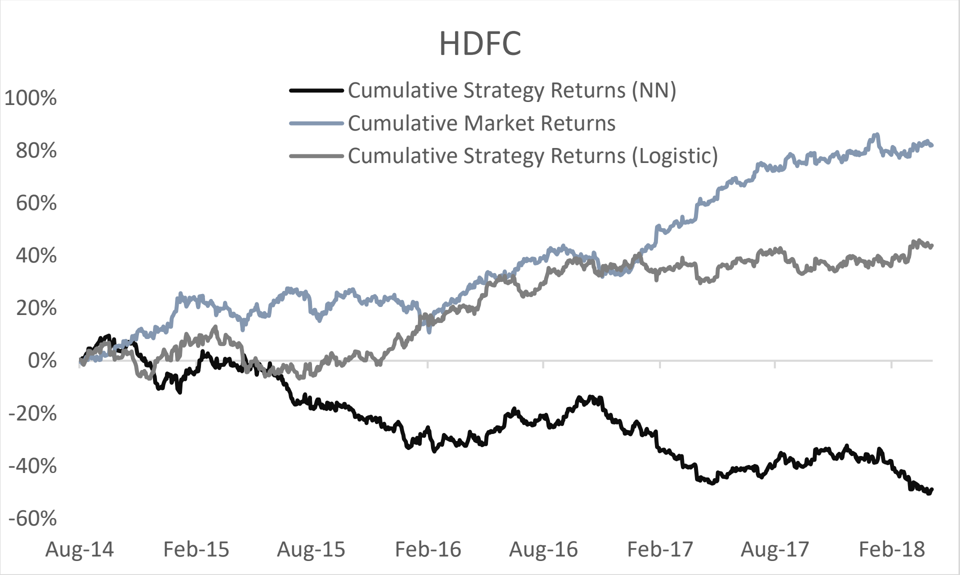

HDFC
Cumulative Strategy Returns (NN)
Cumulative Market Returns
Cumulative Strategy Returns (Logistic)
25%
20%
15%
10%
5%
0%
-5%
-10%
May-17
Aug-17
Nov-17
Feb-18

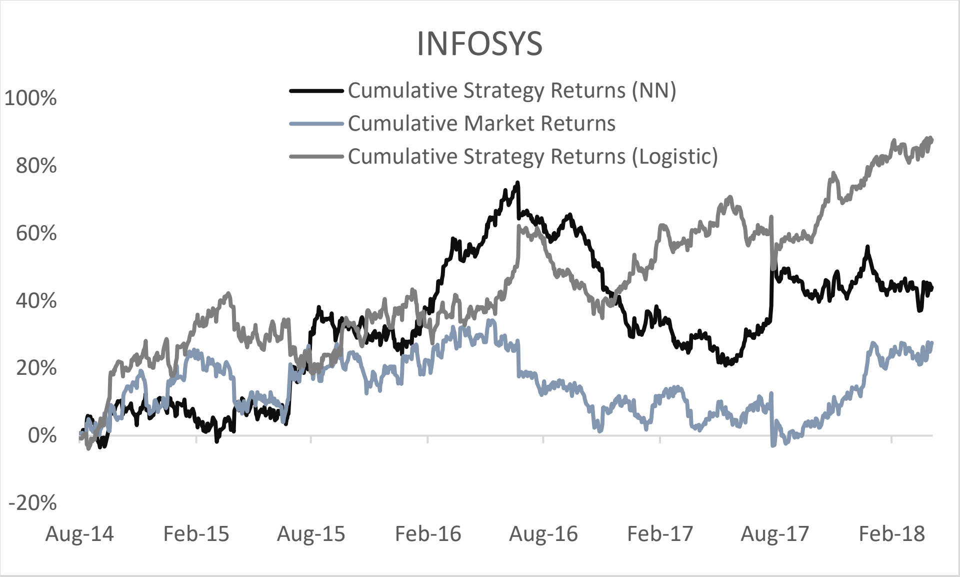

INFOSYS
Cumulative Strategy Returns (NN)
Cumulative Market Returns
Cumulative Strategy Returns (Logistic)
50%
40%
30%
20%
10%
0%
-10%
-20%
May-17
Aug-17
Nov-17
Feb-18

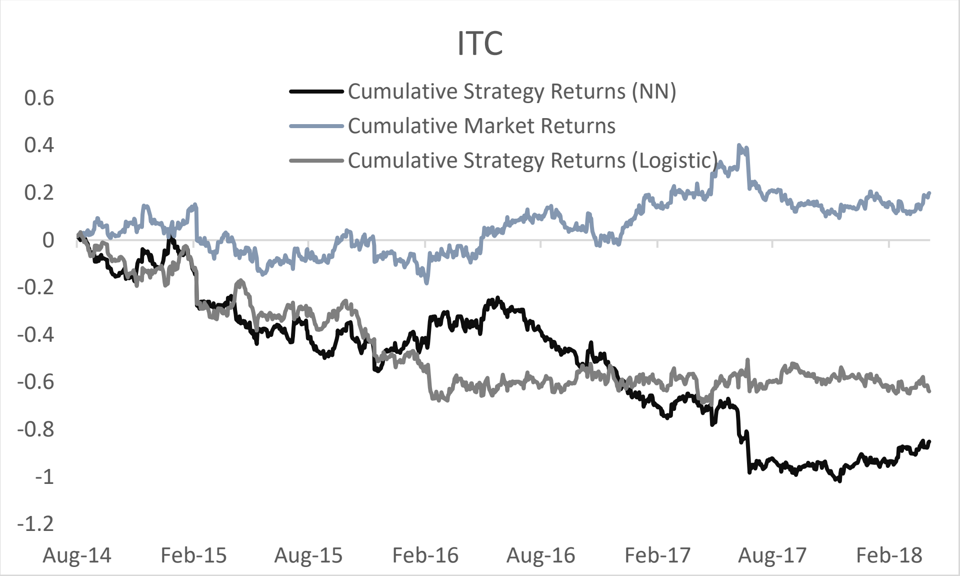


	Return vs Vol (Annualized)		
	HDFC	**INFY**	**ITC**
Return (NN)	9.8%	38.9%	-4.0%
Return (Logistic)	3.5%	11.4%	14.2%
Return (Market)	18.2%	21.8%	-12.2%
Volatility (NN)	13.7%	23.7%	24.0%
Volatility (Logistic)	13.8%	23.8%	24.0%
Volatility (Market)	13.7%	23.7%	24.0%

Test Size 20%







	Return vs Vol (Annualized)		
	HDFC	**INFY**	**ITC**
Return (NN)	-17.1%	10.7%	-41.2%
Return (Logistic)	10.7%	19.2%	-24.8%
Return (Market)	18.2%	7.0%	5.2%
Volatility (NN)	16.0%	24.4%	25.9%
Volatility (Logistic)	16.0%	24.3%	25.9%
Volatility (Market)	16.0%	24.4%	25.9%

The key findings from our study are:

- Each stock has its own idiosyncratic behavior and we need to calibrate our model individually for each stock
- Neural Networks perform very well for HDFC and ITC over short and medium term (1% and 5% test size)
- Neural Networks perform very well for INFOSYS over medium term (5% test size) and relatively well over long term (20% test size)
- Logistic Regression perform very well for INFOSYS over long term (20% test size)
- The timing (entry and exit) of strategies is very important

CONCLUSION

The prediction of stock prices is a very difficult problem to solve due to inherent data issues. The time series data needs to be stationary with no autocorrelation to be used in a machine learning or time series model.

Every stock has got idiosyncratic behavior and we need to model more fundamental factors apart from the technical ones. Sentiment based factors can also help to capture some of the stock behavior. The stock under goes many regime changes with mean and variance changing over time. We could introduce a new factor which captures the given regime of the stock and expectation maximization algorithms can be used to capture the regimes.

Finally the daily Long-Short strategy can be expensive and we have not deducted the cost of daily rebalancing from the strategy returns. An ideal strategy should move from daily signals to a systematic monthly or fortnightly strategy which is more based on fundamental factors than technical ones.