Comparing Different Machine Learning Techniques for Stock Market Predictions.

Tian Cater[a]

[a] *Data Science 871: Machine Learning Project*

1. Introduction

Predicting future outcomes based on historical data is known as a forecasting algorithm. Accordingly, this project implements Facebook's Prophet, the K-Nearest Neighbours (KNN), and the Feed-forward Neural Network (FNN) algorithms to predict the movement in value of the Johannesburg Stock Exchange's (JSE) Top 40 Index. Importantly, these predictions are not aimed at analysing stock price movements to provide investor insights. Instead, the purpose is to show the models fitted, compare the different forecasting approaches, and encourage their use.

Since the outset of financial markets, investor's have been attempting to predict market trends and random behaviour. However, as indicated by some of the most formidable market investors, predicting stock market returns is almost impossible. However, all improving techniques surrounding machine learning and its implications for forecasting time series might someday provide more robust stock market predictions. It is in investigating some of these machine learning techniques that give inclination to this project.

All the models performed very well inside the prediction intervals and the accuracy metrics, with the FNN algorithm displaying the most accurate predictions. The KNN model did not serve as well as the Prophet and FNN models under our metrics. This could be because KNN may need more tuning phases, training, and testing approaches, or they are not as effective as the other models because they mainly use classificatory terms more than forecasting.

Email address: `19025831@sun.ac.za` (Tian Cater)

Code and data:
The code and data used in this project is readily available at https://github.com/TianCater/Data_Science_871_Machine_Learning. I would also like to thank Katzke (2017) for this seamless integration of Latex in R.

2. Forecasting using the Prophet Algorithm

The Prophet model is an additive regression model developed by Taylor & Letham (2018) at Facebook's Core Data Science team, providing an effective solution to forecast time series with trending and seasonal properties. The model consists of four main components: A logistic growth curve or piecewise linear trend ($g(t)$), a yearly seasonal element using a Fourier series ($s(t)$), a weekly seasonal part using dummy variables, and the effects of holidays or significant events ($h(t)$).

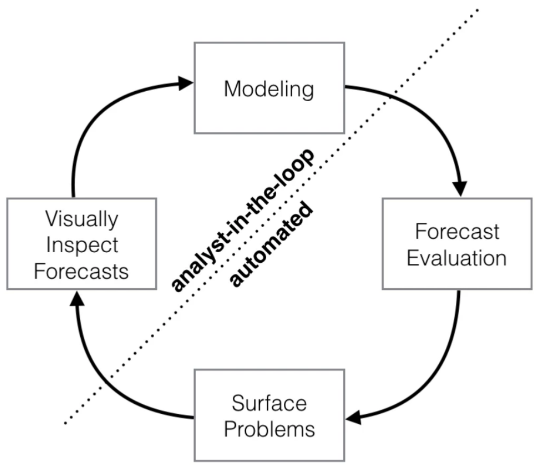

Figure 2.1: The dynamics of the Prophet algorithm.

Figure 2.2 compares the actual data versus the predicted values fitted using a piecewise linear trend and forecasting 500 observations in advance. Additionally, the number of lags are selected based on the Akeike Information Criteria (AIC). Visually, the predicted values seem to do the actual data rather well. However, to better understand the data generating process, the expected prophet components divided by a trend component, weekly seasonality and yearly seasonality are depicted in figure 2.3. Outside of the apparent weekly seasonality due to markets being closed on weekends, the predicted values show higher volatility during the year's first quarter. This evidence makes sense as most companies in the JSE Top 40 index have their financial year-end during this period.

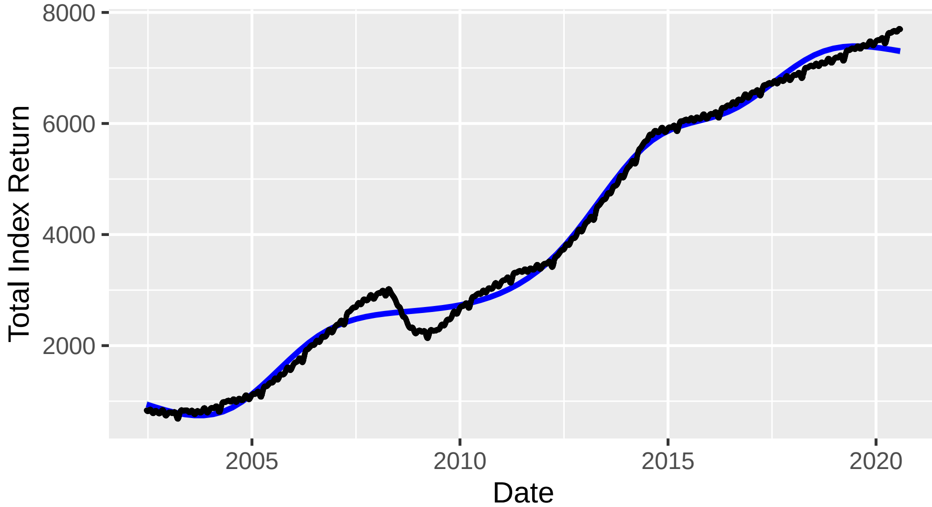

Figure 2.2: Visualisation of Train Prediction (blue) vs Observed Data (black).

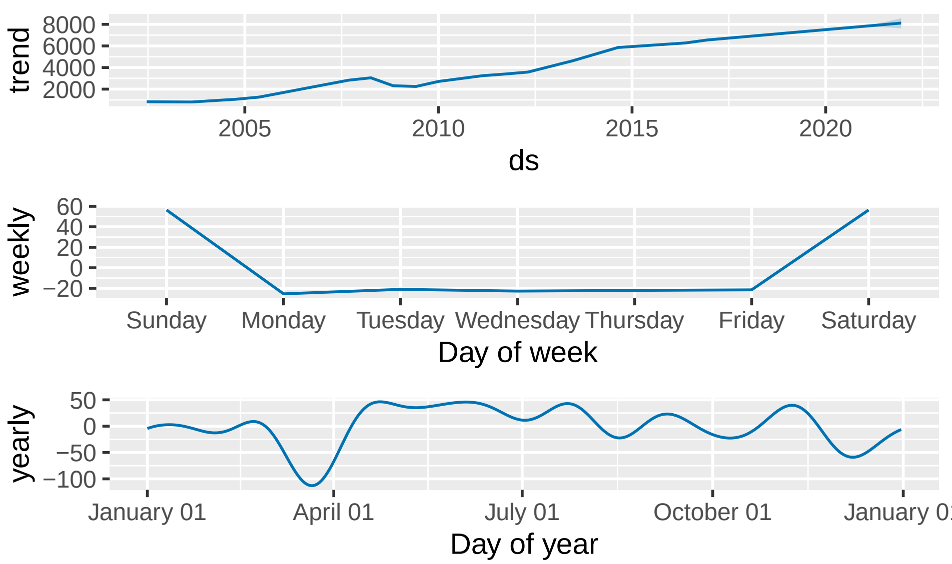

Figure 2.3: Cross validation of Prophet components controlling for trend, weekly seasonality, and yearly seasonality.

The corresponding statistics reflecting the accuracy of the prediction are provided in table 2.1. All of the goodness-of-fit statistics report that the Prophet predicted values are sufficiently accurate in predicting the actual observations. For ease of interpretation we consider the Mean Absolute Percentage Error (MAPE) of 4.631%, indicating that the actual and predicted value are off by 4.63%.[3]

[3]The formula for the MAPE is given by: $(1/n)\sum_{t=1}^{T}|\frac{actual\ value - predicted\ value}{actual\ value}| \times 100$.

Table 2.1: Goodness of fit statistics reflecting the accuracy of the Prophet model forecasts.

	ME	RMSE	MAE	MPE	MAPE
Test set	0.015	234.43	156.524	-0.365	4.361

3. The K-Nearest Neighbours (KNN) Algorithm

The K-Nearest Neighbours (KNN) algorithm is a supervised machine learning algorithm first introduced by Keller, Gray & Givens (1985). The algorithm applies a method of classifying data to estimate the likelihood that a data point will become a member of one group or another depending on the group to which the data points nearest to it belong (the training set). In this section, our primary objective is to forecast values for the JSE Top 40 index through a KNN experimental approach and compare its forecast accuracy with the other models adopted.

The following 500 values forecasted are shown in figure 3.1. The number of parameters in the KNN regression is set to 100 because this project focuses on comparing different projection models rather than deducing the actual movement of stock prices. Furthermore, the lags are selected based on the AIC, and recursive methods are applied as the multiple-step ahead strategy.[4]

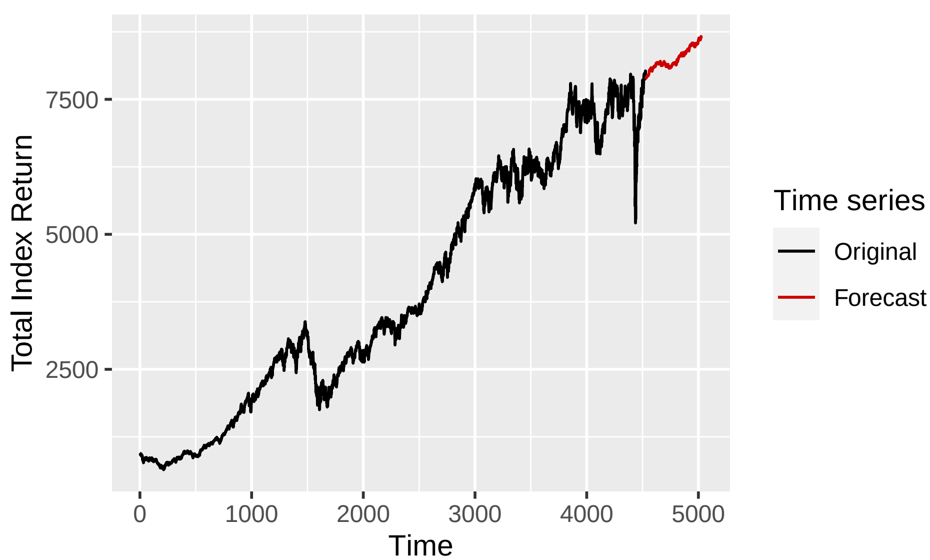

Figure 3.1: Forecasted JSE Top 40 index using the k-nearest neigbours algorithm.

The forecasts depicted in figure 3.1 indicate that the JSE Top 40 index will experience growth, with

[4]See Taunk, De, Verma & Swetapadma (2019) for a detailed description of the intricacies surrounding the KNN algorithm.

a marginally slight decline after two months that recovers rather quickly towards its stable growth path. The statistics that measure the accuracy of the prediction are given in table 3.1 below. Again, referring to the MAPE value of 6.95%, the projections fit the actual data sufficiently, with a 6.95% error between the predicted and actual values. However, this is a slightly worse fit than the predictions computed using the Prophet algorithm.

Table 3.1: Goodness of fit statistics reflecting the accuracy of the KNN algorithms forecasts.

	x
RMSE	662.689
MAE	481.607
MAPE	6.947

4. The Feed-forward Neural Network (FNN)

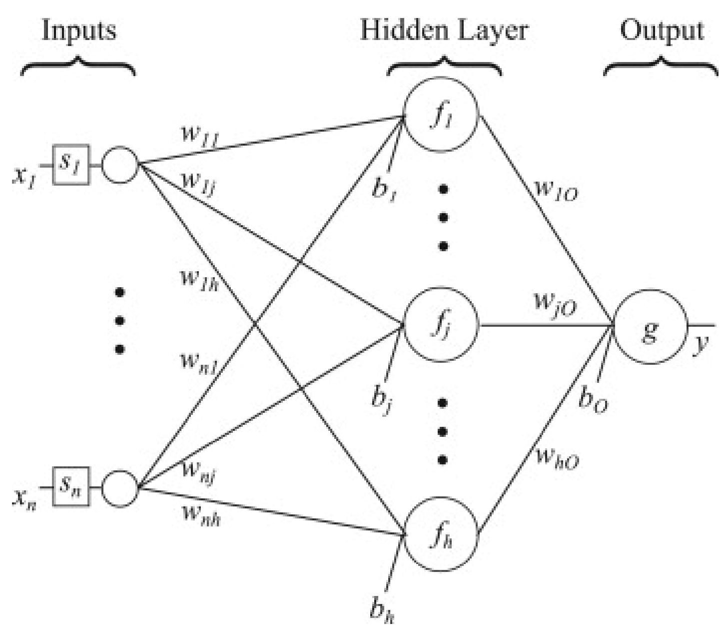

Figure 4.1: The dynamics of the feed-forward neural network algorithm.

A single hidden layer feed-forward neural network (FNN) is the first and most straightforward kind of artificial neural network where connections between nodes do not form a loop or cycle (Sanger, 1989). Consequently, the information only flows forward from the input nodes through the hidden nodes and to the output nodes. In its form considered here, only one layer of input nodes transmits weighted inputs to the next layer of receiving nodes.

This section fits a single hidden layer FNN to the JSE Top 40 time series. Similar to the approach by Wang, Er & Han (2014), the functional model involves using lagged values of the process as the input data, resulting in an estimated non-linear autoregressive model.

The specific number of hidden nodes is half the number of input nodes, including external independent variables, plus one. To ensure that the residuals will be approximately homoscedastic, the Box-Cox lambda approach is used. In figure 4.2, the following 500 values are forecasted with the neural net fitted and the number of lags selected based on the AIC. In contrast to the Prophet and KNN predictions, figure 4.2 predicts that the JSE Top 40 index will experience a sharp decline until levelling off approximately three months later.

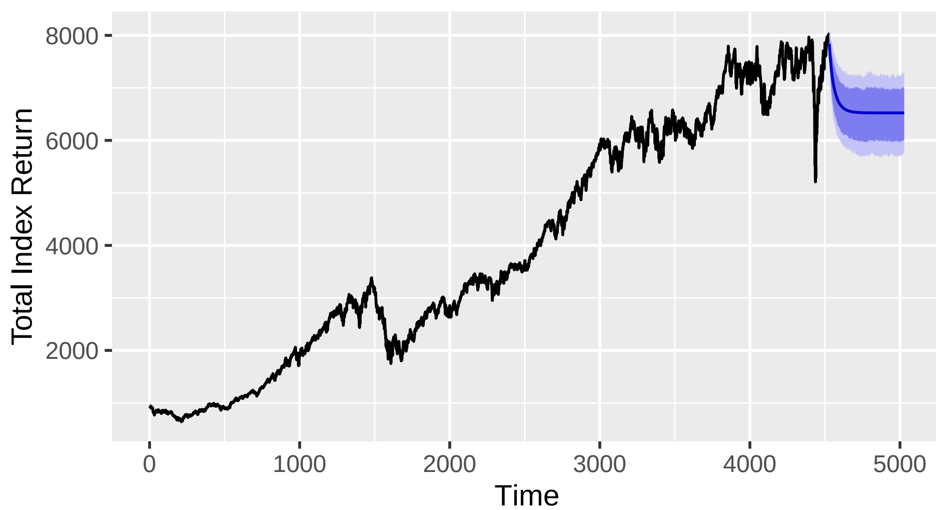

Figure 4.2: Forecasted JSE Top 40 index using the feed-forward neural network.

The accuracy statistics are given in table 4.1 below. The MAPE of 0.95 is an improvement on both the Prophet and KNN predictions, indicating that the actual and forecasted values are off by 0.95%.

Table 4.1: Goodness of fit statistics reflecting the accuracy of the FNN algorithms forecasts.

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	1.414	54.976	36.298	-0.006	0.991	1.042	0.039

5. Conclusion

This study focused on applying different models, learning how to use them to forecast JSE Top 40 index values, and showcasing contrasting results. The Prophet and KNN algorithms predicted a price increase over the next 500 days, whereas the FNN algorithm a price decrease.

All the models performed very well inside the prediction intervals and the accuracy metrics, with the FNN algorithm displaying the most accurate predictions. The KNN model did not serve as well as the Prophet and FNN models under our metrics. This could be because KNN may need more tuning phases, training, and testing approaches, or they are not as effective as the other models because they mainly use classificatory terms more than forecasting.

References

10 Katzke, N.F. 2017. *Texevier: Package to create elsevier templates for rmarkdown*. Stellenbosch, South Africa: Bureau for Economic Research.

Keller, J.M., Gray, M.R. & Givens, J.A. 1985. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*. (4):580–585.

Sanger, T.D. 1989. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*. 2(6):459–473.

Taunk, K., De, S., Verma, S. & Swetapadma, A. 2019. A brief review of nearest neighbor algorithm for learning and classification. In IEEE *2019 international conference on intelligent computing and control systems (ICCS)*. 1255–1260.

Taylor, S.J. & Letham, B. 2018. Forecasting at scale. *The American Statistician*. 72(1):37–45.

Wang, N., Er, M.J. & Han, M. 2014. Generalized single-hidden layer feedforward networks for regression problems. *IEEE transactions on neural networks and learning systems*. 26(6):1161–1176.