CONTENTS

Predicting Stock Prices Using Large Multiple Kernel Learning Model (MKL)

Akshay Kumar Valappil Thodi
School of Computing, Engineering and Intelligent Systems
University of Ulster
Derry~Londonderry

Abstract: *This paper aims to build a machine learning model which will predict the stock price and help make an investment decision at the end. Sentiment analysis on the web scraped data, time series analysis and technical analysis on the historic stock prices data are combined into a single model using multiple kernel learning model. This model, merged with the decision-making matrix gives the final result. Out of the 4 models built during the study, the combined model performed the best with an R^2 value of 0.9998 and that model integrated with the final decision-making matrix gives a more accurate objective method to the ultimate question of whether to invest in a stock at this point of time or not. The model doesn't just help capture the potential gains from investments but also help avoid potential losses by telling when to divest.*

INTRODUCTION

There are plenty of research papers on predicting stock prices as accurately as possible. As a result, there are multiple ways of gaining insights into understanding and predicting stock prices. One method is the technical analysis where trends and charts from historical prices are used to calculate technical indicators such as exponential moving averages, bias, moving average convergence/divergence etc and in turn gain actionable insights [1]. "Technical Analysis of The Financial Markets" by John J Murphy is one such paper. Another way of understanding stock price movements is fundamental analysis. This method measures the intrinsic value of the capital by studying related macro-economic and company financial factors [2]. 'A systematic review of fundamental and technical analysis of stock market predictions' [3] by authors Isaac Kofi Nti, Adebayo Felix Adekoya & Benjamin Asubam Weyori discuss both of these methods at a fundamental level as obviously mentioned in the title.

There have been multiple attempts in the past to introduce machine learning and AI in the area of stock price analysis. Franklin Allen and Risto Karjalainen used a genetic algorithm to find trading rules [4]. P.K.H. Phua and team used neural networks to forecast stock index increments in 2003 [5]. W. Huang and team used the support vector machine to serve the same purpose [6]. There has even been a study by Shangkun Deng and team where they used multiple kernel learning (MKL) to combine sentiment scores from natural language processing (NLP) and indicators from technical analysis to better predict stock prices [7]. Each of those studies was better than the previous one in terms of providing better insights into how to predict stock price or volume or make an investment decision.

This study intents to take this one more step ahead. Learning from Shangkun Deng et al. the aim is to combine multiple sources of information regarding stock prices and build a single model using multiple kernel learning. The sources considered are time series analysis, technical analysis on historic price data and sentiment analysis on news. Since these sources are diverse in its nature, combining them, even though seems impossible, is enlightening at best.

This study is different from its predecessor in two major aspects. One, the introduction of the time series along with technical analysis which is missing from the earlier study. And second is the method of application of multiple kernel learning. Instead of simply inserting the data features into a MKL algorithm, this study has used the definition of MKL to combine different features in a more practical and simpler way which, will be explained in the following sections.

The data used for this study comes from two major different sources. The first one is the historic price data which has been taken from "Yahoo finance" [8] and has variables date time, opening price, closing price, highest price, lowest price and volume. This data was used for both time series analysis as well as the technical analysis. The second data is news articles and reviews scraped from AutocarIndia.com [9] which was used for the unsupervised sentiment analysis.

METHODOLOGY

Figure 1 is a diagram explaining the steps involved in this analysis.

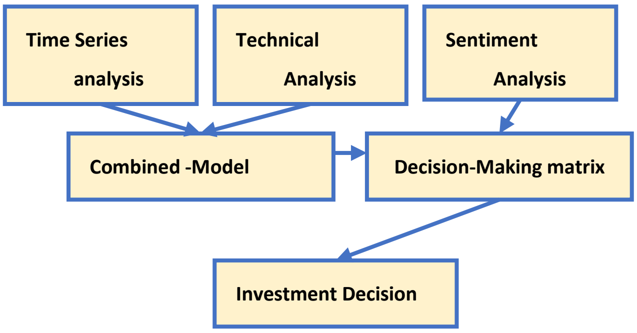


Figure 1: Steps involved in this analysis

In the first step, a time series analysis is conducted on the historic closing prices. In the second step the major technical indicators are calculated and added to the same data and a simple linear regression is fit to predict prices. At this point there will be two different price prediction models which are combined to form a single, better regression model. Parallelly, using R studio, AutoCarIndia.com website is scrapped for news articles and reviews which in turn is used for sentiment analysis. This result along with the earlier regression model is sent through a decision-making matrix which gives the investment decision. All these steps are explained in detailed further.

Time series analysis:

Time series analysis is a machine learning method used to analyse and understand trends in a sequential data collected over a time period where these data points are equidistant [10]. This means these set of techniques can be used to analyse for example monthly sales data of a company or daily weather data of earth or weekly patient data at a hospital etc. The most important or attractive aspect of this analysis is that it allows us to predict the future behaviour of the numbers under analysis from the past numbers, i.e. it can give next month's sales of the company from current sales or predict whether it will rain tomorrow based on the raining pattern for the past year or even tell us how rushed the hospital could be in the next week. Such information helps authorities make decisions like production planning or employee rota at a hospital or simply helps us make the decision to carry an umbrella or not.

There are four major components to a time series data. The first component is trend, which is the general nature of the data to either increase or decrease in the long term. It is also not necessary for the data to either increase or decrease. It could also have a zero trend i.e. the data stays within a limit no matter what. Another component of time series data is seasonality. This corresponds to events that happens periodically in a span of less than a year. Sales during Christmas could be one example for seasonality. Another component is cyclic variation which is yet another rhythmic variation but one that happens in a span of over a year. The last and final component is irregularities which as the name suggests are random and is difficult to account for in an analysis.

In this time series analysis, the data used is historic closing stock price data for Bajaj Auto Ltd. for the past five years. Figure 2 is a line graph of the closing prices over time.

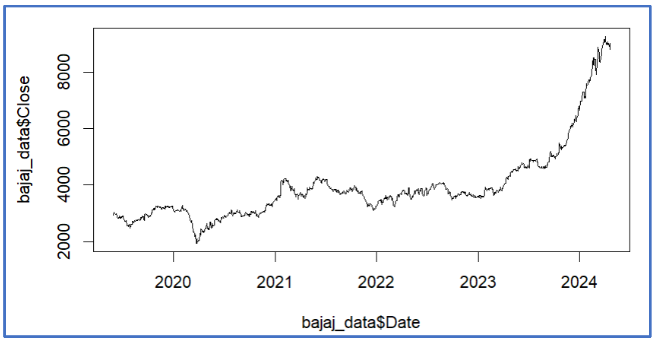


Figure 2: stock price vs time

It is clear from the graph that there is an upward trend in the stock price from 2019 to 2024. After initial exploratory analysis of the data, it was found that there were plenty of missing values even though sum(is.na(df)) was coming up to be zero. That is because the frequency of the data was daily, but since the stock exchanges don't work on weekends or holidays, they were missing from the data. But logically thinking, this can simply be ignored because that doesn't affect the natural flow of data and whatever the forecasted data point is, that can be assigned to the next business day and not the next calendar day. There were no other discrepancies associated with the data.

Then the data was split into train and test and a simple Auto regressive integrated moving averages (ARIMA) model was fit to it. ARIMA model is an algorithm used in time series analysis for future forecasting where there are two parts. One is the auto regressive (AR) component where the algorithm allows the forecasted value to be dependent on the previous values and the second component is the moving average (MA) component which models the error term as the combination of both current data point as well as previous data points. The data needs to be stationary to be suitable for an ARIMA model. That means the statistical properties like mean, SD etc needs to be constant and there shouldn't be seasonality. Also, in an ARIMA model, 'p', 'd' and 'q', which are the order of AR model, differences and order of MA model respectively. These are determined using auto correlation function (ACF) and partial auto correlation functions (PACF). All these steps needs to be manually done in the 'arima()' function in R studio while it is automatically done in the 'auto.arima()' function. So, using the latter, a simple time series model is fit to the data.

Now, it is in a forecasting models' nature to be less and less accurate the more it forecasts into the future i.e., the performance of any forecasting model will be much better when it predicts 30 numbers to the future compared to a 100 number into the future. Method used in this study to tackle this problem is to predict only one data point at a time and that fits perfectly with logic as well. There is no need to predict the stock price for example a hundred days into the future, there is only need to predict tomorrow's price. So, taking this fact into consideration and to improve the model's performance, a loop was created which would split the data into train of size (n) and test of size (N-n) where N is the total number of data points and n is any number between 1 and N (here is n =846 for a 70-30 split) , fit a model and predict one data point into the future, append that in a predicted list and then split the data again into a new train of size (n+1) and test of size (N-n-1) and repeat the steps again. Finally, this appended list of forecasted values is compared against the actual stock prices and are found to have a better accuracy compared to the earlier model. These predicted values are attached to the actual test data and saved as a CSV file for later use.

Technical analysis:

Technical indicators used in this study are simple moving averages (MA), exponential moving averages (EMA) and moving average convergence/divergence (MACD). These indicators tell us how the numbers are behaving over time and how they will behave provided the trend continues.

Moving average (MA): is a simple statistical calculation where the mean is calculated based on a pre-determined interval. In this study the intervals under consideration are 5 and 10. MA smooths out the data and takes out any random or short-term fluctuations from the data.

Exponential moving average (EMA): is another statistical indicator which assigns a higher weightage to the recent values compared to MA where it assigns equal weightage to all data points. The intervals taken into consideration here are 12 and 26 for the calculation of moving average convergence/divergence (MACD) which is explained later in this section.

$EMA_{(t)} = P_{(t)}*(2/(1+i)) + EMA_{(t-1)}*(1-(2/(1+i)))$

Where t = time/day, P = price and i = interval chosen (here it is 12 or 26)

Moving average convergence/divergence (MACD): is an indicator which tell the trader when to buy or sell the stocks. When the MACD line crosse above signal line stock is purchased and it is sold when the line crosses below the signal line. A signal line is an MACD line with nine period EMA [11]. It is calculated as follows,

MACD = EMA (12) – EMA (26)

As a first step, these technical indicators are calculated using excel. And then new features are created for each of these indicators by taking one day lag since the intent here is to find the price of tomorrow using the indicator today, i.e., the requirement is to find out the relationship between price and previous day indicators. After reading this new file into R studio and basic exploratory and pre-processes, a simple linear regression model is fit into the data. And as done earlier, a loop is created to find the price for next day using train data of size n and test data of size (N-n) and the steps are repeated for train data of size (n+1) and test data of (N-n-1). The performance of both models is compared and the resulting test data of size (N-n) attached with the forecasted values is wrote into a csv file and saved for further procedures.

Sentiment analysis:

Next stage of the analysis is scraping the relevant content off of the website. This is done using the r studio packages 'rvest' and 'httr'. First a user defined function is created to scrape the paragraphs from a given URL. A loop is created using the same packages to scrape off all the URLs from the search result page where the search term was ‘Bajaj’ and these URLs are saved into a list. And this list of URLs is fed into the user defined function which, as a result will go into each of those web pages and scrape off all the paragraphs based on a key term which in this case is ‘Bajaj’.

Second step in this stage is to analyse the gathered text content for user sentiment. The content includes news articles about the company products and events, reviews from the customers, advices from experts, features of new motorcycles etc. The idea here is that the impact company products create in the users will directly affect the image of the company in the market and which in turn affect the stock prices. So, it was decided to analyse the customer reviews rather than every page, which will have diminishing impact on the model and final decision. The analysis was done using the R package ‘sentimentr’. This package has the function ‘sentiment_by’ which, when input with raw text, does all the basic pre-processes and gives out the sentiment scores in a list. This list of scores is as done before written into a CSV file format for later compilation procedures.

Multiple kernel Learning (MKL):

MKL is a set of machine learning methods used to combine different kernels into a linear or non-linear regression models. Fundamentally, instead of putting all features into a single data frame and fitting a single model, in MKL, features of diverse nature are used to build separate models and these models and/or results are defined as a set of new features to fit a single, new, better model. Now, there are multiple ways of combining models. The simple way of doing it would be to manually allot weightages for each model as per the industry knowledge and simply fit a linear equation i.e. regression. Another way of doing it is to use machine learning algorithms to find out the weightage and then define the model using those weightages. In r studio, the package named ‘kernlab’ helps to define the kernel functions for the root models using a function ‘rbfdot’, calculate the weightages using kernel mean matching (KMM) and define the combined model using these weights.

In this study, the method followed is very simple. MKL is used to combine the time series model and the regression mode from technical analysis. The weightage of these two models is calculated by simply fitting a linear regression model where the dependent variable is the actual price and the independent variables are time series forecasted price and technical analysis predicted prices. The combined model seems to have a much better performance than the previous models.

Decision matrix:

Finally, the predicted prices are combined with the matrix as shown in the figure 3 to make the investment decision.

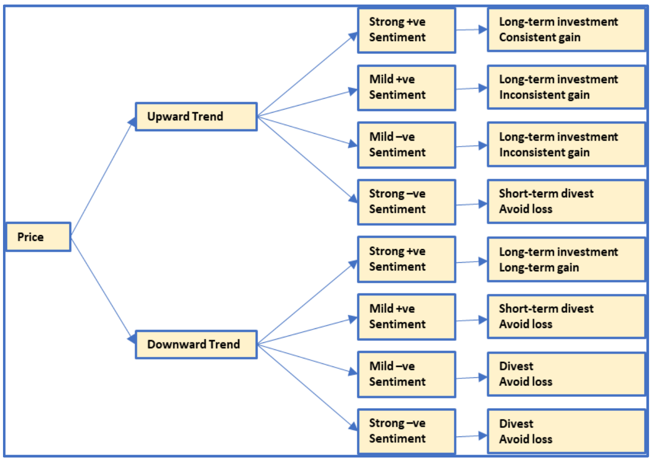


Figure 3: Decision matrix

The predicted price can have two different behaviours. It can be either an increasing trend or a decreasing trend. If it is on an increasing trend with a positive average sentiment, it shows good prospects for an investor. The decision would be a to invest long term and have long and consistent gain. On the other hand, if the sentiment is negative despite the increasing trend, it is obvious that the price will go down in the near future and the decision would be to divest for the short term to avoid major/minor losses. If the price is showing a decreasing trend but has a positive sentiment in the market, then there is a possibility of market improvement. So, the decision would be to invest long term for long term gains. There won't be any short-term gains in this investment, might even have short term loss, but in the long term, it will gain market. And finally, if the price has a decreasing trend and a negative market sentiment, then that is a sign of capital loss and the investor needs to sell everything as soon as possible because every day waited would mean loss.

RESULTS

Time series analysis:

Initial 'auto.arima()' model was one of the least performing models. The model had a RMSE value of '56.42' on the training set and '2437.6' on the test data. RMSE is the root mean squared error value which is calculated, as the name suggests, by square rooting the mean error/residuals. The second model which was built on loop predicting one day value at a time had a RMSE value of '25.42' on the test data. This shows that the second model is much better at forecasting the prices. The difference in performance can also be seen from the figures below.

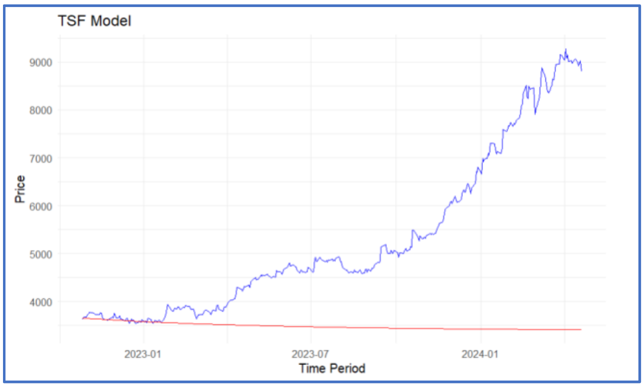


Figure 4: Initial TSF Model

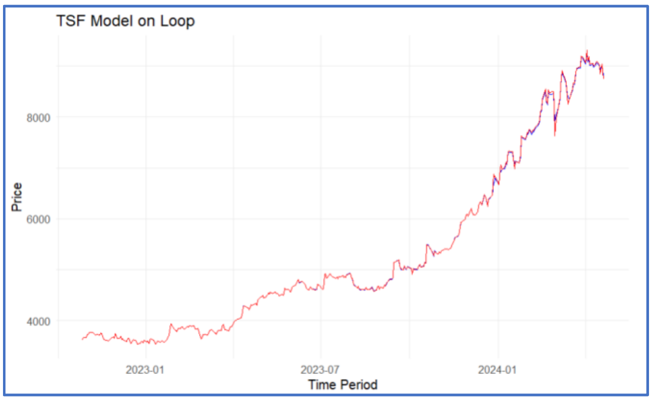


Figure 5: TSF Model on loop

The blue line is the actual prices and the red lines are the forecasted prices. In figure 5, the lines completely overlap each other which signifies the accuracy of the model compared to figure 4 where the actual prices and predicted prices are in two completely different paths.

Technical analysis:

The initial model, where lagged MA and MACD are the independent variables and closing price is the dependent variable, has an R^2 value of 0.975 on the training data and a RMSE value of 2859 on the test data. Given below is the Q-Q plot (figure 6) of the model which suggests that the data is more or less normally distributed with some extreme value suggested by the curve off in the extremities [12].

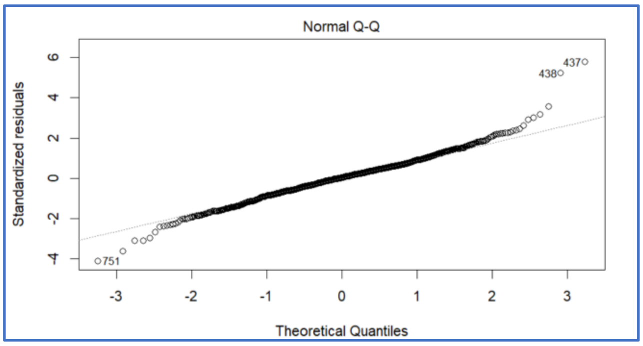


Figure 6: Q-Q plot

And the residual plot (figure 7) proves the homoscedasticity of the model since there is no systematic increase or decrease in the variance and is completely random in its pattern.

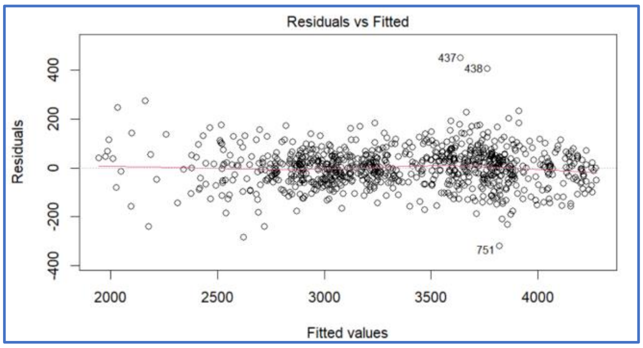


Figure 7: Residual plot

Figure 8 is the summary of the regression model.

```
Call:
lm(formula = bajaj_train$Close ~ bajaj_train$MA_5_.1 +
bajaj_train$MA_10_.1 +
    bajaj_train$MACD_.1, data = bajaj_train)

Residuals:
    Min      1Q  Median     3Q    Max
-322.32  -47.49    1.25   44.91  450.49

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         58.77127   19.49065   3.015  0.00264 **
bajaj_train$MA_5_.1   1.42848    0.06559  21.780  < 2e-16 ***
bajaj_train$MA_10_.1 -0.44566    0.06466  -6.893 1.07e-11 ***
bajaj_train$MACD_.1   0.08292    0.05144   1.612  0.10731
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 78.25 on 841 degrees of freedom
Multiple R-squared:  0.9757,Adjusted R-squared:  0.9756
F-statistic: 1.126e+04 on 3 and 841 DF,  p-value: < 2.2e-16
```

Figure 8: Summary of first regression model

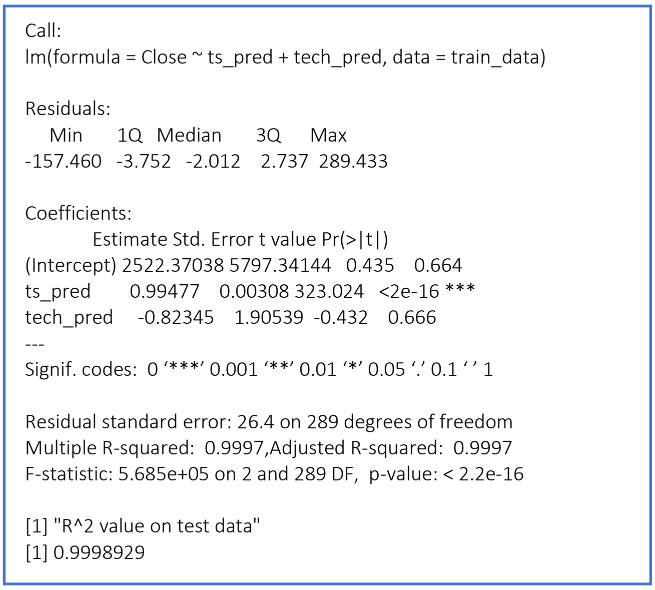
The next model on the loop gave a RMSE value of 2750 on the test data. So, both models aren't good enough predictors of the prices.

Sentiment Analysis:

The result of the sentiment analysis is a set of sentiment scores on the content scraped of the website. The average sentiment score of the content is 0.15 (positive). This is a mildly positive score which means this will be used in the decision-making matrix.

Multiple Kernel Learning (MKL):

The first step in this stage was to read all the data sets that were the results of earlier stages. Then those data namely, test data, the result of technical analysis and the result of time series analysis were combined into a single data frame which in turn is used for MKL. The MKL method used here is simple linear regression to find the weightage of each earlier models. The resulting model which is a simple linear regression model gave an R^2 value of 0.9997 on the train data and 0.9998 on the test data. Figure 9 is the summary of the combined model.



```
Call:
lm(formula = Close ~ ts_pred + tech_pred, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-157.460  -3.752  -2.012   2.737  289.433 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2522.37038 5797.34144   0.435    0.664    
ts_pred        0.99477    0.00308 323.024   <2e-16 ***
tech_pred     -0.82345    1.90539  -0.432    0.666    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.4 on 289 degrees of freedom
Multiple R-squared:  0.9997,Adjusted R-squared:  0.9997 
F-statistic: 5.685e+05 on 2 and 289 DF,  p-value: < 2.2e-16

[1] "R^2 value on test data"
[1] 0.9998929
```

Figure 9: Summary of the combined model

The MKL model has a performance much better than that of the individual results form time series or technical analysis results.

Table 1 is the summary of results from multiple studies in the past that has been referred for this study, along with the results of this study.

Table 1: Summary of results

Journal	Deng et al [7]	P. K. H. Phua et al [5]	Huang et al [6]	This study
Model	MKL (Tech + Senti)	recurrent neural network	SVM	MKL (Tech + TSF)
Performance	0.8865	0.73	0.75	0.9998
Measure	RMSE	Accuracy	Hit Ratio	R squared value
Data	Sony stock price data	NASDAQ	NIKKEI 225	Bajaj Auto Ltd. Stock data

Comparing these models objectively based on the table above is not completely sensible/fair since they use different measure of performance on different data. But it can be easily seen that as far as benchmarks go in any measure of performance, an R squared value of 0.9998 is an excellent result.

DISCUSSION

Time series Analysis:

As mentioned earlier, the initial model performed much worse than the later one where the loop was integrated to find one data point at a time. This is because a time series is based on historical data. So, the model would be more accurate on a shorter time horizon compared to a longer time horizon. There are certain studies which have explored the possibilities of improving the performance of a time series forecasting model (TSF), especially a long-term time series forecasting model. One such study is by Junhong Chen et al. where they use an optimized transformer model to improve the accuracy and efficiency in time series forecasting [13]. Their model involved a variable-isolating mechanism, a cut-up mechanism and an improved transformer model. Jianhua Hao and Fangai Liu uses a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network [14] to resolve the performance issues with time series forecasting.

The reason any of these complex models and resolutions were not used in this study was the fact that there is no need for long term time series forecasting here in this study. The aim is to make the decision of whether to invest in this stock today or not based on tomorrow’s predicted price. The only forecast needed for this is tomorrow’s price. That was the fundamental logic to form a much simpler loop model than to follow any of those above-mentioned complex algorithms. And as logic expected, the second model performed much better compared to the initial model.

The only way to improve the TSF model in this study would be to use the same looping mechanism on other time series forecasting algorithms like SARIMA or FB prophet. Even then, the forecasting would be much better performing when used on a loop rather than using it simply to predict a long terms time horizon.

Technical Analysis:

There are plenty of technical indicators that gives insights on stock price fluctuations. On-balance volume which measures the flow of volume over time, accumulation/distribution line which measures the flow of money in and out of security, average directional index which calculates the strength of an index, aroon indicator which identifies the trend changes, relative strength index which measures the strength of price movements and stochastic oscillators which recognises the over-bought/sold conditions are few of them. Using many/all of them will definitely have a diminishing returns effect on the model. So, the idea behind choosing moving average (MA) and moving average convergence/divergence (MACD) was to find the simplest features with maximum influence/ maximum information. Moving average tells the trends and smooths out the data from seasonality and irregularities and MACD tells the direction of stock movement/decision making. These two covers the two major aspects of stock price stochasticity, one is numerical and the other is directional. And that is more than sufficient to build a model. Adding more features into the data set will increase the complexity of the model than the performance of it.

Another way this study could have conducted was to actually consider all those technical indicators and build a regression model and then improve the model by a feature selection algorithm. Any feature selection algorithm will give a set of most efficient set of features that gives out the best results.

Sentiment analysis:

The 'sentimentr' package used in this stage for calculating the scores is designed to calculate the English language text polarity at both sentence level as well as token level. Unlike other r packages, it is not just a dictionary-based algorithm. The package has a predefined lexicon along with known polarity or valence. In the first step it does a dictionary approach where it finds the valence of dictionary words as any other algorithm. In the second step, the algorithm identifies the valence shifters. They are the words or phrases which shifts the polarity of other words in that sentence. A valence shifter could be a negator, an amplifier, a de-amplifier or an adversative conjunction [15].

Multiple Kernel Learning (MKL):

The advantage of using MKL instead of a simple single model is that it allows multiple models for different natures of data. For example, in this analysis, if it was to form a single model from a single data frame, it would coerce a single model into the dataset i.e. either a regression or a time series on a combined data set of prices and texts. Such a model would only imbibe behaviour within the data that are regressive. The time series nature or the sentiment would never be taken into consideration in the model algorithm. But, when MKL algorithms are applied, it allows to use different models/kernels for different sets of data with different natures and then later on combine them in a different model that extracts all the information from the data and uses it to give the result.

MKL was used here to integrate time series model and the regression model but not the sentiment analysis model. There are certain reasons for that. First reason is that sentiment scores are unsupervised. That means there is no training and test data to measure the performance of the NLP model. In a general scenario there will be content against the sentiment/score for the researchers to train a model and then test it against the data and then in that case it could be possible to merge it with another supervised predictive model. Another reason to not use MKL to merge sentiment scores with other models is that the sentiment data is cross sectional and not a time series. Combining a cross sectional data with a time series data such as historic stock price would defeat the purpose of forecasting a time dependent variable. Another major reason the typical MKL methods of merging features will not work here is the difference in data structure. In the time series/technical data, each row is a data point while in the sentiment scores, each row is a text element which doesn't have any significance on it own.

Since, the reasons not to use MKL to merge sentiment scores with regression model are mentioned, there is a natural responsibility to also mention how it could have been tackled with more time invested. First step would be to collect website data corresponding to each time point in the time series data. Conduct sentiment analysis and calculate sentiment scores. Calculate average sentiment score and add that as a new feature in the data frame. Once there are data points for all the time points in the time series data, this can be used in the merged model as another feature.

Decision matrix:

The idea behind this matrix was to merge the two major aspects of an investment decision namely, quantitative and qualitative. The quantitative aspect was well delivered by the combined MKL model and the qualitative aspect is taken from sentiment analysis report of the product/company reviews. Now, the question would be if the matrix works well or not.

For example, on 18th November 2022, the predicted price is '3636.398' and the price has been on a decreasing trend to that day from the past week. The decision matrix to make a short-term divestment decision since the sentiment is only mildly positive which makes complete sense because the price goes down to 3572.75 in the next two weeks which would have cause a short-term loss. This would be avoided based on the decision matrix. The moment MKL model predicts a better price with improved trend, the mild positive sentiment would suggest re-investment and the whole activity will result in double profit. Profit from actual market improvement and the profit by avoiding the possible loss.

CONCLUSION & FUTURE WORKS

The aim of this study was to build a machine learning model to predict stock prices better than the existing models. The intent was to use multiple Kernel learning method to combine different root models. With time and research there has been some changes to the initial plans but the whole effort has ended up fruitful. Instead of blindly applying some machine learning technique on a random data and expecting good results, this study has changed the perspective on how to adapt during a machine learning project.

Predicting stock prices is a numbers game, a statistical and machine learning problem. But, making a decision on whether to invest in a stock at a point in time involves both numbers and human emotions. That is where the idea of combining human emotions and numbers came up. Human emotions captured through natural language processing of website reviews by sentiment analysis method and integrating that with the machine learning results using a decision matrix turned out to be much better than what the study had intended in the beginning.

The combined MKL model has an excellent R squared value of 0.9998. That along with the decision matrix increase the profitability opportunistically double since it not only captures all the profitable investment periods in timeline but it also accurately identifies the loss areas and helps avoid such investments. Hence it reduces the opportunity cost. There are certain areas in this study that can still be improved.

First one is time series forecasting model. Different time series forecasting algorithms like SARIMA, FB Prophet, exponential smoothing etc. can be tried in the same format of looping and tested for accuracy. Second one is feature-selection in the technical analysis. In this study the technical indicators have been selected manually based on logic and industry knowledge. This can also be done by one of the variable selection machine learning algorithms. That way there will be more objective reasoning behind technical indicators selected.

Another possible future work is on the sentiment analysis. As mentioned earlier fi there is an availability of sentiment scores on a time series, it would have bigger impact on the model. That way it would be possible to consider the average sentiment score as another feature in the MKL and simply apply it in the combined model. In this scenario, there wouldn’t be a decision-matrix. A possible short-coming of such a model is that it will only give the predicted price in an objective manner. The model will not give a subjective reasoning for investment decision making. It is also possible to use other lexicon-based sentiment analysis packages. ‘SentiWordNet’ [16] is an option as used by Deng et al. [7]. There are also advanced resources like Google Cloud Natural Language API by Google or Comprehend by Amazon. This can only improve the accuracy of the sentiment scores in the model and there is a very small chance that it will affect the final decisions made by the matrix.

REFERENCES

[1] A. HAYES, “Technical Analysis: What It Is and How to Use It in Investing,” Investopedia, 21 February 2024. [Online]. Available: https://www.investopedia.com/terms/t/technicalanalysis.asp. [Accessed March 2024].

[2] T. SEGAL, “Fundamental Analysis: Principles, Types, and How to Use It,” Investopedia, 19 december 2023. [Online]. Available: https://www.investopedia.com/terms/f/fundamentalanalysis.asp#:~:text=Fundamental%20analysis%20%28FA%29%20measures%20a%20security%27s%20intrinsic%20value,financial%20situation%20and%20current%20market%20and%20economic%20conditions.. [Accessed March 2024].

[3] A. F. A. &. B. A. W. Isaac Kofi Nti, “A systematic review of fundamental and technical analysis of stock market predictions,” Springer, vol. 53, p. 3007–3057, 2019.

[4] R. K. b. Franklin Allen a, “Using genetic algorithms to find technical trading rules,” Journal of Financial Economics, vol. 51, p. 245—271, 1995.

[5] P. Phua, X. Zhu and C. H. Koh, “Forecasting stock index increments using neural networks with trust region methods,” IEEE, vol. 1, pp. 260-265, 2003.

[6] Y. N. a. S.-Y. W. Wei Huang a b, “Forecasting stock market movement direction with support vector machine,” Computers & Operations Research, vol. 32, p. 2513 – 2522, 2005.

[7] S. Deng, T. Mitsubuchi, K. Shioda, T. Shimada and A. Sakurai, “Combining Technical Analysis with Sentiment Analysis for Stock Price Prediction,” IEEE, pp. 800-807, 2011.

[8] Yahoo Finance, “Bajaj Auto Limited (BAJAJ-AUTO.NS),” Yahoo Finance, [Online]. Available: https://finance.yahoo.com/quote/BAJAJ-AUTO.NS/history.

[9] Autocar India, “Search result for "bajaj" in "all" category,” Autocar India, [Online]. Available: https://www.autocarindia.com/search/all/bajaj.

[10] J. L. F. Wayne F Velicer, “Time Series Analysis for Psychological Research,” Handbook of Psychology, vol. 2, pp. 581-606, 2003.

[11] B. DOLAN, “What Is MACD?,” Investopedia, 8 March 2024. [Online]. Available: https://www.investopedia.com/terms/m/macd.asp. [Accessed march 2024].

[12] C. Ford, “Understanding QQ Plots,” University of Virginia, 15 August 2015. [Online]. Available: https://www.library.virginia.edu/data/articles/understanding-q-q-plots#:~:text=The%20QQ%20plot%2C%20or%20quantile-quantile%20plot%2C%20is%20a,a%20normal%20QQ%20plot%20to%20check%20that%20assumption.. [Accessed April 2024].

[13] H. D. S. W. a. C. L. Junhong Chen, “Improving Accuracy and Efficiency in Time Series,” Engineering Letters, vol. 32, no. 1, pp. 1-11, 2024.

[14] J. H. &. F. Liu, “Improving long-term multivariate time series forecasting with a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network,” Scientific Reports, vol. 14, 2024.

[15] R Documentation, “Sentimentr,” R documentation, 12 October 2021. [Online]. Available: https://rdocumentation.org/packages/sentimentr/versions/2.9.0. [Accessed march 2024].

[16] GitHub, “SentiWordNet,” GitHub, 1 June 2022. [Online]. Available: https://github.com/aesuli/SentiWordNet. [Accessed february 2024].