EE769 : Introduction to Machine Learning Course Project

Stock Prediction using Machine Learning

Rishabh Jain | 20i190008

Supervisor: Prof. A. Sethi

Electrical Engineering
Indian Institute of Technology, Bombay

Spring, 2020-2021

Stock Prediction using SVR and LSTM

ABSTRACT

The stock market's direction is always stochastic and volatile, and the prices of securities are deemed unpredictable. This area has long attracted researchers from different fields. There could be many factors that can affect the stock market, and hence the amount of data one can collect to explain those movements is also very vast. Analysts are now trying to use techniques from Natural Language Processing in the field of finance due to the similarity of sequential data. However, in this project, we will build a model based on only quantitative data, i.e. the stock's fundamental and technical data. Given time series data, we will use one of the most celebrated state-of-art deep learning sequential models, namely Long Short Term Memory (LSTM) and the traditional Support Vector Regression (SVR) model, for predicting stock prices. Our input data not only includes end day prices and trading volumes but also includes carefully selected corporate accounting statistics and technical indicators. The result has shown that SVR with a linear kernel is performing much better than the LSTM model. Even with a lot complex model structure LSTM is not able to attain better predicting results.

1 INTRODUCTION

The financial market, just like any other market, is a place to buy and sell. The only difference is that the traded things are currencies, stocks, derivatives, and other financial assets. Unlike the earlier times, now these transactions are executed virtually and traded either through exchanges or over-the-counter (OTC) markets.

Although humans can take orders and submit them to markets, automated trading systems operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, that does not mean that it does not require human interference. A lot of different factors are considered before executing these strategies. For instance, complex mathematical strategies are formulated, trading environments are checked, risk measures are set, analysis of transaction costs involved and evaluation of the models is performed. These strategies and models use the prediction from the machine learning algorithms to form theories about market movements and test them.

The stock markets are affected by many different dynamic factors acting together that are correlated. Mere the dimensionality of the factors that affect stock markets makes predicting stock movements a very challenging task. Moreover, with the developments in quant trading and algorithm trading in the stock markets, it has become complicated to capture the volatility of stock markets even with complex and dynamic deep learning models. Figure Figure 1 shows the volatility of the S&P 500 index from the year 1995 to the year 2021.

Now the factors that affect the markets can be classified broadly into three types of categories, Fundamental factors, technical factors, and market sentiments. For this project, we will only focus on the fundamental and technical factors.

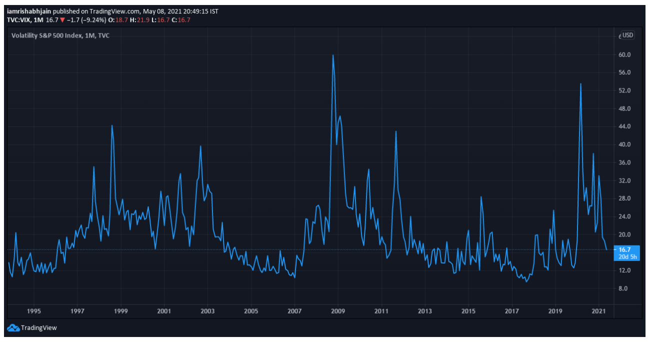

Figure 1: An illustration of volatility of S&P 500 Index from year 1995 to year 2021. Picture Credits : Trading View

Fundamental Factors: It takes into account a subjective view of the industry or company's future direction. It comprises of publicly available information such as market news, corporate statistics as well a series of financial statement releases. Fundamental factors usually take a long term view on the company's stock. They aim to gauge the company's strength for long-term investments and includes many other factors specific to its stock under consideration. Investors look at the intrinsic value of the stocks and performance of company, industry, economy etc. For example, for a cotton manufacturing unit, fundamental factors will also include the factors relating to the climate effects on cotton production, government or other regulatory changes.

Technical Factors: These factors are based on the time-series data. Many researchers have shown that historical prices are capable of predicting future stock movements. All the data related to the price of the stocks can be used to find patterns and make theories about future stock movements.

Market Sentiments: Sentimental data capture the essence of public sentiments. Data from the news feed, Twitter, and other social media platforms are extensively used to predict market fluctuations. This also included the trading psychology of investors in the market, behavioural patterns from certain activities in the market related to the company's stock. Any comment, positive or negative, can have a significant impact on market volatility. For example, news on election results, natural disaster, and pandemic can significantly affect the market.

This project aims to use fundamental and technical data to train a machine learning model that predicts the future stock price the next day. We will use the daily stock prices going back to 2010 for these four stocks- Apple(AAPL), Microsoft(MSFT), Amazon(AMZN) and Chipotle Mexican Grill(CMG). This paper discussed in details all the calculations that are performed to get the prediction. We will fit two models, SVR and LSTM, for each of these stocks and use Mean squared Error(MSE) and Mean Absolute Error(MAE) as our measure to compare the results.

2 BACKGROUND AND RELATED WORK

In the pre deep learning era, several studies have attempted financial time-series predictions. Most of these studies tended to employ statistical methods such as Weighted Moving Average (WMA), the generalised autoregressive conditional heteroskedasticity (GARCH) model, and the autoregressive integrated moving average (ARIMA). However, these approaches made many statistical assumptions, such as normality, stationarity and linearity, where the financial time series is non-linear. These approaches are not suitable for predicting stock prices.

In recent year, deep learning has demonstrated better performance thanks to the improved computational power and the ability of these algorithms to learn the non-linear relationships in financial data. The machine learning algorithms like Support vector machines (SVMs) and the deep learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) architecture like LSTM have shown awe-inspiring results and are capable of forecasting stock trends and movements.

Stock prices are dramatically affected by many factors. A discussed above, there are different type of factors which affect the market differently. Each researcher chooses those factors differently for their prediction model according to their own beliefs and hypothesis about markets. Many studies have been done using fundamental data and technical data to predict stock prices and some of them have shown promising results. For example, some authors have used the raw stock price data to calculate the technical indicators such as RSI, moving average, weighted moving averages and use them as input features. Though the results have shown satisfactory prediction accuracy but choosing those indicators and accessing the effect is in itself a very challenging task.

Fischer and Krauss [1] deployed an LSTM network in predicting the directional movement of constituent stocks of the SP 500 from 1992 to 2015. In their research, they used 240 one-day returns as input variables for the prediction model. They divided the entire time period into 23 study periods, where each study period had 750 days (approximately three years). Their model was compared with other classifier models, such as RF, DNN, and logistic regression. They provided an in-depth guide on data preprocessing, as well as the development of LSTM for financial time series prediction tasks, and concluded that their approach outperformed other baselines over 23 study periods in terms of average prediction accuracy, shape ratio, and profit. However, in their study, they only used one-day returns of the target stock to form input models.

3 METHODOLOGY

This section covers the complete architecture and methodologies used in this project. We will start with the two methods discussed above, SVR and LSTM. We will also discuss the different kernels that are used in our model. Although we tried fitting both radial basis function (RBF) kernel and linear kernel, the linear kernel showed better results and used to measure the fit later on. Then we discuss the literature on the LSTM and how it works. We will cover all the data sources and types of data used to train the model, explain all the features used, all the methods of preprocessing the data, and then finally, compare different measures to access the model's prediction accuracy.

3.1 Support Vector Regression

Support vector machine (SVM) is a popular machine learning took for classification and regression, first identified by Vladimir Vapnik and his colleagues in 1992. SVM can be categorised into two types based on the type of problems they solve, support vector classification (SVC) and support vector regression (SVR). SVR was proposed by Vapnik, Steven Golowich, and Alex Smola in 1997. A very important feature of SVMs are that they do not depend on the complete training data as the cost function associated with these model does not care about the points that lies beyond the margin (or the $\epsilon - threshold$ in case of SVR).

In any regression problem we are given n data points $(x_i, y_i); i = 1, 2, 3, ..., n$ where x_i is the *p-dimensional* vector space of different features and y_i is the target or response variable that belongs to the real number. The motive of the regression is to find a function $f(x)$, which, given the input x, predicts the value of y as close as possible. Similarly, the goal in SVR is to find a function f(x) that deviates from y by a value no greater than ϵ for each training point x, and at the same time is as flat as possible.To find the linear function

$$f(x) = \langle x, w \rangle + b \quad w \in \mathbb{R}, b \in \mathbb{R} \tag{1}$$

where $\langle . \rangle$ denotes the inner product in $\mathbb{R}$. For flatness its required to minimize the euclidean norm i.e. $\frac{1}{2}||w||^2$. This can be written as an convex optimization problem as follows

$$minimize \frac{1}{2}\|w\|^2$$

$$subject\ to \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases}$$

It is possible that no such function $f(x)$ exists to satisfy these constraints for all points. To deal with otherwise infeasible constraints, introduce slack variables ξ_i and ξ_i^* for each point. This approach is similar to the "soft margin" concept in SVM classification, because the slack variables allow regression errors to exist up to the value of ξ_i and ξ_i^*, yet still satisfy the required conditions.

$$Minimize \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to} \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{2}$$

The constant C is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin (ϵ) and helps to prevent overfitting (regularization). This value determines the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than ϵ are tolerated.The linear ϵ-insensitive loss function ignores errors that are within ϵ distance of the observed value by treating them as equal to zero. The loss is measured based on the distance between observed value y and the ϵ boundary. This is formally described by

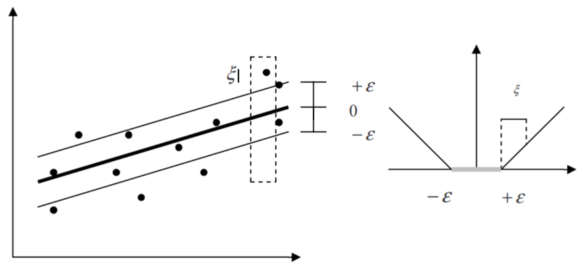

Figure 2: The soft margin loss setting corresponds to a linear SVR

$$L_\epsilon = \begin{cases} 0 & if\ |y - f(x)| \le \epsilon \\ |y - f(x)| - \epsilon & otherwise \end{cases} \tag{3}$$

Figure 2 depicts the situation graphically

Linear SVR : Dual Formula

The optimization problem previously described is computationally simpler to solve in its Lagrange dual formulation. The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem. The optimal values of the primal and dual problems need not be equal, and the difference is called the "duality gap." But when the problem is convex and satisfies a constraint qualification condition, the value of the optimal solution to the primal problem is given by the solution of the dual problem. To obtain the dual formula, construct a Lagrangian function from the primal function by introducing nonnegative multipliers α_n and α_n^* for each observation x_n. This leads to the dual formula, where we minimize

$$L(\alpha) = \frac{1}{2}\sum_{i,j=1}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle + \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha)$$

Subject to $\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0\ and\ \alpha_i, \alpha_i^* \in [0, C]$
(4)

The w parameter can be completely described as a linear combination of the training observations using the equation. Even for evaluating $f(x)$, it is not needed to compute w explicitly (although this may be computationally more efficient in the linear setting).

$$w = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)x_i \tag{5}$$

And therefore,

$$\boxed{f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b} \tag{6}$$

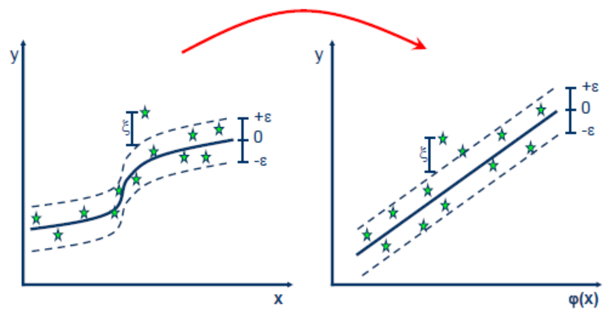

Figure 3: The kernel functions transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

Computation of b is done by exploiting Karush-KuhnTucker (KKT) conditions which states that at the optimal solution the product between dual variables and constraints has to vanish. In the SV case, this means

$$\alpha_i(\epsilon + \xi_i - y_i + \langle w, x_i\rangle + b) = 0$$
$$\alpha_i^*(\epsilon + \xi_i + y_i - \langle w, x_i\rangle - b) = 0$$
$$(C - \alpha_i)\xi_i = 0$$
$$(C - \alpha_i^*)\xi_i^* = 0$$

Non-Linear SVR

Some regression problems cannot adequately be described using a linear model. In such a case, the Lagrange dual formulation allows the previously-described technique to be extended to nonlinear functions.Obtain a nonlinear SVM regression model by replacing the dot product of x_1 and x_2 with a nonlinear kernel function $G(x_1, x_2) = \langle\phi(x_1), \phi(x_2)\rangle$, where $\phi(x)$ is a transformation that maps x to a high-dimensional space.Figure 3

We will mostly be interested in the Gaussian kernel or the radial basis function kernel which is given by

$$G(x_j, x_k) = exp(-\|x_j - x_k|^2)$$

The Gram matrix is an n-by-n matrix that contains elements $g_{i,j} = G(x_i, x_j)$. Each element $g_{i,j}$ is equal to the inner product of the predictors as transformed by ϕ. However, we do not need to know ϕ, because we can use the kernel function to generate Gram matrix directly. Using this method, nonlinear SVM finds the optimal function $f(x)$ in the transformed predictor space.

The dual formula for nonlinear SVM regression replaces the inner product of the predictors $(x_i x_j)$ with the corresponding element of the Gram matrix $(g_{i,j})$. Nonlinear SVM regression finds the coefficients that minimize

$$L(\alpha) = \frac{1}{2}\sum_{i,j=1}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x_j) + \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha)$$

Subject to $\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0\ and\ \alpha_i, \alpha_i^* \in [0, C]$
(7)

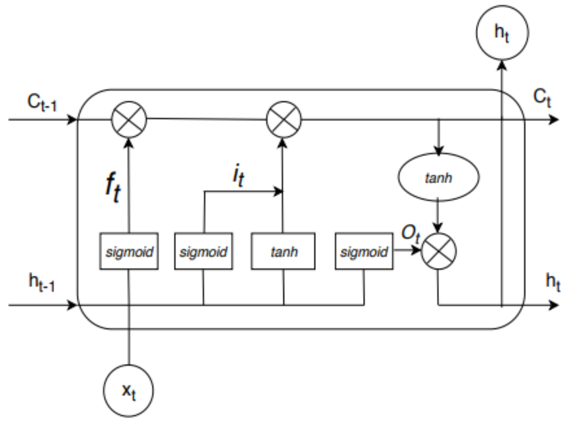

Figure 4: Structure of a LSTM block

The function used to predict new values is equal to

$$\boxed{f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)G(x_i, x) + b} \tag{8}$$

3.2 Long Short Term Memory (LSTM)

LSTMs are a special kind of Recurrent Neural Network (RNN) capable of learning long term dependencies. The popularity of LSTM in financial world comes from there potential to predict time-series data remembering the past data in their chain like structure. This ability of LSTMs to retain or forget the information comes from their special structured cells.

The architecture of LSTM cell is illustrated in the Figure 4 in which x_t, h_t, and c_t are defined as the input state, hidden state and the cell state respectively at time t.

LSTM cell has three gates: f_t, i_t, and o_t which are called as forget gate, input gate and output gate respectively.Gates as the name suggests, act as a barrier to information and only allows the useful information into the model. These gates are composed of sigmoid neural net layer which outputs a number between 0 and 1 where zero means no information is allowed to pass through and one represents that everything is allowed through these gates.

The calculations of the forget gate (f_t), the input gate (i_t) ,the output gate (o_t), the cell state (c_t) and the hidden state (h_t) are performed as follows where the sigmoid function, $\sigma(z) = (1+e^{-z})^{-1}$ and the tanh function, $t(z) = \frac{2}{1+e^{-2z}} - 1$

$$f_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f) \tag{9}$$

$$i_t = \sigma(W_i * x_t + U_i * h_{t-1} + b_i) \tag{10}$$

$$o_t = \sigma(W_o * x_t + U_o * h_{t-1} + b_o) \tag{11}$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_c * x_t + U_c * h_{t-1} + b_c) \tag{12}$$

$$h_t = o_t * tanh(c_t) \tag{13}$$

where $W_{f/i/o}, U_{f/i/o}$ are the weights matrices and the $b_{f/i/o}$ are bias vectors. These vectors are calculated according while back propagating which updated the weights of the neural network, using Adam optimization algorithm.

3.3 Dataset and Preprocessing

This section will discuss all the data and its sources used in this project for the stock price prediction model. The dataset used to train and test our model can be classified into the fundamental and historical data for all four stocks(APPL, MSFT, AMZN and CMG) on NASDAQ. Fundamental data includes various corporate accounting statistics like Earning per share (EPS), Gross Profit, Total Revenue, and Operating Income. The historical data used comprised the open, high, low, close and the adjusted close price of the stocks. We used the Alpha Vantage API*(https://www.alphavantage.co/documentation/)* to directly extract and push the data into our python scripts for the historical price data. We used daily adjusted OHLC data with adjusted close prices(adjusted taking into account all corporate actions that affect the stock prices like stock splits, bonus issues, dividends etc.).

For the fundamental data, we used two data sources and two different types of data frequency. For the EPS, we used the alpha vantage API (Earnings) and extracted the quarterly data. Moreover, we used alpha vantage and yahoo finance for the other corporate accounting statistics mentioned above (Gross Profit, Total Revenue, Operating Income) to get the data. However, the frequency of these factors from alpha vantage was quarterly, but we extracted the yearly data from yahoo finance for these three factors.

Time Frame: We used the data going back from 2008 to 2021. However, the exact date from which our first data starts cannot be declared in advance as it would directly depend on the frequency of moving averages, and other features used in the model. For example, if the 200 days moving average is also a feature column in the data, then the first 200 data points would obviously not be included in the dataset.

Preprocessing: For all the accounting statistics, we have used the forward fill method to compute a daily figure in each row of the data starting from the first data point. We have directly used the figures of these statistics without any changes to them.

For the historical price data, the data matrix X has been made so that the open, high, low, and volume of the previous day is used to predict the price of the current day close price, which is our target variable here. Note, the target column is set to the 'Adjusted Close Price' instead of the 'close' column. The 'close' price column has been dropped from the data.

Since the features extracted from the input data are of different units and scale, normalization is needed to scale the data between 0 and 1, which will also help in faster convergence. To normalize our data, we use the minmaxscaler function provided by scikit-learn framework. This function gets the max and the min values of each column and performs the following formula:

$$\frac{x_i - \min_i x}{\max_i(x) - \min_i(x)}$$

Train-Validate-Test Split: The data has been split into training, validation and testing set to check model hyperparameters on the validation set and check model performance finally on the test set, in the ratio of 75:15:10 respectively.

3.4 Features

Features are an essential part of any machine learning algorithms. Every researcher has their research backing the validity of their assumptions in using those features. In this section, we discuss all the features that make up our data, and we explain how they can be helpful in our stock market prediction model.

- **Past Prices (Open/Close):** "The best prediction for tomorrow's stock price is today's stock price". Hence we use the lookback period parameter that we will discuss in the coming section to get the past prices and use them as features. For example, if the lookback period is ten, and we are considering closing prices, then each row of our data will contain the previous ten days' closing price as a feature.
- **Simple Moving Average(SMA):**Moving averages are running averages of a finite size window over a dataset used as a trend-following device. SMA are lagging indicators which means that the change in SMA is observed when the action in the stock market prices has already happened. It is the most widely used figure for observing the general trend in the market. SMA can be calculated for any number of days, which is the parameter of this feature. If we are considering N-days simple moving average at time t, we can calculate it as follows (here p_t is the price at time t)

$$SMA_{t,N} = \frac{p_t + p_{t-1} + p_{t-2} + ... + p_{t-N-1}}{N}$$

- **Accounting Statistics:** As discussed in the earlier section, we will include EPS, Gross Profit, Total revenue and Operating Income as the feature in our model. Earning per share is the accounting ratio which is a good measure to check the profitability of the company(at least in theory). Gross profit, total revenue and operating income are other figures that represent the operating capability of the company.
- **Relative Strength Index(RSI):** The Relative Strength Index (RSI), developed by J. Welles Wilder, is a momentum oscillator that measures the speed and change of price movements. The RSI oscillates between zero and 100.Traditionally the RSI is considered overbought when above 70 and oversold when below 30. Signals can be generated by looking for divergences and failure swings. RSI can also be used to identify the general trend.RSI is a leading indicator which means that it looks forward and give signal in advance of the price action. RSI is generally calculated for a 14 days period. Let p_t be the price of the stock at time t,

$$Change(t) = p_t - p_{t-1}$$

$$Gain(t) = \begin{cases} Change(t) & if\ Change(t) > 0 \\ 0 & otherwise \end{cases}$$

$$Loss(t) = \begin{cases} Change(t) & if\ Change(t) \leq 0 \\ 0 & otherwise \end{cases}$$

$$S_G(t) = \sum_{i=0}^{13} Gain(t)$$

$$L_G(t) = \sum_{i=0}^{13} Loss(t)$$

$$AvgGain = \frac{G(t-13) + G(t-12) + ... + G(t)}{14}$$

$$AvgLoss = \frac{L(t-13) + L(t-12) + ... + L(t)}{14}$$

$$RSI(t) = 100 - \frac{100}{1 + \frac{AvgGain}{AvgLoss}}$$

- **Moving Average Convergence Divergence (MACD):** MACD is a momentum oscillator primarily used to trade trends. Although it is an oscillator, it is not typically used to identify over bought or oversold conditions. It appears on the chart as two lines which oscillate without boundaries. The crossover of the two lines give trading signals similar to a two moving average system.MACD crossing above zero is considered bullish, while crossing below zero is bearish. Secondly, when MACD turns up from below zero it is considered bullish. When it turns down from above zero it is considered bearish. When the MACD line crosses from below to above the signal line, the indicator is considered bullish. The further below the zero line the stronger the signal.An approximated MACD can be calculated by subtracting the value of a 26 period Exponential Moving Average (EMA) from a 12 period EMA.
- **Average True Range(ATR):** ATR is the average of true ranges over the specified period. ATR measures volatility, taking into account any gaps in the price movement. Typically, the ATR calculation is based on 14 periods, which can be intraday, daily, weekly, or monthly. To measure recent volatility, use a shorter average, such as 2 to 10 periods. For longer-term volatility, use 20 to 50 periods.
  An expanding ATR indicates increased volatility in the market, with the range of each bar getting larger. A reversal in price with an increase in ATR would indicate strength behind that move. ATR is not directional so an expanding ATR can indicate selling pressure or buying pressure. High ATR values usually result from a sharp advance or decline and are unlikely to be sustained for extended periods.A low ATR value indicates a series of periods with small ranges (quiet days). These low ATR values are found during extended sideways price action, thus the lower volatility. A prolonged period of low ATR values may indicate a consolidation area and the possibility of a continuation move or reversal. ATR can be calculated as follows where p_t is the price at time t, $High_t$ is the highest of time t, Low_t is the lowest of time t, and n is the period.

$$TR(t) = \max\{High_t - Low_t\ , |High_t - p_{t-1}|\ , |Low_t - p_{t-1}|\}$$

$$ATR(t) = \frac{ATR(t-1) * (n-1) + TR(t)}{n}$$

- **Bollinger Bands:**Bollinger Bands are a type of price envelope developed by John Bollinger. (Price envelopes define upper and lower price range levels.) Bollinger Bands are envelopes plotted at a standard deviation level above and below a simple moving average of the price. Because the distance of the bands is based on standard deviation, they adjust to volatility swings in the underlying price. Bollinger Bands use 2 parameters, Period and Standard Deviations, StdDev. The default values are 20 for period, and 2 for standard deviations, although you may customize the combinations.
  When the bands tighten during a period of low volatility, it raises the likelihood of a sharp price move in either direction. This may begin a trending move. Watch out for a false move in opposite direction which reverses before the proper trend begins. When the bands separate by an unusual large amount, volatility increases and any existing trend may be ending.
  First, calculate a simple moving average. Next, calculate the standard deviation over the same number of periods as the simple moving average. For the upper band, add the standard deviation to the moving average. For the lower band, subtract the standard deviation from the moving average.
- **On Balance Volumes:** On Balance Volume (OBV) measures buying and selling pressure as a cumulative indicator that adds volume on up days and subtracts volume on down days. When the security closes higher than the previous close, all of the day's volume is considered up-volume. When the security closes lower than the previous close, all of the day's volume is considered down-volume.
  When both price and OBV are making higher peaks and higher troughs, the upward trend is likely to continue. When both price and OBV are making lower peaks and lower troughs, the downward trend is likely to continue. During a trading range, if the OBV is rising, accumulation may be taking place—a warning of an upward breakout. During a trading range, if the OBV is falling, distribution may be taking place—a warning of a downward breakout.
  On Balance Volume is calculated by adding the day's volume to a cumulative total when the security's price closes up, and subtracting the day's volume when the security's price closes down.

$$OBV(t) = \begin{cases} OBV(t-1) + Volume(t) & if\ p_t > p_{t-1} \\ OBV(t-1) - Volume(t) & if\ p_t < p_{t-1} \\ OBV(t-1) & if\ p_t = p_{t-1} \end{cases}$$

- **Average Directional Moving Index:** can be used to help measure the overall strength of a trend. The ADX indicator is an average of expanding price range values. The ADX is a component of the Directional Movement System developed by Welles Wilder. This system attempts to measure the strength of price movement in positive and negative direction using the DMI+ and DMI- indicators along with the ADX. Wilder suggests that a strong trend is present when ADX is above 25 and no trend is present when below 20. When the ADX turns down from high values, then the trend may be ending. You may want to do additional research to determine if closing open positions is appropriate for you. If the ADX is declining, it could be an indication that the market is becoming less directional, and the current trend is weakening. You may want to avoid trading trend systems as the trend changes. If the ADX is rising then the market is showing a strengthening trend. The value of the ADX is proportional to the slope of the trend. The slope of the ADX line is proportional to the acceleration of the price movement (changing trend slope). If the trend is a constant slope then the ADX value tends to flatten out.

4 EXPERIMENTS AND RESULTS

In this project, we will be using data from the past to predict the return on the next trading day. To assess the model, our primary model performance metric is Mean Squared Error (MSE). The MSE will be performed on the test data, which is totally unseen from the training and development stages.It is calculated between our predicted price and the true price.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

One more thing to note is that we only used the train data once to train the model and calculated the parameters of the model. And then, with those results, we tried predicting the stock prices on test data, so although we are predicting daily prices, ideally, after each days' prediction, we should have updated our training data with the predicted value and then fit the model again to predict next day's value. However, this process of updating training data, again and again, is very computationally expensive, so we could not follow that technique. This technique of updating the training model each time a new prediction is made is called Online learning. This is a beneficial and famous method, especially in financial data, since data in finance is a sequential flow of continuous price data.

4.1 SVR Results

In the earlier discussion, we saw a very detailed explanation of how Support vector regression models work. In this section, we will compare the stock prediction model of all four companies (AAPL, MSFT, CMG and AMZN) built on the SVR algorithm. In all the SVR model we used the scikit learn's svm module to fit the model. For the hyperparameter tuning, we used GridSearchCV. The parameter search space used in the grid search is shown in Table 2.

After putting these results into the GridsearchCV model the best model that came out are shown in Table 1. And the results of MSE are shown in Table 3. We observe that the SVR is performing quite good on the AAPL and the MSFT but it did not perform that well on CMG and AMZN. The best parameter value taken by the parameter C is 2400 and 2600 in CMG and AMZN respectively. Both of these values are the last values in the list of parameter search space. Similar thing can also be said for the gamma parameter. This

Table 1: Best SVR Model by GridsearchCV

Stock Ticker Kernel	C	Gamma
AAPL linear	100	0.001
MSFT linear	1200	10^{-12}
CMG linear	2400	10^{-22}
AMZN linear	2600	10^{-26}

Table 2: SVR Parameter search space summary of all stocks

Stock Ticker Kernel	C	Gamma
AAPL ['linear', 'rbf']	[1,10,50,100]	[0.001,0.005,0.01,0.05,]
MSFT ['linear', 'rbf']	[400,500,700,900,1200]	[1e-9,1e-8,1e-7,1e-6]
CMG ['linear', 'rbf']	[1800, 2000, 2200, 2400]	[1e-22,1e-20,1e-18,1e-16,1e-15]
AMZN ['linear', 'rbf']	[1500,1800,2000,2200,2400,2600]	[1e-20, 1e-18,1e-16, 1e-14]

Table 3: MSE of different stocks using Best SVR

Stock Ticker Test MSE	Train MSE	Val MSE
AAPL 2.80105	0.04834	0.76865
MSFT 4.72526	0.12730	4.20808
CMG 250.12444	19.67572	230.97951
AMZN 1060.91431	72.30711	358.29510

means that if we increase the search space of these two parameters we can hope to find even better solutions for CMG and AMZN.

Figure 5 shows us that not all features are adding value to our model. This plot shows that for the stock of AAPL, the five most important feature that are very significant in predicting stock prices are MACD, MACD signal, Macd fast, and SMA close 20. Also there are many redundant feature in this model that are not able to predict or have any affect on AAPL prices like, OBV, SMA 100, SMA 200 and ADX. This graph clearly shows that it is possible to have redundant features in the model even after carefully researching the factor that affect the stock prices.

Feature importance did not vary much across the companies. The factors which were important in predicting AAPL stock price were also important in other companies as well. Although there were some exceptions and some of the features did change their importance rank but not by too much. Figure 6 show the prediction on AAPL test set.

5 CITATIONS

Some great work that already has been done in this areas. [1][3][4][6][2][5]

ACKNOWLEDGMENTS

I am grateful to Professor Amit Sethi for his teaching on this subject and his help and guidance. I am very thankful to my project mentor Arun Jain, who have spent a lot of time suggesting new idea and techniques to get better results on the model.

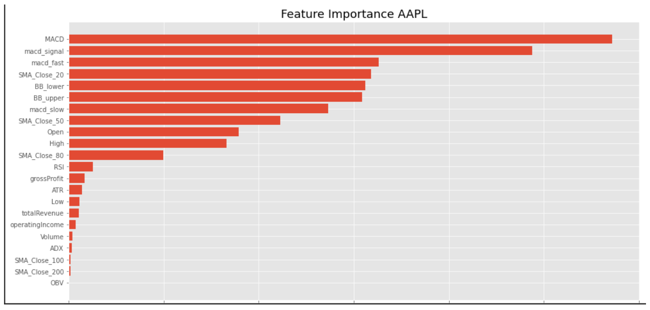

Figure 5: Feature Importance of AAPL SVR

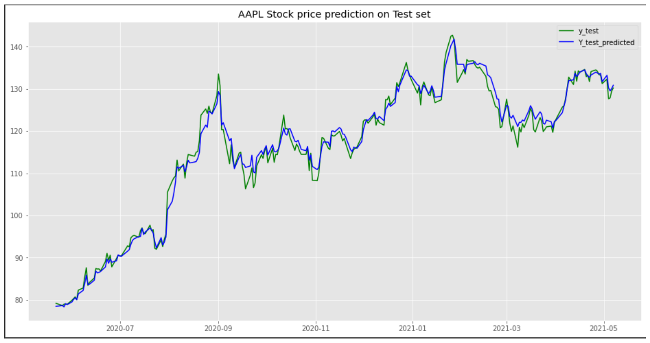

Figure 6: Prediction using SVR on AAPL

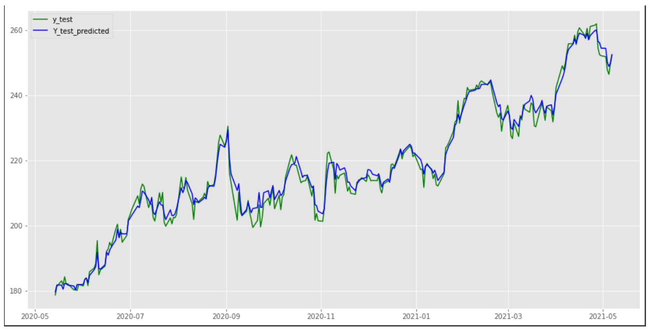

Figure 7: Prediction using SVR on MSFT

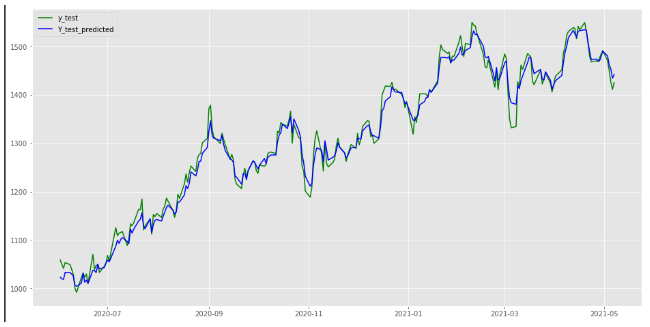

Figure 8: Prediction using SVR on CMGL

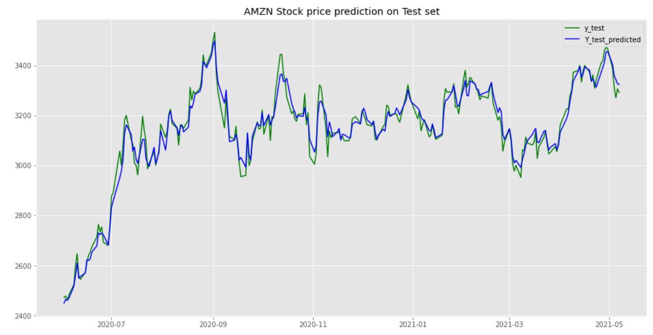

Figure 9: Prediction using SVR on AMZN

REFERENCES

[1] C Fischer, T.; Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. (2018).

[2] Mallikarjuna Shastry P. M. Pramod B S *. May - June 2020. Stock Price Prediction Using LSTM. (May - June 2020).

[3] Tongda Zhang Shunrong Shen, Haomiao Jiang. [n.d.]. Stock Market Forecasting Using Machine Learning Algorithms. ([n. d.]).

[4] Tongda Zhang Shunrong Shen, Haomiao Jiang. [n.d.]. Stock Market Forecasting Using Machine Learning Algorithms. ([n. d.]).

[5] Seokhoon Yoon Thi-Thu Nguyen. OCt 2019. A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning. (OCt 2019).

[6] Zihao Qu Zhichao Zou. 2020. Using LSTM in Stock prediction and Quantitative Trading. (2020).