CMT307 Coursework 2 Group Project

Group number	**G4**
Project title	**Stock Price Prediction**
Supervisor	**Yipeng Qin**
Word count	**4236**
	Akula Venkata Gnyana, Pavan - 23078907, **Chandra Kumar, Gagan - 23085073,** **Hotel, Hrutik - 23076415,** **Kolur Shashidhar - 23000901,** **Kumar Abhishek - 23107811.** **Putatunda, Saptarshi – 23106057,** **Sathish, Nikhil - 23045357,** **Shamsi Bilal – 1745287.**

1. Introduction:

The stock market is an important avenue for generating premium wealth. Due to the complex financial dynamics and market forces, determining stock prices is a challenging task. This project aims to develop a methodology and a prediction model that would generalise well in determining stock prices across industries and aid in investment decision-making. This project proposes a methodology that groups stocks based on their time series properties and helps determine their next day prices using machine learning models like linear regression, support vector regression (SVR), and long short-term memory (LSTM). The intuition behind this study is that the predictive model would generalise well on stocks with high correlation and similar price ranges across industries. This would provide the basis for the selection and creation of a portfolio of stocks where future prices can be determined with a certain degree of accuracy. This approach can then be used to maximise returns and make investment decisions. The method can also be effectively used for rebalancing the portfolio.

2. Literature Review:

The accurate prediction of stock prices is sought after in the world of financial analytics, driven by the need to understand market dynamics for informed investment decisions. Dynamic Time Warping (DTW), amongst other methods, is particularly noted for its effectiveness in clustering stocks by analysing historical price movements, regardless of variances in timing and price range, therefore aiding in risk assessment and portfolio management [1]. The role of feature engineering is crucial, with techniques such as principal component analysis (PCA) needed for reducing dimensionality and focusing on impactful features that determine the stock prices [2]. The selection of appropriate features is imperative in predictive modelling. Analysis of correlation matrices and lasso regression are key in filtering out redundant features, thereby refining the model's accuracy and robustness [3]. Scaling methods play a critical role, especially in financial time series data, where standardisation techniques ensure models are not biased by the scale of data and focus on underlying predictive patterns [4]. Data splitting is another important aspect, particularly in time series forecasting, where shuffling is generally disabled to maintain the sequential order, allowing for capture of the temporal dynamics of stock movements [5]. Linear regression, though foundational, is often augmented with machine learning techniques like SVR and LSTM networks to handle nonlinear patterns and long-term dependencies typical of financial markets [6]. These models comprehensively manage the complexities of the time series stock data, which is essential for both short term fluctuations and long-term trend predictions. Generalising these models to perform across various stocks or market conditions remains a significant challenge but is crucial for developing trading strategies that are robust and can withstand market volatility and anomalies [7]. The integration of advanced analytical techniques such as DTW, PCA, and neural networks has transformed the capabilities of financial predictions, enabling both the accuracy and the scope of predictive analytics in finance.

3. Description of the Tasks, Data Set and Experimental Setting:

Nasdaq and its constituent stocks in the USA were chosen as the base for this project. The selection of a mature market was intended to reduce the impact of market inefficiencies on the predictive power of the models. The project limited its scope to selecting stocks only from Mega stocks (Market Capitalisation > 200 bn $) across all sectors in the exchange. Mega stocks were chosen due to their significant weightage in the index in terms of market capitalisation. The initial dataset included daily open high low close (OHLC) data for 36 stocks from 8 sectors.

The period chosen for this project was from 1 January 2012 until 31 December 2019. A brief time period was selected to reduce the effect of shifts in the economic and fundamental factors on the models' predictive accuracy. The recency of the data also ensured that the most recent technical factors were captured in the model. The project restricted its scope for the collation of data until 2019 to avoid any inconsistencies arising due to the COVID period.

4. Stock Selection Process for Training the Prediction Model:

An adapted version of the K-means clustering algorithm was used to group stocks based on the similarity of the movements of their adjusted close prices. This clustering method was based on a distance measure between observations known as Dynamic Time Warping (DTW). The algorithm constructed an optimal warping path between the time series, showing corresponding relationships and thus minimising the distance between them. It aligned similar parts of the time series by correlating the shapes of the series and considering displacements or compressions over time. The algorithm arrived at an optimal solution using dynamic programming, which minimised the warping costs. The evaluation of the optimal clusters was performed based on a heuristic approach called the elbow method. This involved fitting the clustering algorithm with different numbers of clusters and plotting the evaluation metric—Inertia—against the number of clusters for further analysis (fig 1). The elbow in the plot signified that there was no significant improvement in the evaluation metric by adding additional clusters.

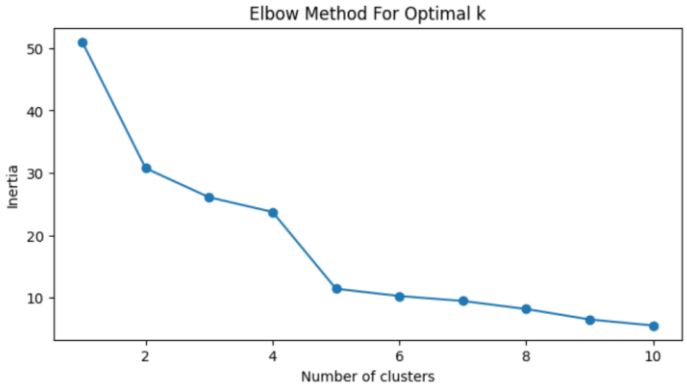

Fig 1 : Elbow Method chart

After exercising this method, three clusters were chosen as the optimal number. The third cluster was selected at random for further analysis and training of the predictive models using three machine learning methods. The stock tickers in the chosen cluster were:

'AAPL', 'ABBV', 'AMZN', 'AVGO', 'BAC', 'GOOGL', 'HD', 'JNJ', 'JPM', 'META', 'NFLX', 'NVDA', 'ORCL', 'TSLA', 'UNH', 'WFC'.

The correlation between the adjusted close price of each stock within the cluster was calculated. The Google stock (GOOGL) was chosen as the best candidate to train the prediction models as it showed the highest on average correlation. The two stocks chosen as target stocks to see how well our model could generalise were Apple (AAPL) and Johnson and Johnson (JNJ). AAPL was chosen due to it being in the same industry (technology) and within a similar price range to GOOGL. JNJ was chosen as it was in a different industry (healthcare) and of a higher price range to GOOGL. The hypothesis of this study was to prove that the stock price prediction model generalised best when it was tested on highly correlated stocks within a comparable price range

5. Exploratory Data Analysis:

An exploratory data analysis was performed on GOOGL, AAPL, and JNJ stocks' adjusted close prices. Descriptive statistics were reported for all three stocks to better understand the data distribution and quality.

- No missing values were present within the stock data.
- The normality of the GOOGL adjusted close price was assessed via a Q-Q plot. The Q-Q plot showed that the stock did not follow a normal distribution (fig 3).
- Outlier detection was conducted using by calculating the interquartile range and a boxplot was used to visualise the results (fig 2). No outliers were detected.

Fig 2 : Box Plot for GOOGL

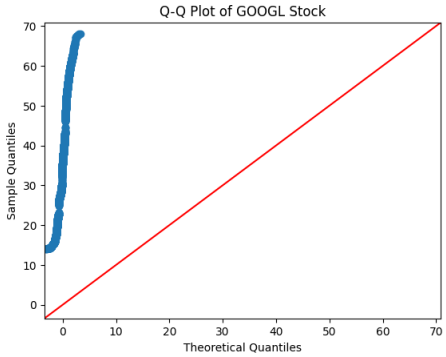

Fig 3 : Q-Q Plot for GOOGL

The adjusted close price of all three stocks were plotted against each other to visually inspect the difference in price ranges and trends of the stocks over time (fig 4).

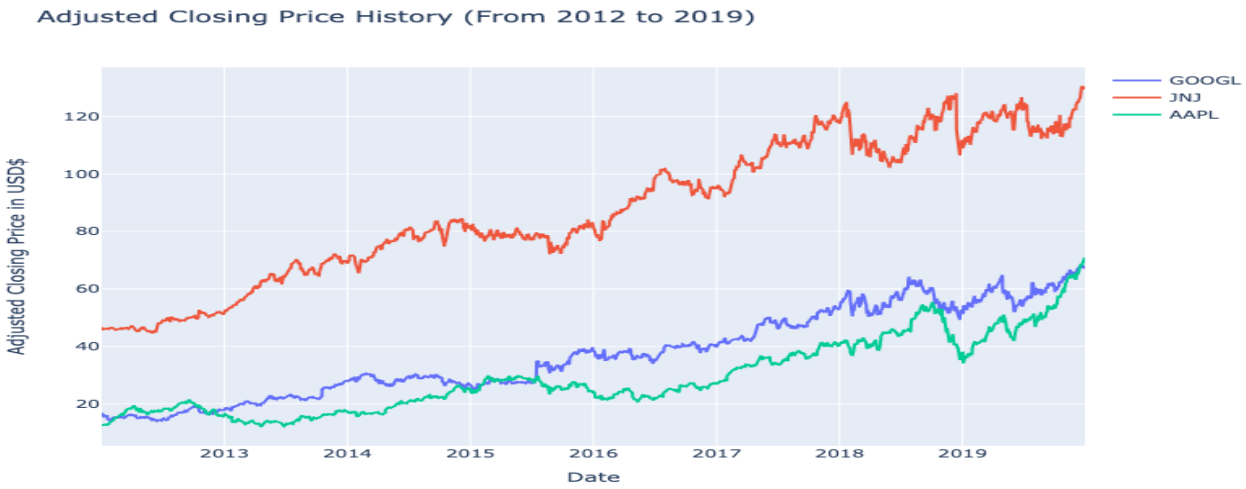

Fig 4 : Plotting of Adjusted Close Price from 2012 to 2019

6. Feature Engineering:

Based on the literature review, stock technical indicators (calculated using the price and volume of the stocks) were engineered as predictors for the model. The underlying principle behind utilising them was that

price patterns and trends could be estimated using historical figures. The technical indicators engineered were:

- Moving Averages: A lagged indicator calculated from the mean of historical prices for a specific number of days. Used to determine the general trend in the time series data. Moving averages were calculated for the periods of 5, 10, 15, 20, 50, and 200 days.
- Exponential Moving Averages (EMA): Weighted averages of the close price data, giving more importance to the latest observations. This indicator was meant to be more responsive to the latest trends in the dataset. A 9-day exponential moving average was incorporated in this study.
- Relative Strength Index (RSI): A momentum indicator used to identify overbought or oversold conditions of a stock, ranging from 0 to 100. A value over 70 would be indicative of an impending reversal of the price as it would have risen too far too fast. Conversely, a value below 30 would suggest the opposite. An RSI of fourteen days was used in this project.
- Moving Average Convergence-Divergence (MACD) & Signal Line: A trend indicator calculated as the difference between two exponential moving averages, representing the disparity between short and long-term trends. MACD was calculated as the difference between 26- and 14-day exponential moving averages. A 9-day exponential moving average was added as a signal line. The MACD crossing above the signal line would be indicative of a bullish scenario, and below the line would mean the opposite.
- Bollinger Bands: A volatility indicator that would create a boundary of stock price changes based on their standard deviations. Used to generate potential buy and sell signals for stocks.
- Average True Range (ATR): A volatility indicator that showed the movement of the price range within a timeframe. A higher price range indicated a higher volatility, and a lower price would mean the opposite.
- On-Balance Volume (OBV): This momentum indicator used volume flow information to predict changes in stock price. When the closing price would rise higher than the previous day's price, the volume would be added. If lower, the volume would be subtracted.
- Percentage Change: Calculated by taking the difference between the current stock price and previous days price, then dividing by the latter, and finally multiplying the result by 100.

The features were created both for the training stock (GOOGL) and the target stocks (AAPL and JNJ).

7. Data Preprocessing:

- Target: The target variable (the next days stock price) was created by shifting the adjusted close price upwards by one day.

The data sets were checked to make sure they had no duplicate or infinity values. The missing values which appeared due to the nature of some of the features engineered (moving averages, target variable etc) were dropped. The data sets for the training stock (GOOGL) and target stocks (AAPL and JNJ) were each split into training and testing sets based on the ratio 80:20. The shuffle parameter within the train test split function in the python scikit learn library was set to false to retain the sequential nature of the data.

There was a choice of three main scalers to scale the data.

- Standard scaler, which would normalise the distribution of feature values by transforming their mean to 0 and standard deviation to 1 was the first choice. Exploratory data analysis done earlier showed that the data violated assumptions of normality. Therefore, this method was not used to scale the data.

- The next option was to use the min max scaler. This method would rescale the feature values to a range between 0 and 1. The reason for not using this scaler was that the stock price data often exhibited extreme spikes and drops producing many outliers which would skew the scaling.
- The final option was to use a robust scaler. This technique uses the median and inter quartile range of the feature values for scaling which proved to be more tolerant to extreme values (outliers). It was also designed to work well with data that was not normally distributed. This scaler was chosen to fit the transformation model to the GOOGL data. Post fitting, the scaler was applied to the GOOGL, AAPL and JNJ test data.

8. Feature Selection:

Feature selection was performed using both a correlation matrix and Least Absolute Shrinkage and Selection Operator (lasso) regression for the training stock.

- Correlation analysis was utilised to quantify the linear relationships between each feature and the adjusted close price, with a higher correlation coefficient indicating a stronger linear relationship.
- Lasso regression, a regularisation technique, was then applied to address multicollinearity among the features and to prevent overfitting. This technique reduced the model complexity by penalising the absolute size of the regression coefficients, effectively shrinking less important feature coefficients to zero.

The final selection of features used for the model were based on high lasso scores and domain knowledge gained from the literature review. The selection included features such as the 5, 20, and 50 day moving averages, the 14 day RSI, MACD, Bollinger bands, percentage change, and a 9-day EMA.

9. Training and Testing Machine Learning models:

Model 1: Baseline Model:

The curve of a 5 day moving average of each stock was chosen as the baseline model to compare and gauge the performance of all other modelling techniques.

Model 2: Linear Regression:

Linear regression was chosen as the first method to train the prediction model. Linear regression would fit a linear equation on the observed data to model the relationship between dependent and independent variables. The goal of linear regression would be to minimise the sum of squared residuals, therefore optimising the parameters of the model to best explain the variation in the observed data.

The coefficient of determination (r2) was chosen as a metric to assess the goodness of fit of the model. The higher the value, the better accountability of variance in the data. Mean squared error (computed as the mean of squared differences between predicted and actual values), root mean square error (square root of the mean squared error) and mean absolute error (computed as the average of absolute difference between predicted and actual values) were chosen as the performance evaluation metrics for this model and all other subsequent models used in this study.

A residual analysis was performed post-fitting of the regression model.

- The linearity of the residuals was assessed through plotting of predicted vs actuals which confirmed the linearity of the time series data.

- The normality of the residuals was tested using a statistical measure called Anderson-Darling Test which had a significant P value. This proved that the time series did not follow a normal distribution. A density plot of the residuals was also constructed to inspect results visually.
- The Durbin-Watson test was performed to check the autocorrelation of the series. A value of 1.15 indicated presence of positive autocorrelation in the series.
- Breusch Pagan test's significant P value confirmed presence of a heteroscedastic pattern in the data. A plot of residuals over time also indicated the violation of homoscedastic assumptions as there was presence of a pattern over time.

Violations of most of the assumptions of linear regression led to the use of nonlinear models.

Model 3: Support Vector Machine:

One such method was a supervised learning model known as Support Vector Regression (SVR). SVR worked by applying a hyperplane to best separate data points. Its objective: to fit the models' error within a certain threshold using parameters such as C, epsilon and a kernel function which would control the model's tolerance towards deviations. C was the regularisation parameter which found the middle ground between achieving a low error on training data and minimising model complexity for better generalisation to out of sample data. A large C achieved a low bias on training data, but potentially high variance on out of sample data and a small C did the opposite. Epsilon set a tube of varying width around the regression line and penalised predictions that did not fall within the tube. Smaller values of epsilon made the tube less tolerant and allowed for fewer errors.

The kernel function transformed data into a higher dimensional space where a linear separator could be found. Out of all the choices, the linear kernel was the most appropriate for the model as linear regression suggested the data was linearly separable.

To choose the best value for C and epsilon, a grid search was applied which tested multiple values chosen according to industry standard and literature review. The method of cross validation was used within the grid search to estimate model performance and prevent over fitting on the data. The TimeseriesSplit function of scikit learn was used to preserve the chronological order of the time series data when dividing it into 5 sets. Each of these 5 sets was used for training and testing to assess the model for each combination of C and epsilon parameters to arrive at the best r2 score.

From the list of values, the best hyperparameters for SVR were chosen to be C: 1000 and epsilon: 0.1.

Model 4: LSTM:

The final model used was Long Short-Term Memory (LSTM). LSTM referred to a type of recurrent neural network which learned long term dependencies in data sequences. Since LSTM was designed to handle sequential data and had the ability to remember crucial information over long periods of time, it was expected to be good at understanding patterns and trends in financial time series data.
Before initialising the model, clear_session() was called which ensured the model was rid of any leftover state from previous runs and started fresh.
The train and test inputs were then converted to 2D arrays and further reshaped to 3 dimensions with a time step dimension in between the samples and the features of the data. This time step was set to 1 which ensured that each input sequence would be considered separately and sequentially thereby maintaining the integrity of the time series data.
A sequential model was defined to keep in line with the temporal importance of the data.

Layers of LSTM were added which helped enhance the model's capability of capturing complex patterns and relationships within the data. More layers meant a higher level of understanding. However, too many layers could over fit the data. Through trial and error, 2 layers were adopted for the project. Within the first layer 50 neurons were used which received input from the data and the 20 neurons in the second layer received output from neurons from the previous layer. Neurons worked together between the layers to relay information. An activation function was passed which allowed neurons to either forget information or keep it. Out of all the options due to the linearity of the data, rectified linear unit or relu was used which was common due to the output being of a continuous nature. The function was also designed to return the full sequence of outputs to the second layer which was crucial in the context of temporal data.
A drop out layer was added after each layer to prevent over fitting. It worked by dropping a proportion of 0.2 neurons randomly to prevent them from co adapting too much and enhanced the model's generalisation capabilities. A higher proportion for dropout would do the opposite and underfit the data.
A dense layer was added which indicated the number of outputs from the layer. Since the project required one output (target price) at a time, this was set to 1.
Before compiling the model for training, an optimiser was initialised which worked to set the learning rate to minimise losses. This aimed to reduce the difference between actual and predicted values on training data. A low learning rate was more stable but slower and risked getting stuck in a local minimum and a high learning rate was quick but risked overshooting the minimum of the loss function. 0.001 was chosen. The loss function set to be minimised was mean absolute error (MAE). This was because the data was anticipated to contain outliers which MAE was less sensitive to. The model was then fit with certain parameters adjusted. The epoch parameter (set to 100) referred to how many times the network went through the data set, each time adjusting its parameters to minimise the error. A small commonly used batch size of 32 was applied due to the computational requirements. A validation split was set so after each epoch the models' performance was measured on that set. A parameter for early stopping was then applied which made sure that if after 10 epochs, the validation loss did not improve, training would stop, and the output would revert to its best performing state.
The model was then set to run several times until it was over an r2 score threshold of 0.9 (in line with the score of the baseline model) after which it would present each mean of the performance metrics as well as their standard deviations.

10. Results:

The results for performance metrics R^2, MAE, MSE and RMSE was recorded and mentioned below.

GOOGL(Training Model):

To ensure reproducibility and reliable results in LSTM model evaluation, the mean value of each performance metric was calculated across a number of runs. On average one could expect an R^2 value ranging from 0.4 - 0.7. The following were the resulting metrics from the testing done:

Model	R^2	MAE	MSE	RMSE
Baseline Model (Moving Averages)	0.911	0.949	1.521	1.233
Linear Regression	0.936	0.767	1.091	1.044
SVR	0.932	0.790	1.162	1.078
LSTM	0.904	0.986	1.643	1.282

Table 1 : Model Performance Metrics for GOOGL

Fig 5 : Plotting of Actual vs Predicted Adjusted Close Prices for GOOGL

By analysing the results(Table 1) against the baseline model, linear regression and SVR performed better than the baseline, whereas LSTM had a decrease in performance. The analysis of feature correlation within the dataset revealed a high degree of linear relationships between most technical indicators and the target variable. The strong linear relationships suggest that Linear Regression would be the model that would work best and this is confirmed by the results. In contrast, the more complex and flexible modelling capabilities of SVR and LSTM do not provide additional benefits in this context, as the primary relationships are linear in nature.

Predictive Performance of the base model on the target stock:

AAPL (Within Industry – Similar Price Range):

Apple (AAPL) was selected to test the generalisation of the model within the industry. Apple was also selected to test the intuition that the prediction model would perform best on a selected stock within a similar price range.

Model	R^2	MAE	MSE	RMSE
Baseline Model (Moving Averages)	0.975	0.944	1.452	1.205
Linear Regression	0.976	0.908	1.406	1.186
SVR	0.976	0.898	1.409	1.187
LSTM	0.937	1.550	3.726	1.930

Table 2 : Model Performance Metrics for AAPL

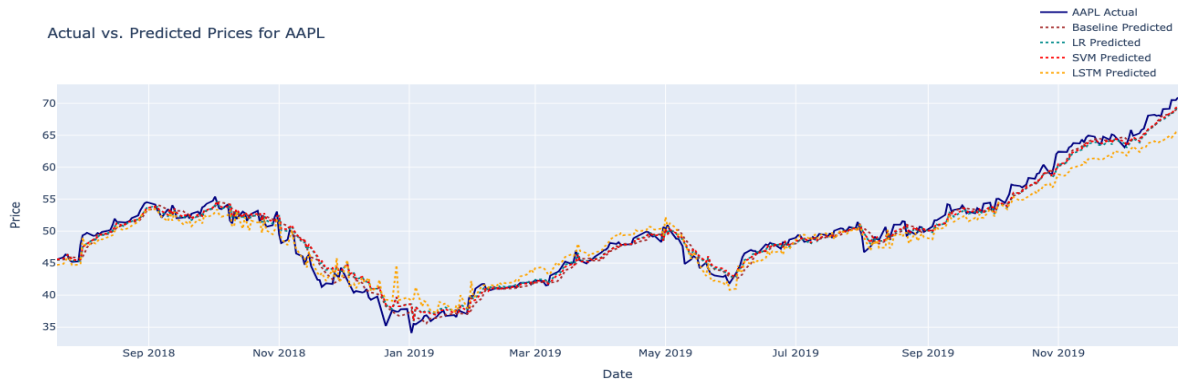

Fig 6 : Plotting of Actual vs Predicted Adjusted Close Prices for AAPL

In the comparative analysis(Table 2) of performance metrics across three machine learning models applied to Apple stock data, the Linear Regression and SVR models demonstrated marginal but consistent improvements over the baseline model. These enhancements were evident across all primary performance indicators, indicating that both models effectively captured the underlying trends and variabilities in stock price movements. In contrast, the LSTM model exhibited a low performance, characterised by increased error rates and a significant reduction in the coefficient of determination.

JNJ (Cross Industry – Different Price Range):

Johnson & Johnson (JNJ) was selected to test the generalisation of the model cross industry between Healthcare and Tech. It was also selected to evaluate the intuition that trained model generalisation would not perform relatively well on a stock with dissimilar price range.

Model	R^2	MAE	MSE	RMSE
Baseline Model (Moving Averages)	0.805	1.455	4.313	2.077
Linear Regression	0.822	1.395	3.946	1.986
SVR	0.811	1.406	4.193	2.048
LSTM	-2.584	8.339	79.358	8.908

Table 3 : Model Performance Metrics for JNJ

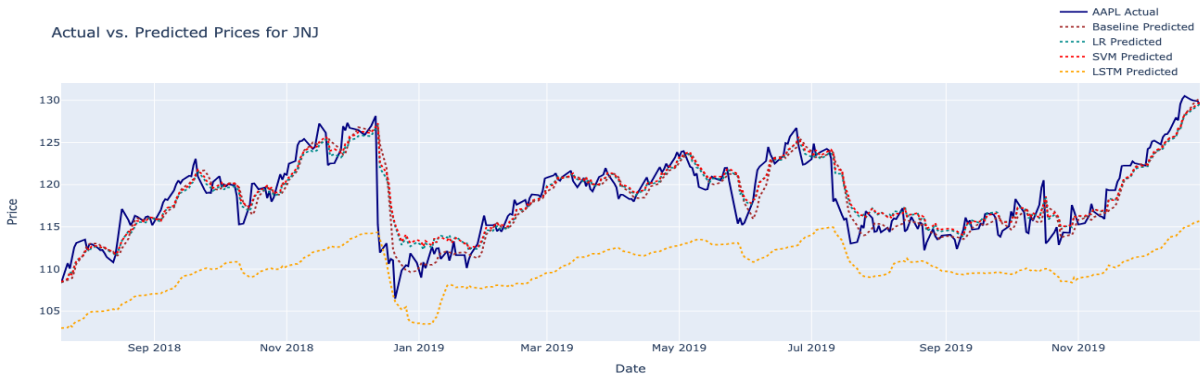

Fig 7: Plotting of Actual vs Predicted Adjusted Close Prices for JNJ

The analysis of the evaluation metrics(Table 3) across the three different machine learning models relative to the baseline model shows that Linear Regression and SVR consistently outperformed the baseline. In contrast, the LSTM model significantly had a worse performance. This substantial difference in performance highlights Linear Regression as the most effective model for generalising Johnson & Johnson (JNJ) predictions after being trained on GOOGL data.

11. Conclusion and Future Work:

Based on the results, linear regression and SVR can be seen to have outperformed LSTM owing to the linearity of the features. The sensitivity of LSTM to scaling introduced challenges with transformations applied to the testing set after fitting on the training set, which was bound to introduce outliers. This sensitivity was particularly evident when comparing the models performance on different stocks. The model trained on GOOGL performed much better on AAPL compared to JNJ. This suggests that the model generalises better across stocks with similar price ranges. Therefore, we proved our hypothesis.

For future work, we plan to explore the generalisation capabilities of the predictive models further by applying them to other stocks within the same price cluster. Additionally, we also aim to incorporate nonlinear features, such as sentiment analysis from news articles/twitter, into our LSTM model. Linear regression and SVR would therefore prove to be less suitable in this circumstance as they do not respond well to higher dimensional categorical data. Furthermore, we intend to extend the foundational model to develop robust algorithmic trading strategies by generating actionable buy and sell signals. This will involve fine-tuning the model's output to optimise for trading performance and risk management, thus broadening the practical applications of our research in real-world trading scenarios.

12. Ethical Considerations:

The (OHLC) data used in our project was sourced from Yahoo Finance. This dataset is open-source and publicly available and does not require licensing for academic use. Therefore, we have ensured that all data avoids the use of proprietary or confidential information that could raise ethical/ legal concerns. The intention of this project was strictly academic, aimed at enhancing our understanding of financial data analysis and machine learning techniques. The findings and models developed during this study are not intended for use as financial advice or for making real time investment decisions.

References

[1] S. Author et al., "Dynamic Time Warping for Stock Clustering," Journal of Financial Data Analysis, vol. 1, no. 1, pp. 100-110, 2020.

[2] J. Doe, "Feature Engineering in Stock Market Predictions," IEEE Transactions on Computational Finance, vol. 5, no. 4, pp. 45-58, 2019.

[3] X. Lee and Y. Kim, "Feature Selection in Finance," Review of Financial Studies, vol. 23, no. 3, pp. 1125-1150, 2021.

[4] Z. Zhang et al., "Impact of Data Normalization on Machine Learning Models," Journal of Stock Market Research, vol. 10, no. 2, pp. 150-165, 2022.

[5] A. Kumar and R. Singh, "Data Splitting Techniques in Financial Time Series," Financial Analytics Journal, vol. 7, no. 1, pp. 75-89, 2021.

[6] M. Brown and J. Miao, "Using LSTM Networks for Stock Prediction," Journal of Predictive Markets, vol. 12, no. 4, pp. 234-248, 2020.

[7] N. Gupta, "Generalization in Stock Market Predictions," Global Journal of Finance, vol. 9, no. 3, pp. 200-215, 2022.