CSCI-B 565 DATA MINING

DATA MINING ON S&P 500 INDEX COMPANIES

Project Report

Submitted By

Arvind Kumar Nalli Kuppuswami

Jayanth Kodur Kumar

Siddharth Choudhary

ABSTRACT

Stock markets, as always are volatile in nature and hence it’s difficult to predict the accurate price of stocks of companies. Stock price prediction helps us to try and get to know future movement of stocks of companies. This can be achieved through data mining techniques such as clustering and prediction. Clustering is a process where data items are grouped according to similarities. Prediction technique is defined as the process of determining missing value for an observation in a dataset based on previous data.

KEY WORDS: Clustering, Dendrogram, Silhouette Score, LSTM.

1. INTRODUCTION

The S&P 500(Standard and Poor’s 500) is a stock market index which measures the performance of stocks of the top 500 companies listed in the United States of America that includes NYSE, Nasdaq, and CBOE BZX Exchange. Our dataset has the list of all the S&P 500 index companies and their stock information over the period of 5 years (2013 to 2018) along with the collection of individual datasets representing each company. The goal of the project is to perform stock price analysis and predict future prices using time-series analysis by taking the closing price at the end of each month.

2. DATASET

Our dataset has attributes for all the 5 years for 500 companies such as:

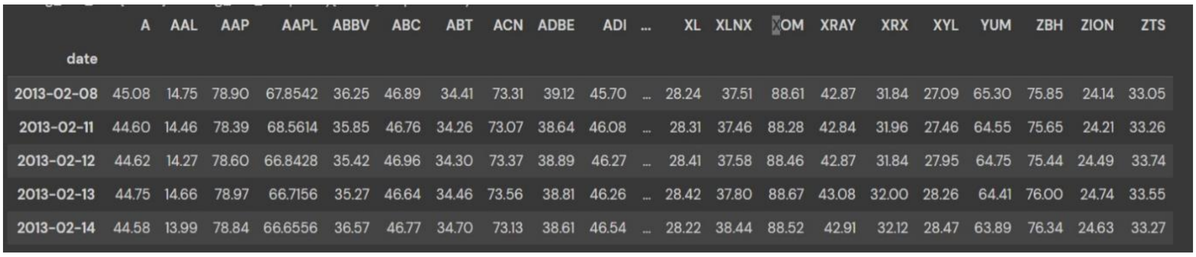
1. date
2. open
3. high
4. low
5. close
6. volume
7. Name

The open attribute denotes the opening price of the stock, high and low attributes denote the highest price and lowest price of that stock in that day respectively. The close attribute denotes the closing price of that stock in that day. Volume is the total number of traded shares for that day. Name refers to the company. Here, close is our target variable. Similarly, we have the same kind of dataset but for individual companies. In other words, we have individual datasets that has attributes for that company.

3. METHODS AND RESULTS

3.1 DATA PREPROCESSING

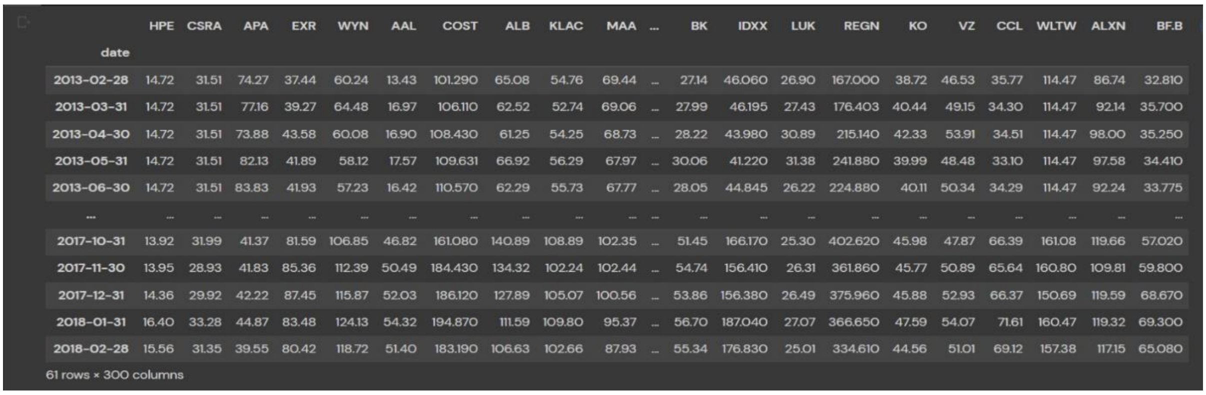
All the individual datasets have been taken and collated into a single dataframe where company names are columns and dates are indices. Each row has its corresponding closing price. Fig. 1 shows the resultant collated dataframe.



	A	AAL	AAP	AAPL	ABBV	ABC	ABT	ACN	ADBE	ADI	...	XL	XLNX	XOM	XRAY	XRX	XYL	YUM	ZBH	ZION	ZTS
date																					
2013-02-08	45.08	14.75	78.90	67.8542	36.25	46.89	34.41	73.31	39.12	45.70	...	28.24	37.51	88.61	42.87	31.84	27.09	65.30	75.85	24.14	33.05
2013-02-11	44.60	14.46	78.39	68.5614	35.85	46.76	34.26	73.07	38.64	46.08	...	28.31	37.46	88.28	42.84	31.96	27.46	64.55	75.65	24.21	33.26
2013-02-12	44.62	14.27	78.60	66.8428	35.42	46.96	34.30	73.37	38.89	46.27	...	28.41	37.58	88.46	42.87	31.84	27.95	64.75	75.44	24.49	33.74
2013-02-13	44.75	14.66	78.97	66.7156	35.27	46.64	34.46	73.56	38.81	46.26	...	28.42	37.80	88.67	43.08	32.00	28.26	64.41	76.00	24.74	33.55
2013-02-14	44.58	13.99	78.84	66.6556	36.57	46.77	34.70	73.13	38.61	46.54	...	28.22	38.44	88.52	42.91	32.12	28.47	63.89	76.34	24.63	33.27

Fig. 1

The collated dataframe is resampled to fetch date indices that indicate the last date of a month. From this resampled dataframe, we select 300 random companies to perform clustering. Fig. 2 shows the resultant resampled dataframe where null values are handled using bfill method.

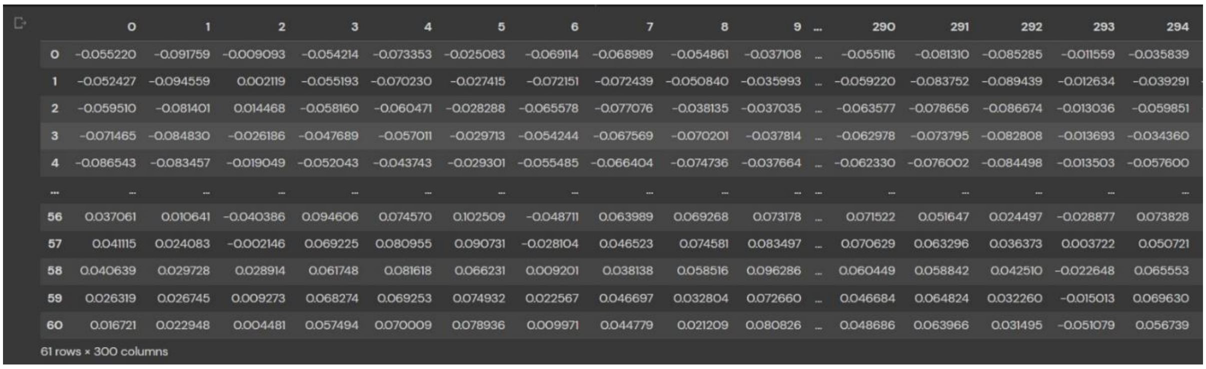


	HPE	CSRA	APA	EXR	WYN	AAL	COST	ALB	KLAC	MAA	...	BK	IDXX	LUK	REGN	KO	VZ	CCL	WLTW	ALXN	BF.B
date																					
2013-02-28	14.72	31.51	74.27	37.44	60.24	13.43	101.290	65.08	54.76	69.44	...	27.14	46.060	26.90	167.000	38.72	46.53	35.77	114.47	86.74	32.810
2013-03-31	14.72	31.51	77.16	39.27	64.48	16.97	106.110	62.52	52.74	69.06	...	27.99	46.195	27.43	176.403	40.44	49.15	34.30	114.47	92.14	35.700
2013-04-30	14.72	31.51	73.88	43.58	60.08	16.90	108.430	61.25	54.25	68.73	...	28.22	43.980	30.89	215.140	42.33	53.91	34.51	114.47	98.00	35.250
2013-05-31	14.72	31.51	82.13	41.89	58.12	17.57	109.631	66.92	56.29	67.97	...	30.06	41.220	31.38	241.880	39.99	48.48	33.10	114.47	97.58	34.410
2013-06-30	14.72	31.51	83.83	41.93	57.23	16.42	110.570	62.29	55.73	67.77	...	28.05	44.845	26.22	224.880	40.11	50.34	34.29	114.47	92.24	33.775
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2017-10-31	13.92	31.99	41.37	81.59	106.85	46.82	161.080	140.89	108.89	102.35	...	51.45	166.170	25.30	402.620	45.98	47.87	66.39	161.08	119.66	57.020
2017-11-30	13.95	28.93	41.83	85.36	112.39	50.49	184.430	134.32	102.24	102.44	...	54.74	156.410	26.31	361.860	45.77	50.89	65.64	160.80	109.81	59.800
2017-12-31	14.36	29.92	42.22	87.45	115.87	52.03	186.120	127.89	105.07	100.56	...	53.86	156.380	26.49	375.960	45.88	52.93	66.37	150.69	119.59	68.670
2018-01-31	16.40	33.28	44.87	83.48	124.13	54.32	194.870	111.59	109.80	95.37	...	56.70	187.040	27.07	366.650	47.59	54.07	71.61	160.47	119.32	69.300
2018-02-28	15.56	31.35	39.55	80.42	118.72	51.40	183.190	106.63	102.66	87.93	...	55.34	176.830	25.01	334.610	44.56	51.01	69.12	157.38	117.15	65.080

61 rows × 300 columns

Fig. 2

The resampled dataframe is scaled using standard scaler and then normalized. Scaling is a technique where we change the original range of our data into a defined range without changing the shape of our data. Normalizing involves changing our data such that, we can fit our observations in a normal distribution bell curve. Fig. 3 displays the normalized dataframe.



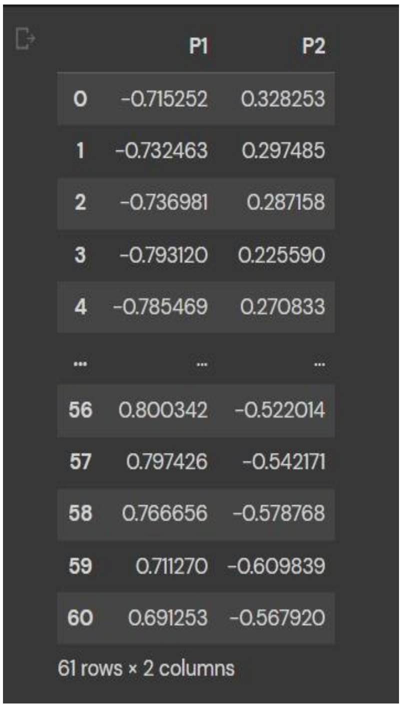
	0	1	2	3	4	5	6	7	8	9	...	290	291	292	293	294
0	-0.055220	-0.091759	-0.009093	-0.054214	-0.073353	-0.025083	-0.069114	-0.068989	-0.054861	-0.037108	...	-0.055116	-0.081310	-0.085285	-0.011559	-0.035839
1	-0.052427	-0.094559	0.002119	-0.055193	-0.070230	-0.027415	-0.072151	-0.072439	-0.050840	-0.035993	...	-0.059220	-0.083752	-0.089439	-0.012634	-0.039291
2	-0.059510	-0.081401	0.014468	-0.058160	-0.060471	-0.028288	-0.065578	-0.077076	-0.038135	-0.037035	...	-0.063577	-0.078656	-0.086674	-0.013036	-0.059851
3	-0.071465	-0.084830	-0.026186	-0.047689	-0.057011	-0.029713	-0.054244	-0.067569	-0.070201	-0.037814	...	-0.062978	-0.073795	-0.082808	-0.013693	-0.034360
4	-0.086543	-0.083457	-0.019049	-0.052043	-0.043743	-0.029301	-0.055485	-0.066404	-0.074736	-0.037664	...	-0.062330	-0.076002	-0.084498	-0.013503	-0.057600
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56	0.037061	0.010641	-0.040386	0.094606	0.074570	0.102509	-0.048711	0.063989	0.069268	0.073178	...	0.071522	0.051647	0.024497	-0.028877	0.073828
57	0.041115	0.024083	-0.002146	0.069225	0.080955	0.090731	-0.028104	0.046523	0.074581	0.083497	...	0.070629	0.063296	0.036373	0.003722	0.050721
58	0.040639	0.029728	0.028914	0.061748	0.081618	0.066231	0.009201	0.038138	0.058516	0.096286	...	0.060449	0.058842	0.042510	-0.022648	0.065553
59	0.026319	0.026745	0.009273	0.068274	0.069253	0.074932	0.022567	0.046697	0.032804	0.072660	...	0.046684	0.064824	0.032260	-0.015013	0.069630
60	0.016721	0.022948	0.004481	0.057494	0.070009	0.078936	0.009971	0.044779	0.021209	0.080826	...	0.048686	0.063966	0.031495	-0.051079	0.056739

61 rows × 300 columns

Fig. 3

3.2 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) is technique which helps us to reduce the number of dimensions. Dimensionality reduction is helpful as more dimensions can sometimes lead to underfitting. PCA is performed on the normalized data and the number of dimensions is reduced to two and store them in a dataframe. Fig. 4 shows the normalized dataframe after PCA is applied.



	P1	P2
0	-0.715252	0.328253
1	-0.732463	0.297485
2	-0.736981	0.287158
3	-0.793120	0.225590
4	-0.785469	0.270833
...	...	...
56	0.800342	-0.522014
57	0.797426	-0.542171
58	0.766656	-0.578768
59	0.711270	-0.609839
60	0.691253	-0.567920

61 rows × 2 columns

Fig. 4

3.3 CLUSTERING

Clustering is an unsupervised technique where data items that are more similar to each other are grouped. Fig. 5 shows dendrogram which is used to show the hierarchical relationship between clusters. The height of the dendrogram indicates the order in which the clusters are combined. Here, we have used hierarchical clustering analysis which aims to build hierarchy of clusters. Hierarchical Agglomerative Clustering is a method through which an object begins to form cluster upon itself and then goes on to merge with other clusters based on similarities till we reach n clusters or a single large cluster. Hierarchical Agglomerative Clustering is done for different number of clusters ranging from 2 to 7 on the dataframe that has PCA data. Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, and Fig. 11 shows the plotted results for each number of clusters (n).

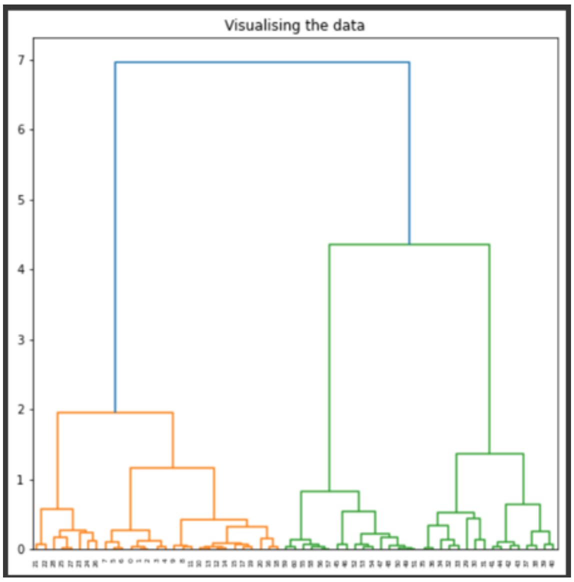


Fig. 5

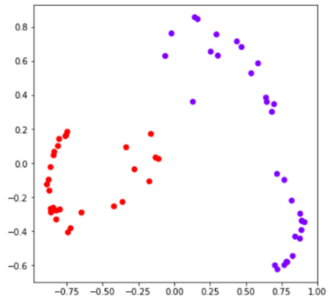


Fig. 6 (n = 2)

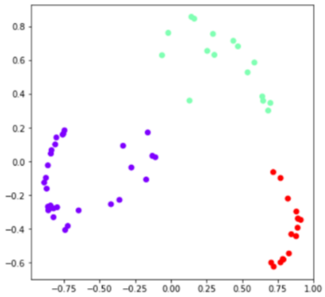


Fig. 7 (n = 3)

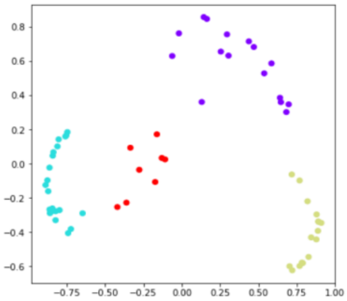


Fig. 8 (n = 4)

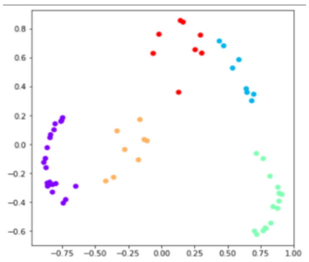


Fig. 9 (n = 5)

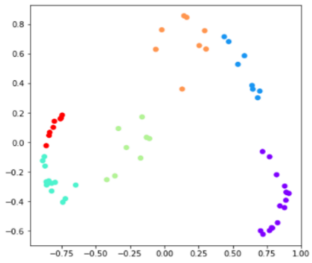


Fig. 10 (n = 6)

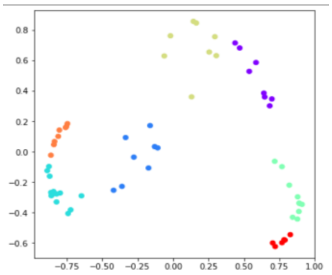


Fig. 11 (n = 7)

3.4 SILHOEUTTE SCORE:

Silhouette score or index is a metric that helps to understand the goodness of our clustering technique. Silhouette score ranges between -1 and +1. Here, silhouette score is calculated for each number of clusters and the optimal number of clusters is the one with highest silhouette score. Fig. 12 shows the number of clusters and their silhouette scores plotted in a bar graph, from which we can infer that the optimal number of clusters is 3.

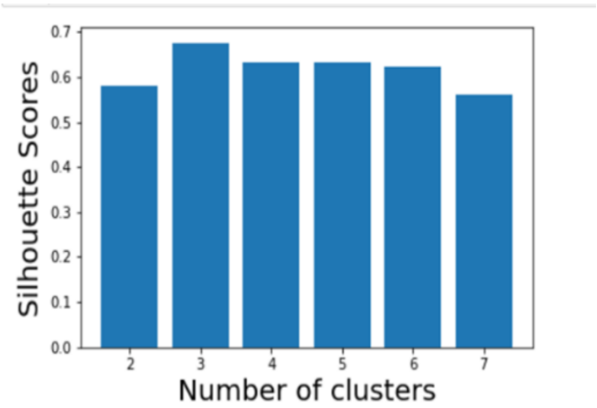


Fig. 12

3.5 LSTM PREDICTION

LSTM is an Artificial Neural Network that is extensively used in the field of prediction, Artificial Intelligence and predominantly in the field of deep learning. It is a kind of Recurring Neural Network that has both long and short term memory. The practical behavior of this type of Network is it can remember information for a longer period.

The stock details for GOOGL company are taken and loaded into a dataframe. Fig. 13 shows the graphical representation of the closing prices over a period of five years for GOOGL. After dropping all the fields which are null, the data is scaled using min-max scaler and normalized.

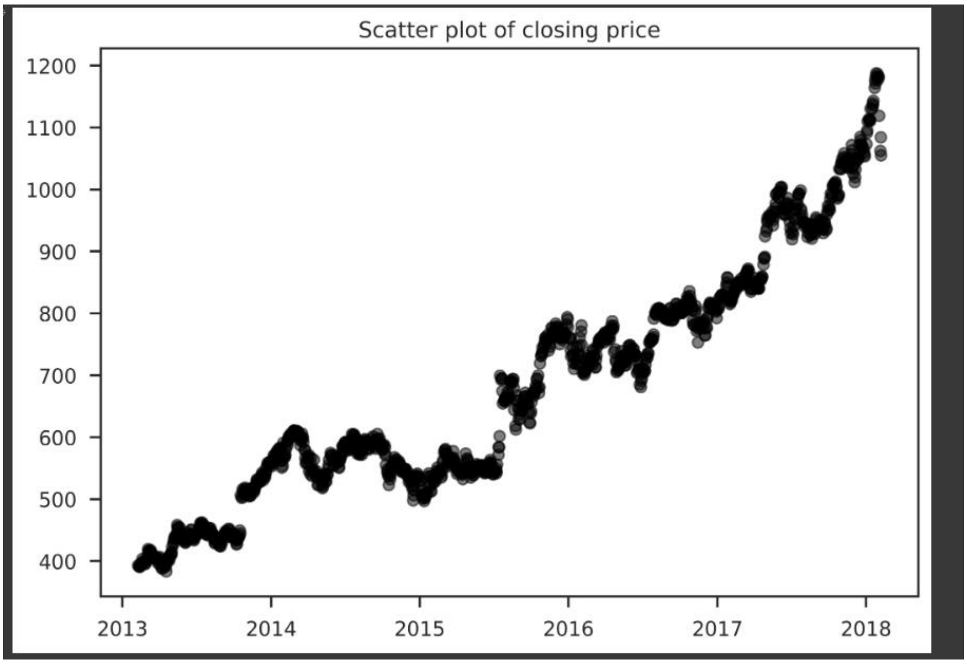
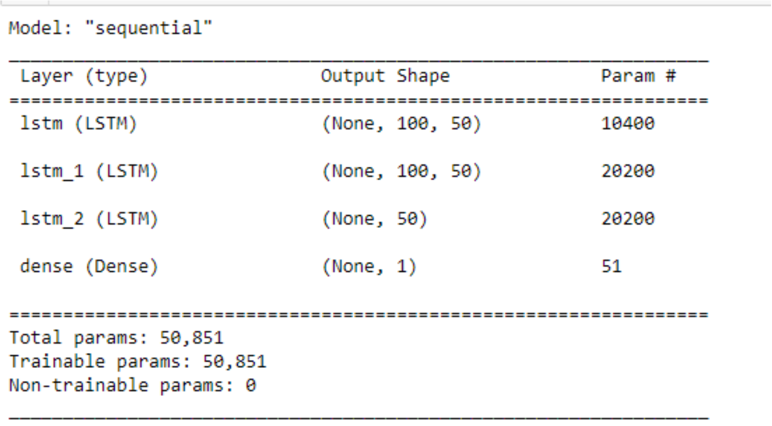


Fig. 13

A Vanilla LSTM contain basically – 3 components – an INPUT layer, a single HIDDEN layer and a standard FEEDFORWARD OUTPUT layer. A Stacked LSTM contains multiple HIDDEN layers as compared to the Vanilla one. Fig. 14 the SUMMARY of how the model is being created and the number of parameters after each LSTM layer. The process starts with converting the dataset into train and test and converting those train and test into feature and target variables. These variables are then subjected to reshaping that is to be fed into the input layer of the model. Then the model is layered properly by applying various hyper parameter values and is compiled with LOSS function as ‘mean_squared_error’ and the ‘ADAM’ optimizer. The model is then fit using the train values and is being evaluated with the testing data. Below, you can find the loss function being graphed with each iteration giving out a loss value. As we can see, the loss function is decreasing for each EPOCH value and generating an elbow function.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

Fig. 14

After predicting the appropriate values for each test values, graph is being plot for the actual values of stock and the corresponding predicted values. As perceived, from fig. 15, we can see that the predicted values are as close to the actual values in the graph.

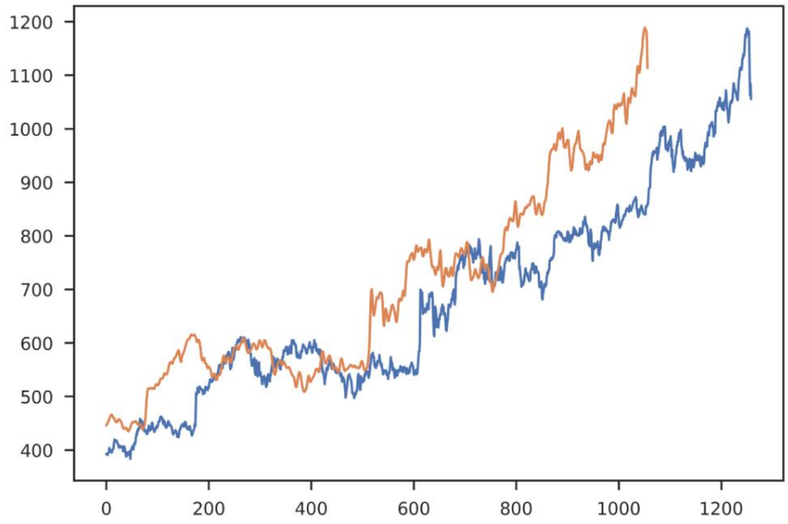


Fig. 15

To predict the future price of the stock, the time frame is being set to the future 30 days. Using the reshaped value, we are predicting the prices with the last 100 values of the stock's prices for training. The model is being set such that based on the values of those 100 values, the next 30 values can be predicted using the modelled LSTM.

From fig. 16 we can see the graph below depicting the values of the predicted values and the last 100 closing values of the GOOGL stock.

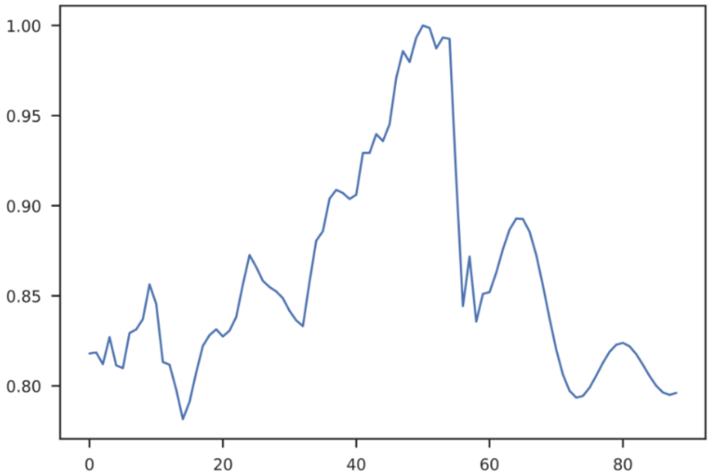


Fig. 16

The below graph (Fig.17) shows us the forecasting values of stock price in the future by predicting the values the past closing prices. The last actual price of the stock is 1055.41 and the last predicted value is 1023.6

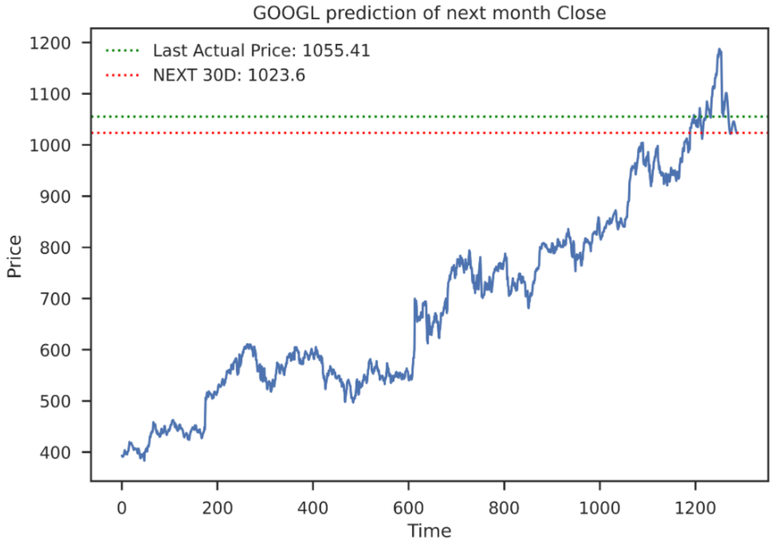


Fig. 17

4. CONCLUSION

The prediction of the closing price of GOOGL has been done and the predicted values are quite close to the actual values. Firstly, the stock prices on the S&P Index were collated and clustering techniques were applied to cluster the stocks based on their price movements. For this, various clustering techniques were considered, and the hierarchical clustering method was promising. We performed hierarchical agglomerative clustering and were able to cluster the stocks based on their movements. Then, we took the GOOGL stock price to evaluate the future stock price movements based on the LSTM method. LSTM was used because it could 'remember' the past price movements and predict future prices. Then the dataset was cleaned, pre-processed, reshaped to be fed into the model for prediction. The model was trained, and the future values were plotted based on the values given and the prediction values came out to be good enough based on the past movement of the stock.

5. REFERENCES

1. Saima Bano, M. N. A. Khan. *International Journal of Advanced Science and Technology (2018).* A Survey of Data Clustering Methods.
2. Anupriya Vysala, Dr. Joseph Gomes. Evaluating and validating cluster results (*2020).*
3. Hasim Sak, Andrew Senior, Francoise Beaufay**s,** Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling p [1-3]
4. https://www.javatpoint.com/clustering-in-machine-learning
5. https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/
6. https://www.kaggle.com/code/alexisbcook/scaling-and-normalization