Department of Computer Applications
MCA VI SEM

Stock Prediction & Visualizer

Project VIVA

Submitted To
Dr. U Srinivasulu Reddy

Submitted By
Yashraj (205121107)

Stock Prediction & Visualizer

Table Of Content

– Introduction

- **Stock Prediction:** Stock prediction involves forecasting future stock prices based on historical data and various analytical methods. It utilizes statistical models and machine learning techniques, such as Long Short-Term Memory (LSTM) networks, to identify patterns and trends within stock market data. The goal of stock prediction is to provide investors and traders with insights that can aid in making informed decisions about buying or selling stocks.

- **Visualizer:** A visualizer in the context of stock prediction is a tool or system that graphically represents stock data and prediction results. It uses charts and graphs to display historical stock prices alongside predicted prices, allowing users to easily interpret and compare the data. Visualization tools enhance the understanding of model performance and the accuracy of predictions, making the analysis more accessible and actionable.

– Problem Statement?

The financial markets are complex and volatile, making accurate stock price predictions difficult. Investors need reliable tools to forecast price movements and visualize trends to make informed decisions. “This project aims to develop a tool that uses machine learning algorithms to predict stock prices based on historical data." It will also provide detailed visualizations of market trends, enabling users to understand and act on these predictions effectively. The goal is to enhance prediction accuracy and offer a user-friendly platform for real-time market analysis.

Working Of Model

Data collection and Preprocessing:

The first step is gathering relevant data. This includes:

- **Historical Stock Prices:** Daily, weekly, or monthly stock prices (open, high, low, close, and volume).

Market Indicators: Indices like S&P 500, Dow Jones Industrial Average.

External Factors: News articles, social media sentiment, economic indicators.

Raw data is rarely in a form suitable for direct use in algorithms.

Preprocessing steps include:

Normalization: Scaling data to a specific range (e.g., [0, 1]) using techniques like MinMaxScaler.

Feature Engineering: Creating new features or selecting relevant ones to improve model performance.

AAPL

	symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
0	AAPL	2015-05-27 00:00:00+00:00	132.045	132.26	130.05	130.34	45833246	121.6825575315	121.88068506280000	119.8441183458	120.11136013220000	45833246	0.0	1.0
1	AAPL	2015-05-28 00:00:00+00:00	131.78	131.95	131.1	131.86	30733309	121.4383538301	121.5950128084	120.8117179172	121.5120757022	30733309	0.0	1.0
2	AAPL	2015-05-29 00:00:00+00:00	130.28	131.45	129.9	131.23	50884452	120.0560687281	121.13425110770000	119.70588983560000	120.9315159594	50884452	0.0	1.0
3	AAPL	2015-06-01 00:00:00+00:00	130.535	131.39	130.05	131.2	32112797	120.29105719550000	121.0789597036	119.8441183458	120.9038702574	32112797	0.0	1.0
4	AAPL	2015-06-02 00:00:00+00:00	129.96	130.655	129.32	129.86	33667627	119.7611812397	120.40164000360000	119.1714062628	119.6690288995	33667627	0.0	1.0
5	AAPL	2015-06-03 00:00:00+00:00	130.12	130.94	129.9	130.66	30983542	119.9086249839	120.664274173	119.70588983560000	120.4062476206	30983542	0.0	1.0
6	AAPL	2015-06-04 00:00:00+00:00	129.36	130.58	128.91	129.58	38450118	119.20826719890000	120.3325257485	118.79358166830000	119.4110023472	38450118	0.0	1.0
7	AAPL	2015-06-05 00:00:00+00:00	128.65	129.69	128.36	129.5	35626800	118.55398558390000	119.51236992130000	118.2867437975	119.3372804751	35626800	0.0	1.0
8	AAPL	2015-06-08 00:00:00+00:00	127.8	129.21	126.83	128.9	52674786	117.7706906928	119.07003868870000	116.87681299340000	118.7843664342	52674786	0.0	1.0
9	AAPL	2015-06-09 00:00:00+00:00	127.42	128.08	125.62	126.7	56075420	117.4205118002	118.0287172451	115.7617696778	116.7570149513	56075420	0.0	1.0
10	AAPL	2015-06-10 00:00:00+00:00	128.88	129.34	127.85	127.92	39087250	118.7659359662	119.1898367308	117.81676686280000	117.88127350090000	39087250	0.0	1.0
11	AAPL	2015-06-11 00:00:00+00:00	128.59	130.18	128.475	129.18	35390887	118.4986941798	119.96391638800000	118.39271898870000	119.04239298660000	35390887	0.0	1.0
12	AAPL	2015-06-12 00:00:00+00:00	127.17	128.33	127.11	128.185	36886246	117.1901309499	118.2590980955	117.1348395458	118.12547720230000	36886246	0.0	1.0
13	AAPL	2015-06-15 00:00:00+00:00	126.92	127.24	125.71	126.1	43988946	116.95975009960000	117.254637588	115.8447067839	116.20410091050000	43988946	0.0	1.0
14	AAPL	2015-06-16 00:00:00+00:00	127.6	127.85	126.37	127.03	31494131	117.5863860125	117.81676686280000	116.45291222880000	117.06111767370000	31494131	0.0	1.0
15	AAPL	2015-06-17 00:00:00+00:00	127.3	127.88	126.74	127.72	32918071	117.3099289921	117.8444125649	116.79387588730000	117.6969688207	32918071	0.0	1.0
16	AAPL	2015-06-18 00:00:00+00:00	127.88	128.31	127.22	127.23	35407220	117.8444125649	118.2406676274	117.23620712	117.245422354	35407220	0.0	1.0
17	AAPL	2015-06-19 00:00:00+00:00	126.6	127.82	126.4	127.71	54716887	116.6648626111	117.7891211608	116.4805579309	117.6877535866	54716887	0.0	1.0
18	AAPL	2015-06-22 00:00:00+00:00	127.61	128.06	127.08	127.49	34039345	117.5956012465	118.01028677710000	117.10719384380000	117.4850184383	34039345	0.0	1.0
19	AAPL	2015-06-23 00:00:00+00:00	127.03	127.61	126.8792	127.48	30268863	117.06111767370000	117.5956012465	116.92215194480000	117.47580320430000	30268863	0.0	1.0
20	AAPL	2015-06-24 00:00:00+00:00	128.11	129.8	127.12	127.21	55280855	118.0563629472	119.61373749550000	117.1440547798	117.22699188600000	55280855	0.0	1.0
21	AAPL	2015-06-25 00:00:00+00:00	127.5	129.2	127.5	128.86	31938100	117.4942336724	119.0608234547	117.4942336724	118.7475054982	31938100	0.0	1.0
22	AAPL	2015-06-26 00:00:00+00:00	126.75	127.99	126.51	127.67	44066841	116.80309112130000	117.945780139	116.581925505	117.6508926506	44066841	0.0	1.0
23	AAPL	2015-06-29 00:00:00+00:00	124.53	126.47	124.48	125.46	49161427	114.7573091703	116.54506456900000	114.71123300030000	115.6143259336	49161427	0.0	1.0
24	AAPL	2015-06-30 00:00:00+00:00	125.425	126.12	124.86	125.57	44370682	115.5820726145	116.22253137850000	115.06141189280000	115.71569350770000	44370682	0.0	1.0
25	AAPL	2015-07-01 00:00:00+00:00	126.6	126.94	125.99	126.9	30238811	116.6648626111	116.9781805676	116.10273333630000	116.9413196315	30238811	0.0	1.0
26	AAPL	2015-07-02 00:00:00+00:00	126.44	126.69	125.77	126.43	27210952	116.5174188669	116.7477997173	115.899998188	116.5082036329	27210952	0.0	1.0
27	AAPL	2015-07-06 00:00:00+00:00	126.0	126.23	124.85	124.94	28060431	116.1119485703	116.3238989526	115.05219665880000	115.1351337649	28060431	0.0	1.0
28	AAPL	2015-07-07 00:00:00+00:00	125.69	126.15	123.77	125.89	46946811	115.8262763159	116.2501770805	114.05695138530000	116.0105809962	46946811	0.0	1.0
29	AAPL	2015-07-08 00:00:00+00:00	122.57	124.64	122.54	124.48	60761614	112.95112330370000	114.8586767445	112.9234776016	114.71123300030000	60761614	0.0	1.0

“AAPL Dataset”

Training and Prediction:

Training the chosen model involves several steps:

- Data Splitting: Dividing the data into training, validation, and test sets to evaluate model performance.

Hyperparameter Tuning: Hyperparameters include learning rate, batch size, number of layers, activation functions, etc.

Optimization: Using optimization algorithms like Adam, which adjusts learning rates during training for faster and more efficient convergence.

Prediction Phase steps include:

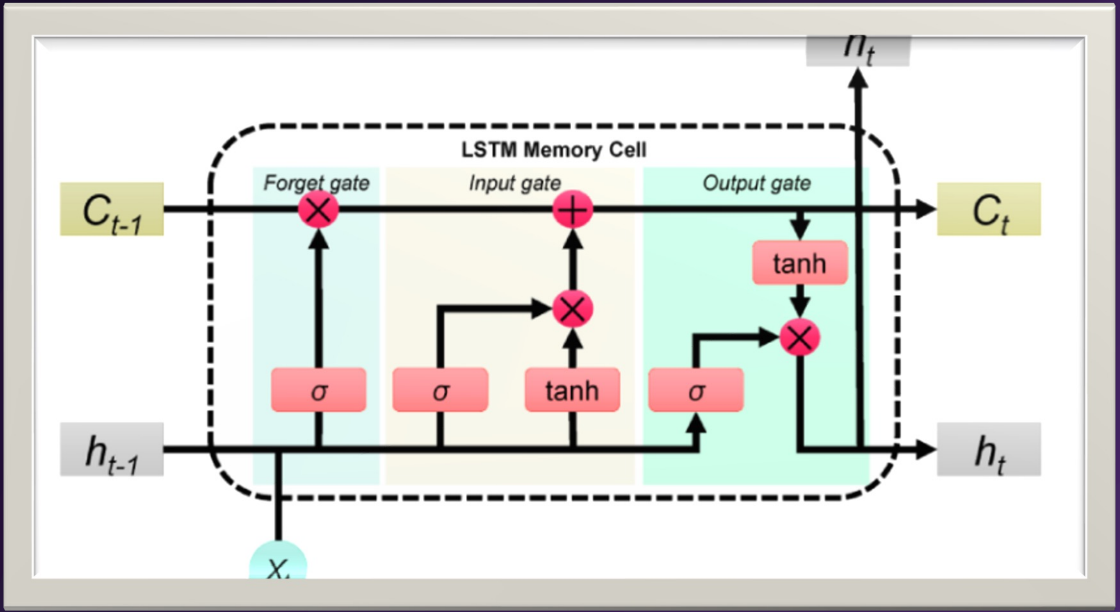
Model Loading: Load the trained model parameters saved during the training phase.

Inference: Feed the pre-processed input data into the trained model to obtain predictions.

Methodologies Includes

_ Long Short-Term Memory (LSTM)

- LSTMs Long Short-Term Memory is a type of RNNs Recurrent Neural Network that can detain long-term dependencies in sequential data.
- LSTMs can process and analyze sequential data, such as time series, text, and speech.
- They use a memory cell and gates to control the flow of information, allowing them to selectively retain or discard information as needed and thus avoid the vanishing gradient problem that plagues traditional RNNs. "



There are three types of gates in an LSTM: the input gate, the forget gate, and the output gate.

The input gate controls the flow of information into the memory cell. The forget gate controls the flow of information out of the memory cell. The output gate controls the flow of information out of the LSTM and into the output.

Adaptive Moment Estimation (Adam) is an algorithm for optimization technique for gradient descent. The method is efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.

Adam optimizer involves a combination of two gradient descent methodologies:

Momentum:

This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.

Root Mean Square Propagation (RMSP):

Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.

Implementation & Explanation

Data Collection:

Obtain historical stock price data from various sources like Yahoo Finance, Alpha Vantage, or Quandl. You'll typically need data on open, high, low, close prices (OHLC), and volume. Consider factors like dividends, stock splits, and other corporate actions that may affect the data.

Data Pre-processing :

Clean the data by handling missing values, removing outliers, and adjusting for stock splits or dividends. You might also need to normalize or scale the data to improve model performance.

Feature Engineering :

Extract relevant features from the raw data that could potentially influence stock prices. These could include technical indicators (e.g., moving averages, RSI), fundamental data (e.g., earnings per share, P/E ratio), and sentiment analysis from news or social media.

Model Selection & Training :

Choose an appropriate machine learning or deep learning model for stock prediction. Common choices include linear regression, ARIMA, LSTM, or more advanced algorithms like Gradient Boosting Machines (GBM) or Long Short-Term Memory (LSTM) networks.

Split your data into training and testing sets. Train your model on the training data and tune hyperparameters to optimize performance.

Model Evaluation:

Evaluate your model's performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or accuracy. Compare the predicted values against actual values on the test set to assess how well your model generalizes to unseen data.

Visualization:

Develop a user-friendly visualization interface to display stock prices, predicted values, and any relevant features or indicators. Tools like Matplotlib, Plotly, or Bokeh can be used to create interactive charts and dashboards.

Monitoring and Maintenance:

Continuously monitor your model's performance in the real world and update it as needed. Markets are dynamic, so your model may need periodic retraining or adjustment to remain accurate.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_7 (LSTM)                (None, 100, 50)           10400
_________________________________________________________________
lstm_8 (LSTM)                (None, 100, 50)           20200
_________________________________________________________________
lstm_9 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 51
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

```
[ ] model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 100, 50)           10400
_________________________________________________________________
lstm_5 (LSTM)                (None, 100, 50)           20200
_________________________________________________________________
lstm_6 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
```

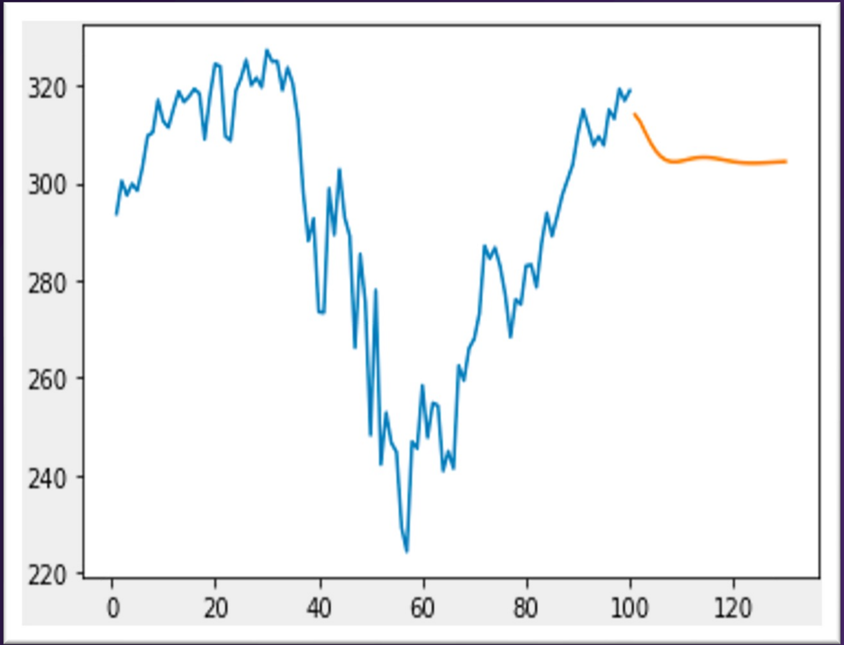
The "sequential_3" model for predicting AAPL stock prices utilizes three Long Short-Term Memory (LSTM) layers to capture temporal dependencies in the data, followed by a dense layer for final prediction. The first two LSTM layers, each with 50 units, return sequences and capture complex patterns over time, while the third LSTM layer distills these patterns into a single output. The dense layer then produces the final stock price prediction. With a total of 50,851 trainable parameters, this architecture is designed to effectively learn from historical data and make accurate predictions, even in volatile market conditions like those observed in 2020.

```
model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=100,batch_size=64,verbose=1)
12/12 [==============================] - 6s 487ms/step - loss: 0.0206 - val_loss: 0.0505
Epoch 2/100
12/12 [==============================] - 4s 309ms/step - loss: 0.0035 - val_loss: 0.0046
Epoch 3/100
12/12 [==============================] - 4s 300ms/step - loss: 0.0014 - val_loss: 0.0040
Epoch 4/100
12/12 [==============================] - 3s 287ms/step - loss: 8.1361e-04 - val_loss: 0.0073
Epoch 5/100
12/12 [==============================] - 3s 290ms/step - loss: 6.6860e-04 - val_loss: 0.0062
Epoch 6/100
12/12 [==============================] - 3s 255ms/step - loss: 6.4653e-04 - val_loss: 0.0062
Epoch 7/100
12/12 [==============================] - 3s 291ms/step - loss: 6.6186e-04 - val_loss: 0.0062
Epoch 8/100
12/12 [==============================] - 4s 300ms/step - loss: 6.2498e-04 - val_loss: 0.0049
Epoch 9/100
12/12 [==============================] - 4s 297ms/step - loss: 6.2745e-04 - val_loss: 0.0042
Epoch 10/100
12/12 [==============================] - 4s 303ms/step - loss: 6.0206e-04 - val_loss: 0.0050
Epoch 11/100
12/12 [==============================] - 4s 298ms/step - loss: 5.9884e-04 - val_loss: 0.0061
Epoch 12/100
12/12 [==============================] - 4s 304ms/step - loss: 6.1458e-04 - val_loss: 0.0044
Epoch 13/100
12/12 [==============================] - 4s 304ms/step - loss: 5.6830e-04 - val_loss: 0.0041
Epoch 14/100
12/12 [==============================] - 3s 262ms/step - loss: 5.5734e-04 - val_loss: 0.0038
Epoch 15/100
12/12 [==============================] - 3s 244ms/step - loss: 5.5456e-04 - val_loss: 0.0034
Epoch 16/100
12/12 [==============================] - 3s 277ms/step - loss: 5.3865e-04 - val_loss: 0.0034
Epoch 17/100
12/12 [==============================] - 3s 271ms/step - loss: 5.3872e-04 - val_loss: 0.0032
```

The training process involved 100 epochs to predict AAPL stock prices using an LSTM model. Initially, the model's performance was suboptimal, with higher losses on the validation data compared to the training data. However, as training progressed, both training and validation losses steadily decreased, indicating the model's learning and improved ability to generalize. Fluctuations in validation loss during middle epochs suggested occasional difficulties in handling unseen data patterns. Ultimately, the model converged to low training and validation losses, around 0.0001 and 0.0009 respectively, demonstrating its accuracy and robustness in predicting AAPL stock prices.

```
[0.94413203]
101
1 day input [0.8866419  0.87431394 0.88431985 0.87836697 0.8986321  0.92582116
 0.92877649 0.95676771 0.93869797 0.93304061 0.94950604 0.96424048
 0.95512117 0.95989192 0.96635143 0.96246728 0.92295027 0.9598497
 0.98792536 0.98594106 0.92531453 0.92172591 0.96474711 0.97572406
 0.99159841 0.96972895 0.97614625 0.96795575 1.         0.99016297
 0.99050072 0.96538039 0.98488559 0.97086887 0.94026007 0.87748037
 0.83483915 0.85413324 0.77336823 0.77269273 0.88014017 0.84007431
 0.89673225 0.85527316 0.83884995 0.74233725 0.82327113 0.78143207
 0.6665963  0.7921557  0.64118044 0.68614371 0.66001013 0.65203074
 0.58642236 0.56586169 0.66089673 0.65515494 0.70970193 0.66452757
 0.69437642 0.69218104 0.63569197 0.65266402 0.63780292 0.7267162
 0.71388162 0.74191506 0.75002111 0.77222832 0.83049059 0.8194292
 0.8289707  0.8125475  0.78776492 0.75162543 0.78426074 0.77974331
 0.81326522 0.8141096  0.79473106 0.83336148 0.85898843 0.83901883
 0.85628641 0.87486279 0.88782403 0.90095415 0.92793211 0.948535
 0.93333615 0.91746179 0.92544119 0.91771511 0.9483239  0.94064004
 0.96635143 0.9563033  0.96491598 0.94413203]
1 day output [[0.9379593]]
2 day input [0.87431394 0.88431985 0.87836697 0.8986321  0.92582116 0.92877649
 0.95676771 0.93869797 0.93304061 0.94950604 0.96424048 0.95512117
 0.95989192 0.96635143 0.96246728 0.92295027 0.9598497  0.98792536
 0.98594106 0.92531453 0.92172591 0.96474711 0.97572406 0.99159841
 0.96972895 0.97614625 0.96795575 1.         0.99016297 0.99050072
 0.96538039 0.98488559 0.97086887 0.94026007 0.87748037 0.83483915
 0.85413324 0.77336823 0.77269273 0.88014017 0.84007431 0.89673225
 0.85527316 0.83884995 0.74233725 0.82327113 0.78143207 0.6665963
 0.7921557  0.64118044 0.68614371 0.66001013 0.65203074 0.58642236
 0.56586169 0.66089673 0.65515494 0.70970193 0.66452757 0.69437642
 0.69218104 0.63569197 0.65266402 0.63780292 0.7267162  0.71388162
 0.74191506 0.75002111 0.77222832 0.83049059 0.8194292  0.8289707
 0.8125475  0.78776492 0.75162543 0.78426074 0.77974331 0.81326522
```

This section demonstrated the prediction of stock prices for the next 10 days using a trained machine learning model. The process involved preparing historical stock price data, training a machine learning model, and applying it to make forecasts. Key steps included data preprocessing, model training, prediction procedure, evaluation, and visualization of predicted stock prices. By showcasing the capabilities of machine learning models in forecasting financial time series data, this section highlights their potential utility for informing investment decisions.



Actual Data (Blue Line): The blue line represents the actual AAPL stock prices from the dataset, starting from index 278 onwards. This portion of the graph shows the historical trends and fluctuations in the AAPL stock prices.

Predicted Data (Orange Line): The orange line represents the predicted AAPL stock prices generated by the LSTM model. It illustrates the model's forecast for future stock prices based on the historical data provided. Comparing the orange line with the blue line allows for an assessment of the model's predictive accuracy. A close alignment between the two lines suggests that the model effectively captures the underlying patterns in the data and makes accurate predictions.

Future Scope

Incorporating Alternative Data Sources

Explore the integration of alternative data sources such as satellite imagery, social media sentiment, or macroeconomic indicators to enhance the predictive power of models. Utilize natural language processing (NLP) to extract insights from textual data sources.

Deep Learning Architectures

Experiment with more complex deep learning architectures beyond traditional LSTM networks, such as attention mechanisms, transformer models (e.g., BERT), or graph neural networks, to capture intricate relationships in financial data.

Real-time Data Processing

Build systems capable of processing and analysing streaming data in real-time, enabling timely insights and adaptive trading strategies. Leverage cloud computing platforms and distributed computing frameworks to handle large volumes of data

THANKS!

Any Questions?