CLOSE PRICE PREDICTION REPORT

12th MARCH 2024

Prepared By:

PRIYANSHU SINGH
PRATEEK RAJPUT
YASH KHANDELWAL

PROJECT OVERVIEW :

The stock market is a dynamic and complex financial ecosystem where investors seek to predict price movements for strategic decision-making. This project focuses on forecasting the closing prices of stocks, a crucial metric for traders and investors.
The primary goal of this project is to develop a predictive model that can accurately forecast the closing prices of various companies.

PROBLEM STATEMENT :

The stock market is a complex and ever-changing ecosystem where investors buy and sell shares of publicly traded companies. Numerous factors influence stock prices, including company performance, economic indicators, geopolitical events, and market sentiment. Understanding these intricate relationships is key to making informed investment decisions.
Predicting stock prices requires a comprehensive analysis of various data points, such as historical stock prices, financial statements, market trends, and external factors impacting the business.
Participants will be provided with historical stock market data, including price trends, trading volumes, and relevant financial indicators. The dataset covers a diverse range of stocks from NSE, spanning different sectors and markets. Your task is to analyse this data, identify meaningful patterns, and create a predictive model that can anticipate future closing prices.

APROACH & ALGORITHMS :

Stock price prediction is a complex task that involves leveraging historical market data to forecast future stock prices. The approach typically follows several key steps. First, data acquisition involves gathering historical stock price data, which may include daily, hourly, or minute-by-minute observations, along with relevant financial indicators. Next, data exploration and preprocessing are crucial for understanding the dataset's characteristics, handling missing values, and addressing potential outliers. Feature engineering plays a vital role, involving the creation of additional relevant features, such as moving averages, technical indicators, or sentiment scores, to enhance the model's predictive power. Various forecasting models can be employed, ranging from traditional statistical methods like Autoregressive Integrated Moving Average (ARIMA) or Seasonal-Trend decomposition using LOESS (STL) to more advanced machine learning techniques such as Long Short-Term Memory (LSTM) networks or gradient boosting algorithms. The model is trained on a historical dataset, with hyperparameter tuning and validation performed to optimize its performance. Evaluation metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE) assess the model's accuracy. Continuous refinement, periodic retraining, and adaptation to changing market conditions are essential aspects of a successful stock price prediction approach. Additionally, considering the inherently unpredictable and volatile nature of financial markets, it's crucial to acknowledge the limitations and uncertainties associated with any prediction model.

In Stock price prediction, we are going to use LONG SHORT-TERM MEMORY (LSTM) model for the prediction of future closing prices. The LSTM model excels at handling the temporal nature of stock prices. The architecture of an LSTM includes memory cells that store information over longer periods, enabling the model to capture trends and dependencies that simpler models might miss. LSTM models offer a powerful tool for capturing complex patterns in stock price data, but the inherent uncertainty of financial markets requires a cautious interpretation of results.

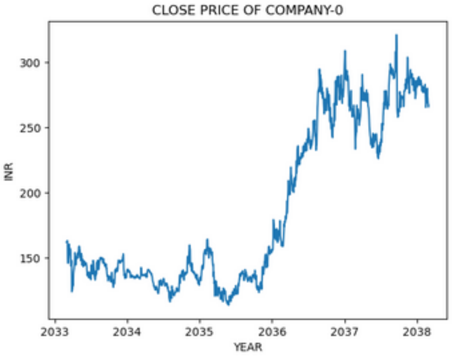
In the given dataset we are taking open price, high price, low price, adj close and volume as input to the model and the model gives the closing price of the company.

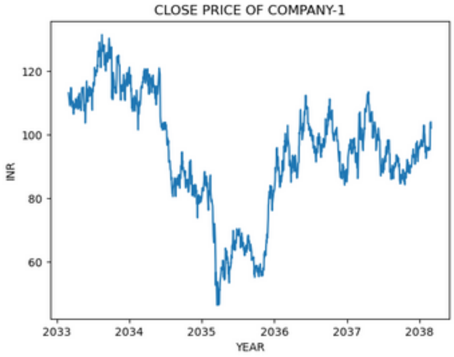
We are predicting the closing price of each company individually by converting the dataset from multivariate to univariate.

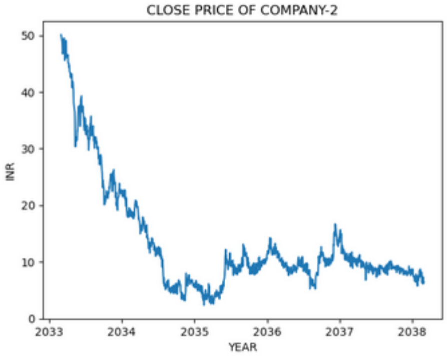
DATA PRE-POCESSING

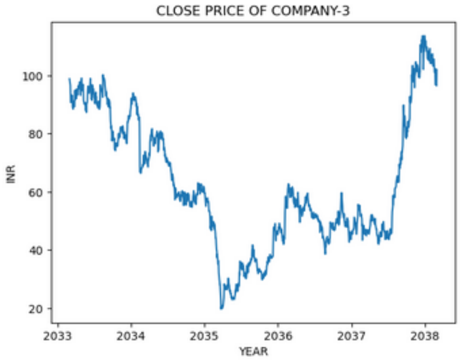
- **DATA ANALYSIS :**

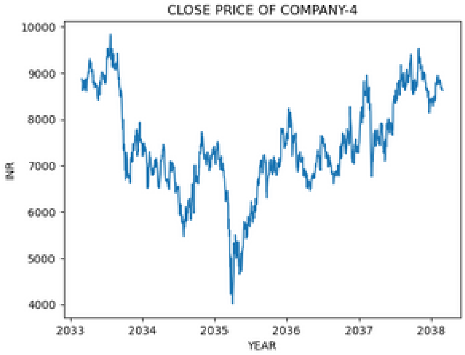
The dataset has stock prices of 6 companies. For each company there is value of open price , close price ,high price , low price, adjacent close and volume from 2033 to 2038 february. Firstly, we analyze the closing price of each company for the particular data. The graph are below:

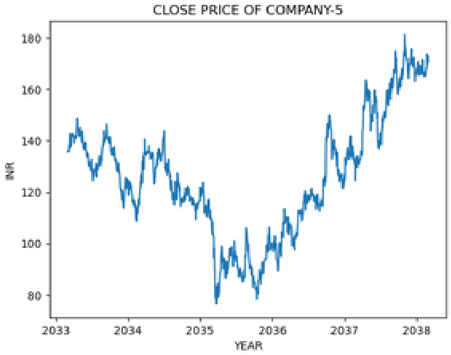












The dataset has no null values. After Pivoting each company has 1236 counts and 24 features.

- In the fig. 1 (closing price of company-0) , the company-0 has average value equals to 187.33 , its max value is 321.06 and its min value is 113.64
- In the fig. 2 (closing price of company-1) , the company-0 has average value equals to 93.931299 its max value is 131.375 and its min value is 46.36
- In the fig. 3 (closing price of company-2) , the company-0 has average value equals to 13.35, its max value is 50.09 and its min value is 2.35
- In the fig. 4 (closing price of company-3) , the company-0 has average value equals to 62.44, its max value is 113.67 and its min value is 19.69
- In the fig. 5(closing price of company-4) , the company-0 has average value equals to 7444.98, its max value is 9832.11 and its min value is 4011.67
- In the fig. 6 (closing price of company-5) , the company-0 has average value equals to 125.69, its max value is 181.28 and its min value is 76.54

- ## SCALING :

Each company has closing price of different scales so it is not good to give this data to model for training and prediction . The predictions can’t be trusted based upon this so we need to scale the given data.
Scaling is a process in data preprocessing that is commonly applied to features in a dataset. The goal of scaling is to transform the values of different features to a similar scale. This is particularly important for machine learning algorithms that are sensitive to the magnitude of the input features. There are two common methods for scaling are Min-Max scaling and Standardization (Z-score normalization), but in our code we have used Min-Max scaling Method:

- Formula: X_scaled= (X−X_min)/(X_max−X_min)
- This method scales the data to a specific range, usually [0, 1].

Scaling is important in machine learning because it can help algorithms converge faster, avoid numerical instability issues, and ensure that no particular feature dominates the learning process due to its scale. The choice between Min-Max scaling and Standardization depends on the characteristics of data and the requirements of the machine learning algorithm .

- ## TRAIN-TEST-SPLITTING

The train-test split is a fundamental step in machine learning model development that involves partitioning a dataset into two distinct subsets: the training set and the testing set. The primary purpose of this split is to evaluate how well a trained model generalizes to new, unseen data. The training set is used to train the model by exposing it to labeled examples, allowing it to learn the underlying patterns in the data. Meanwhile, the testing set, kept separate from the training process, serves as a benchmark to assess the model's performance on data it has not encountered during training. Typically, a common split ratio is 80-20 or 70-30, where the larger portion constitutes the training set. Striking the right balance between the training and testing sets is crucial for developing a model that can make accurate predictions on new, real-world data. This practice helps detect overfitting, where a model memorizes the training data but fails to generalize well to unseen examples. In summary, the train-test split is a vital practice in machine learning to ensure the robustness and reliability of a trained model when deployed in practical scenarios.

MODEL & IT’S TRAINING :

Model Implementation :

- A Sequential model is created to build the neural network layer by layer.
- The first LSTM layer has 128 units, uses the rectified linear unit (‘relu’) activation function, and expects input data with a shape of **(time_step, 6)**. This shape indicates that the model considers a sequence of **time_step** time points, each with 6 features.
- The second LSTM layer has 64 units, uses the hyperbolic tangent (tanh) activation function, and does not return sequences (only the output of the last time step).
- A Dropout layer is added to introduce regularization, randomly setting 20% of input units to zero during training to prevent overfitting.
- The final layer is a fully connected (Dense) layer with a single unit, representing the output for the regression task (predicting a continuous value).
- The model is compiled using the Adam optimizer. The Mean Squared Error (MSE) loss function is chosen, which is suitable for regression , where the goal is to minimize the squared difference between predicted and actual values.
- In the model we have used early stopping which prevents the model from training for too long and potentially overfitting the training data. If the validation loss stops decreasing or improving, the training process halts, and the model with the best performance on the validation set is retained.

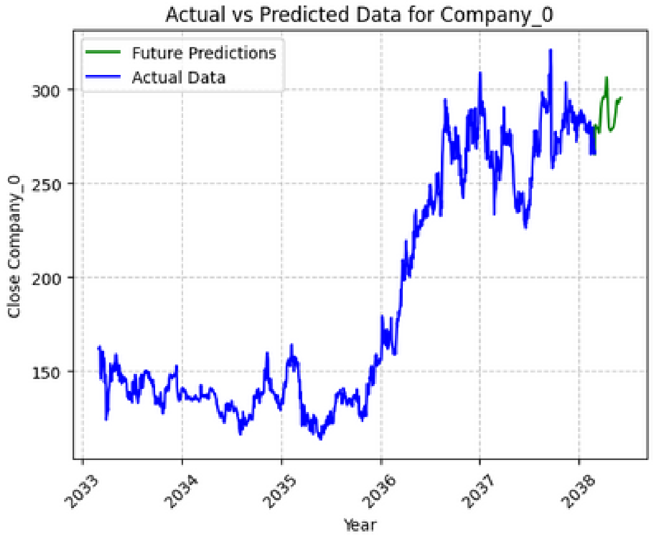
Training:

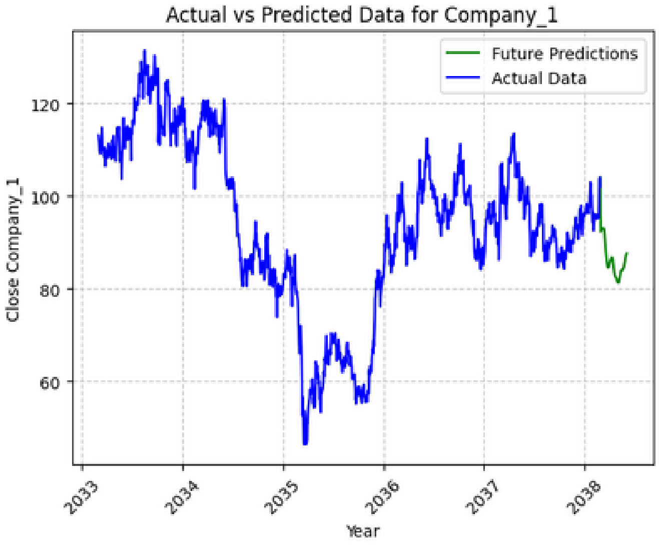
- **X_train** and **y_train** are the training data (input features and target labels, respectively).
- validation_data=(X_test, y_test): The validation data is provided to monitor the model's performance on unseen data during training. This helps prevent overfitting.
- The number of training epochs is set to 100, the model goes through the entire training dataset 100 times.
- Now, we fit the X_train and y_train to the model for its training.

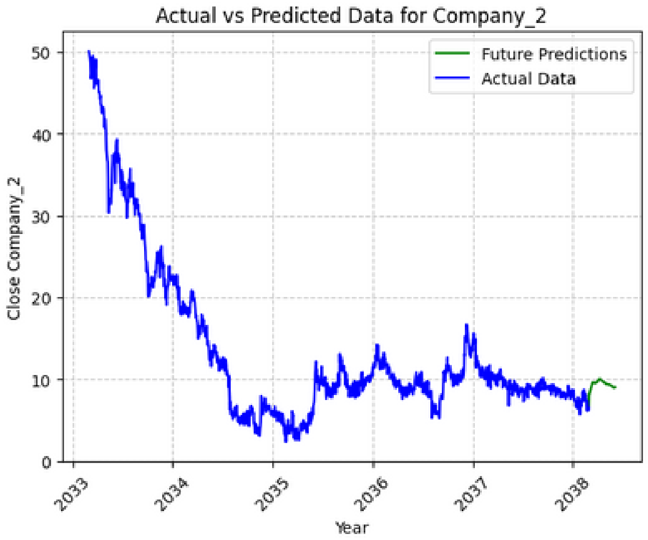
After training the model we predict the values for the test data and calculate mean squared error and root mean squared error for analysis of model. If model is not so accurate then we do hyperparameter tuning of the model by changing values of learning rate, epochs, layers , etc. . At particular value of these parameters we get the highest accuracy of our model

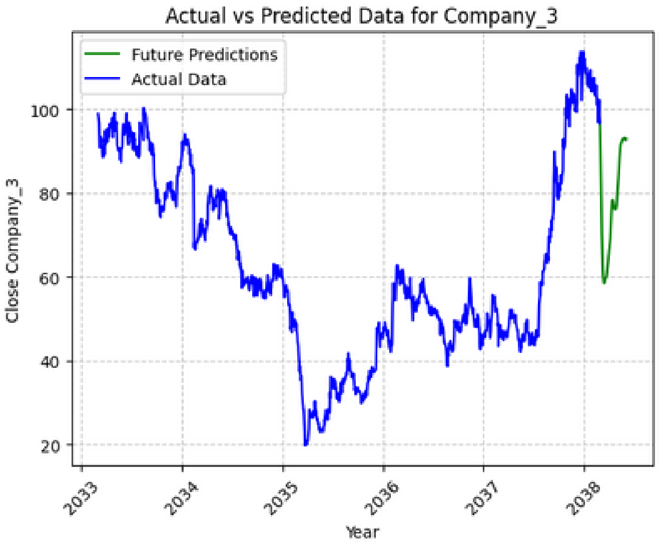
RESULT & DISCUSSION :

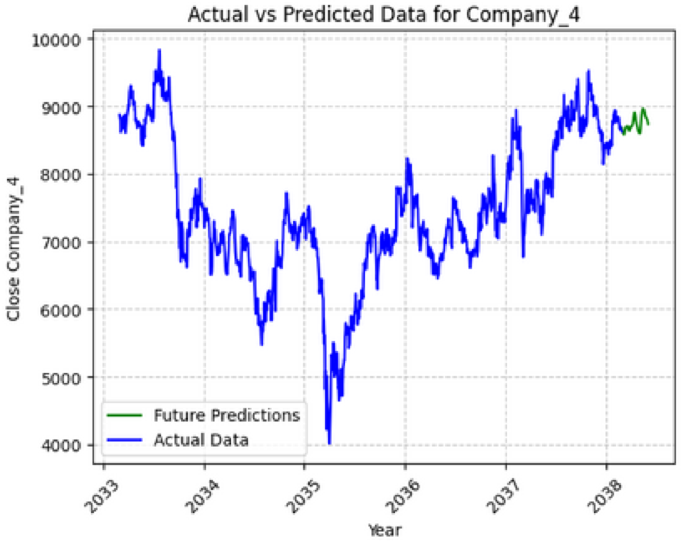
After training the model we have to forecast the closing price of each company by using the previous data for the next 96 days. The Forecast values are represented in the graph:

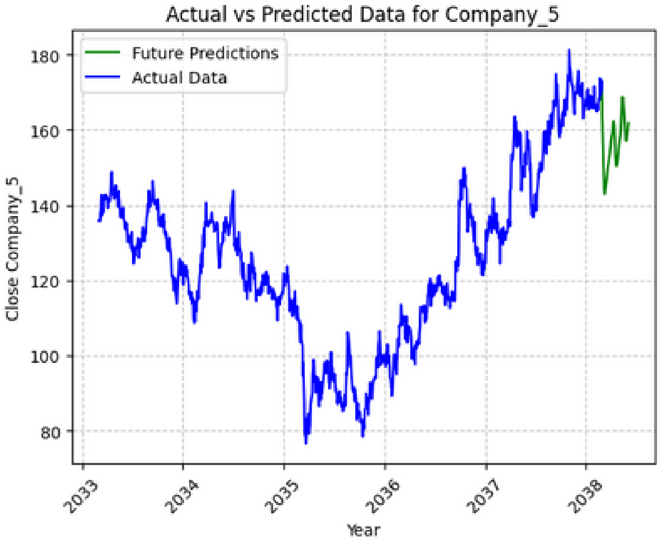












Here, timestep is 150 and the next 96 values are the forecast values of the model.. The forecast values has some root mean squared errors which can be improved by various techniques.

IMPROVEMENTS:

- Adding more LSTM layers or increasing the number of units may capture more complex patterns, but be cautious of overfitting.
- Use bidirectional LSTMs to consider information from both past and future time steps.
- Increase or decrease dropout rates to control overfitting. Dropout can prevent the model from relying too heavily on specific neurons during training.
- Use visualizations such as attention mechanisms to understand which parts of the input sequence the model focuses on during prediction.
- For specific applications, consider using stateful LSTM models to carry the hidden state from one batch to the next, preserving information between batches.
- Fine-tune the patience parameter in the EarlyStopping callback. This parameter determines how many epochs to wait for improvement before stopping the training process.
- Systematically explore different hyperparameter combinations, such as the number of LSTM units, batch size, and sequence length, to find the optimal configuration.

REFERENCES:

- https://www.deeplearningbook.org/
- https://www.researchgate.net/publication/335975993_Understanding_LSTM_--_a_tutorial_into_Long_Short-Term_Memory_Recurrent_Neural_Networks
- https://www.tensorflow.org/about/bib
- https://www.researchgate.net/publication/360351817_Stock_Market_Prediction_Using_LSTM