

# NIFTY 50 Stock Price Prediction & Pairs Trading Classification

Aditya Moza 2022035

Manit Kaushik 2022277

## 1 Introduction

The objective of our project is to predict the stock prices of companies listed in the NIFTY 50 index using machine learning techniques. This project is divided into three main sections: stock price prediction, stock classification based on growth, and classification of profitable pairs for pairs trading. The dataset used for our project is [NIFTY-50 Stock Market Data \(2000 - 2021\)](#).

## 2 Work Done So Far

For our first submission, we have completed the first two sections, stock price prediction and stock classification based on growth. Both of these tasks included using 2 famous Machine Learning techniques and comparison of their results. Currently, have been hard-coded to use stock's of a single company (TCS), not of all 50 stocks in NIFTY 50. This is the data table for TCS Stock used currently:

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable	Volume	%Deliverble
0	2004-08-25	TCS	EQ	850.00	1198.70	1198.70	979.00	985.00	987.95	1008.32	17116372	1.725876e+15	NaN		5206360	0.3042
1	2004-08-26	TCS	EQ	987.95	992.00	997.00	975.30	976.85	979.00	985.65	5055400	4.982865e+14	NaN		1294899	0.2561
2	2004-08-27	TCS	EQ	979.00	982.40	982.40	958.55	961.20	962.65	969.94	3830750	3.715586e+14	NaN		976527	0.2549
3	2004-08-30	TCS	EQ	962.65	969.90	990.00	965.00	986.40	986.75	982.65	3058151	3.005106e+14	NaN		701664	0.2294
4	2004-08-31	TCS	EQ	986.75	986.50	990.00	976.00	987.80	988.10	982.18	2649332	2.602133e+14	NaN		695234	0.2624
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4134	2021-04-26	TCS	EQ	3109.50	3102.05	3153.00	3088.05	3100.05	3100.80	3122.35	2490260	7.775465e+14	87012.0		1345353	0.5402
4135	2021-04-27	TCS	EQ	3100.80	3108.00	3136.10	3103.00	3132.75	3132.00	3122.18	1471417	4.594036e+14	97077.0		849767	0.5775
4136	2021-04-28	TCS	EQ	3132.00	3149.95	3151.00	3112.00	3123.00	3124.10	3125.72	1639037	5.123176e+14	76017.0		847762	0.5172
4137	2021-04-29	TCS	EQ	3124.10	3145.60	3157.35	3105.50	3114.45	3115.25	3130.83	1621395	5.076306e+14	98275.0		837567	0.5166
4138	2021-04-30	TCS	EQ	3115.25	3099.00	3132.05	3020.00	3040.00	3035.65	3063.19	3072305	9.411057e+14	157829.0		1942473	0.6323

4139 rows x 15 columns

## 3 Stock Price Prediction

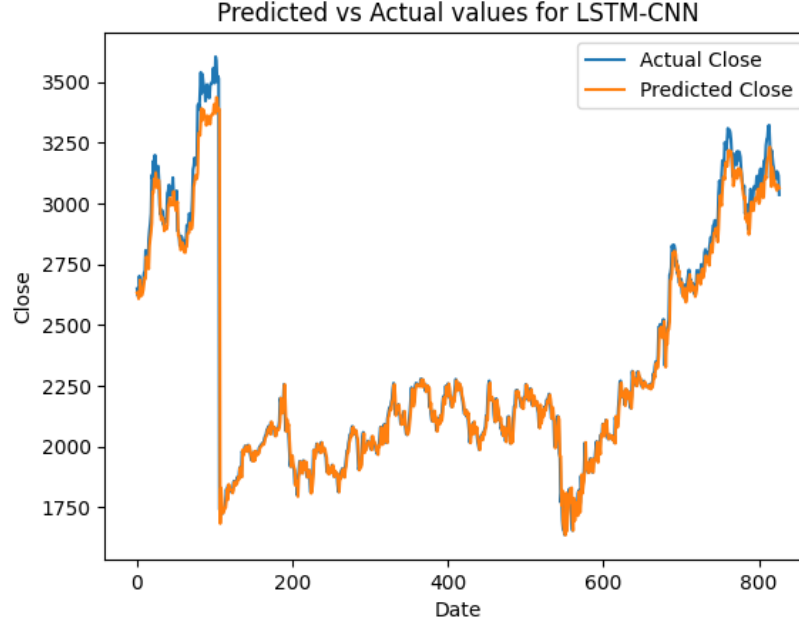
For this task, we have implemented and compared three well-known Machine Learning techniques, Convolutional Neural Network Long Short-Term Memory Model (CNN-LSTM), K-Nearest Neighbours & Multi-Linear Regression (MLR).

### 3.1 CNN-LSTM

This model employs a short history of a stock, utilizing the prices of the previous 5 days as features to forecast future prices. It adopts a hybrid architecture, integrating Convolutional Neural Networks (CNNs) with an LSTM layer, followed by a linear layer for prediction. The model is trained using mean squared error loss and the Adam optimizer over 20 epochs. The results with Mean Squared Error (MSE) are as follows:

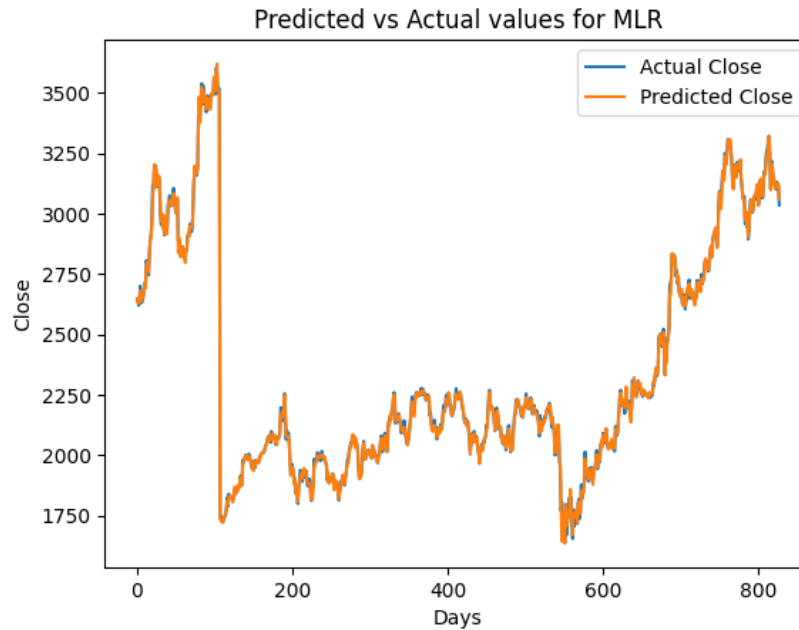
Table 1: LSTM Data Table Used

Date	Close	Close t-1	Close t-2	Close t-3	Close t-4	Close t-5
2004-09-01	987.90	988.10	986.75	962.65	979.00	987.95
2004-09-02	993.65	987.90	988.10	986.75	962.65	979.00
2004-09-03	997.85	993.65	987.90	988.10	986.75	962.65



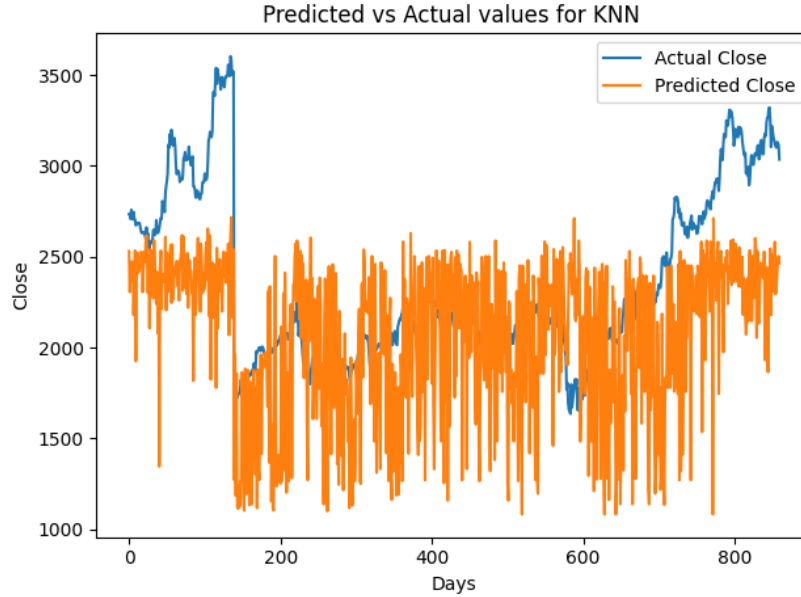
### 3.2 Multi-Linear Regression

In this method, a multi-linear regression (MLR) model is applied to predict the Close Price of a stock using various features including "Open," "High," "Low," "Volume," and "Turnover" from the dataset. The results as follows:



### 3.3 K-Nearest Neighbours

In this method, the K-Nearest Neighbors Regression (KNN) algorithm is applied for predicting the stock's closing price. The KNN algorithm works by finding the 'k' nearest data points in the feature space and averaging their target values to make predictions. The results are as follows:



Method	MSE on Test Dataset
<i>CNN-LSTM</i>	<b>0.04678102</b>
<i>MLR</i>	<b>0.00010230</b>
<i>KNN</i>	<b>0.41147872</b>

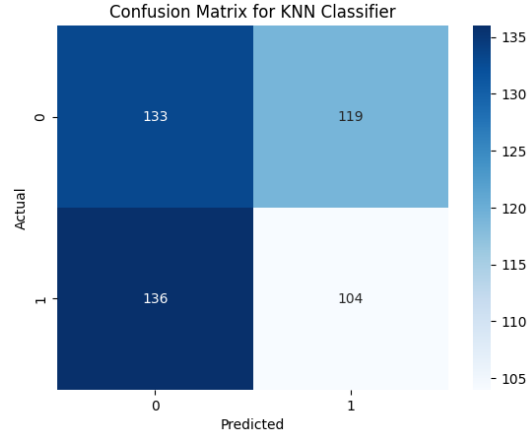
From the Predicted v/s Actual Values Graphs and the MSE Errors Table, we can conclude that for this dataset our MLR Model performs the best out of the three models created and KNN performs the worst.

## 4 Stock Classification Based on Growth

In this section, we have used K Nearest Neighbours and Decision Trees to classify whether the price of a stock will go higher than the price of previous of day or lower.

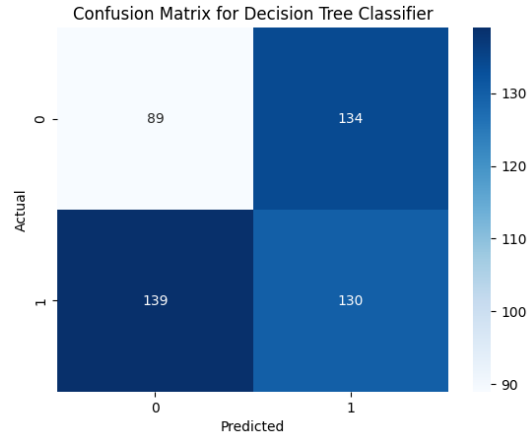
### 4.1 K-Nearest Neighbours

In this method, the K-Nearest Neighbors (KNN) algorithm is used for a classification problem. The objective is to predict whether to Buy (+1) or Sell (-1), essentially predicting whether the stock will rise or fall. The results are as follows:



## 4.2 Decision Trees

In this method, a Decision Trees Classifier is employed to predict stock price movements based on features such as "Open - Close," "High - Low," "Volume," and "Turnover." The target variable is determined by comparing the closing price of the current day with that of the next day. The labels of +1 and -1 mean the same here as it did in the KNN Model. Since this model gives a different result everytime we took the average training and testing accuracies by running it 100 times. The results are as follows:



Method	MSE on Test Dataset
<i>KNN</i>	<b>0.4817073170731707</b>
<i>Dec. Trees</i>	<b>0.4539024390243902</b>

From the Confusion Matrices and the MSE Errors Table, we can see that the accuracies for both the models are very similar and pretty low.

## 5 Future Work

As part of the second submission, we aim to include an option given to the user to choose any of the 50 stocks. Furthermore, the future submission will include the third section of our project, which is the classification of profitable pairs for pairs trading. If time permits, we would also like to add XGBoost method to predict stock prices.