# Medical Stock Interpretation–pre-Covid and the reclamation phase

**Team Members:**
*19BDS0073 (Pavankumar S)*
*19BDS0065 (Hariharan R)*

**Report submitted for the**
**Final Project Review of**

**Course Code: CSE3045**
**Predictive Analysis**

**Slot: A2**

**Professor: Dr. Ilanthenral Kandasamy**

**1. Introduction**:

Medical logistics is the logistics of surgical, medical, and pharmaceutical supplies. It also includes the supplies of medical devices, medical and laboratory equipment, and other items, products, and pieces of equipment to support dentists, doctors, veterinary physicians, nurses, and other healthcare providers. With the ever-increasing demand for medical logistics due to the surge in population, there exists always a need to get the goods, to be delivered on time with minimal cost and handle them with care to avoid any wastage. Faulty machines, expired medicines, and human negligence continue to be significantly contributing to medical errors which culminate in being the 3rd leading cause of death in the United States. With proper estimation of requirement, we can always expect the industry to fulfil the ever-rising demand. Surges in demand of medical equipment were observed during the pandemic especially with the case of oxygen concentrators and anti-viral pills. With the raise in demand of these goods the pharma industry stocks ran bull, as their equity reached all time high values. These patterns can reveal the expected demand for a similar scenario in future. The trends in the pharma sector stock values provide highly valuable information, as the pre-covid and post-covid analysis of these stocks reveal that the pharma sector have benefitted the most from the pandemic. Our research aims to use LSTM model (Long Short-Term Memory) to interpret meaningful information from the pharma industry stock data and compare it with the trends of the other sectors. We also plan to use other machine learning models to estimate the effectiveness of LSTM model and with a hypothesis that the LSTM model would perform better than other models we proceed with this project.

**2. Literature Review Summary Table**

Kindly go through projects, and review papers related to your project and study them. Minimum at least five projects/papers should be reviewed so that you have a considerable understanding of what is achieved in your project area.

| Authors and Year | Title (Study) | Concept/Theoretical model/ Framework | Methodology used / Implementation | Dataset details / Analysis | Relevant Finding | Limitations / Future Research/ Gaps identified |
|---|---|---|---|---|---|---|
| Dattatray P., | Systematic | This work presents the | ANN, SVM, | ANN-based | The commonl | The research |

| | | | | | | |
|---|---|---|---|---|---|---|
| Gandhmal, K. Kumar – 2019 | analysis and review of stock market prediction techniques | detailed review of 50 research papers suggesting the methodologies, like Bayesian model, Fuzzy classifier, Artificial Neural Networks (ANN), Support Vector Machine (SVM) classifier, Neural Network (NN), Machine Learning Methods | SVR, HMM, NN, fuzzy based techniques, K-means. | prediction techniques ANN captures the structural relationship between a stock's performance and its determinant factors. The deep learning methods are utilized for determining and analyzing complicated patterns in the data and allow to speed up the trading process. | y used technique for attaining effective stock market prediction is ANN and the fuzzy-based technique. | gaps and the issues for predicting the stock market are elaborated for suggesting effective future scope. |
| Weiwei Jiang – 2021 | Applications of deep learning in stock market prediction: Recent progress | To give a latest review of recent works on deep learning models for stock market prediction. | FFNN, ANN, CNN, DNN, MLP, IDLNN, STEFNN RBFN | CNN performs a convolution operation to a different region of the input image and the neurons share the same weights, which reduces | Powered by the parallel processing ability of graphics processing unit (GPU), the training of CNN has been shortened and CNN has achieved an | Only the recent progress of the deep learning application in the stock market is covered in this survey. The second point is that the scope of this survey is limited to the stock market. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | the number of parameters compared to the densely connected feedforward neural network. | astonishing performance for image related tasks and competitions. | |
| Alan D.KayeMD, Chikezie N.OkeaguMDAlex D.Pham MDRayce A.Silva,Joshua J.Hurley, Brett L.Arron MD - 2021 | Economic impact of COVID-19 pandemic on healthcare facilities and systems: International perspectives | Effective risk reduction strategies to prevent airborne and contact transmission of the novel COVID 19 requires a system safety approach factoring various practies and human factors. | This is a survey paper, there is no implementation used. | This is a survey paper, there are no datasets being used. | | |

### 3. Objective of the project:

The research is carried out with multiple objectives that could be implemented by a single process. Our objectives include, but not only limited to,

- Identify the patterns analyse the trends of the pharma industry and interpret useful information out of it.
- Identify the pattern that would lead to the crash of stock market and observe the recovery pattern (reclamation phase)
- Identifying patterns to understand the demands of the pharmaceuticals, and with this knowledge we expect the industries to be prepared to address a demand of similar kind in future.
- Finally, the project analyses the effectiveness of different machine learning algorithms in predicting the price of stocks and conveying the performance of the stocks with good visualisations would mean that we have achieved our objective.

**4. Innovation component in the project:** The most innovative part of your project should be explained. How does your project defer from others and the routine experimental results? Highlight the detailed design showing system design with components / modules/parts/layers etc of your project.

**5. Work done and implementation**

**a. Methodology adapted:** The Methodology used should be discussed in detail. Hardware and software requirements must also be mentioned.

**b. Dataset used:**

a. Where from you are taking your dataset? *Narrow down your data to what is needed.*
b. Is your project based on any other reference project (Stanford Univ. or MIT)?
c. How does your project differ from the reference project?*

**c. Tools used:**

1. **Jupyter Notebook App**

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.Jupyter notebooks are used for all sorts of data science tasks such as exploratory data analysis (EDA), data cleaning and transformation, data visualization, statistical modeling, machine learning, and deep learning.

2. **Python**
3. **R Programming Language**
4. **RStudio:**

R Studio is an integrated development environment(IDE) for R. IDE is a GUI, where you can write your quotes, see the results and also see the variables that are generated during the course of programming.

5. **RStudio:**
   **ggplot**
   **dplyr**
6. **Python Libraries**
   a) **Keras**
   b) **Tensorflow**
   c) **Matplotlib**
   d) **Scikit Learn**
   e) **Seaborn**

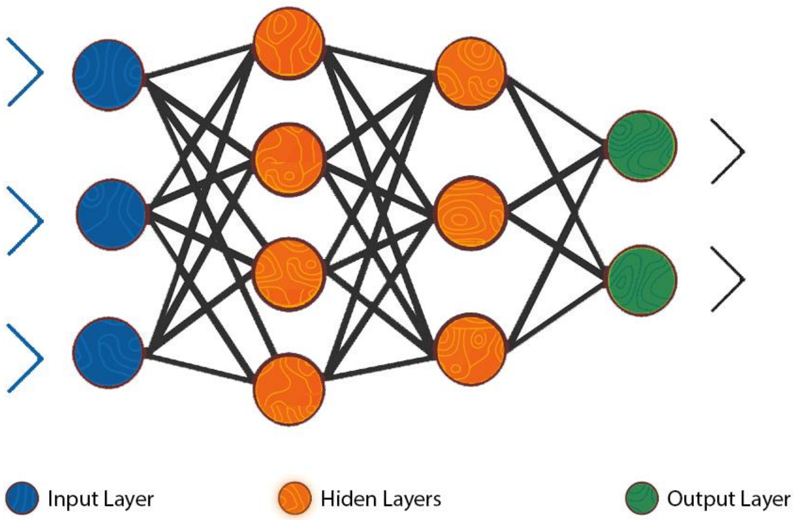
**d. Screenshot and Demo along with Visualization: (Preprocessing)** Each result, and necessary coding part should be substantiated with related screenshot.

**e. Models used:**

**1) Artificial Neural Networks:**

An ANN usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information -- analogous to optic nerves in human visual processing. Each successive tier receives the output from the tier preceding it, rather than the raw input -- in the same way neurons further from the optic nerve receive signals from those closer to it. The last tier produces the output of the system.

Each processing node has its own small sphere of knowledge, including what it has seen and any rules it was originally programmed with or developed for itself. The tiers are highly interconnected, which means each node in tier n will be connected to many nodes in tier n-1 -- its inputs -- and in tier n+1, which provides input data for those nodes. There may be one or multiple nodes in the output layer, from which the answer it produces can be read.

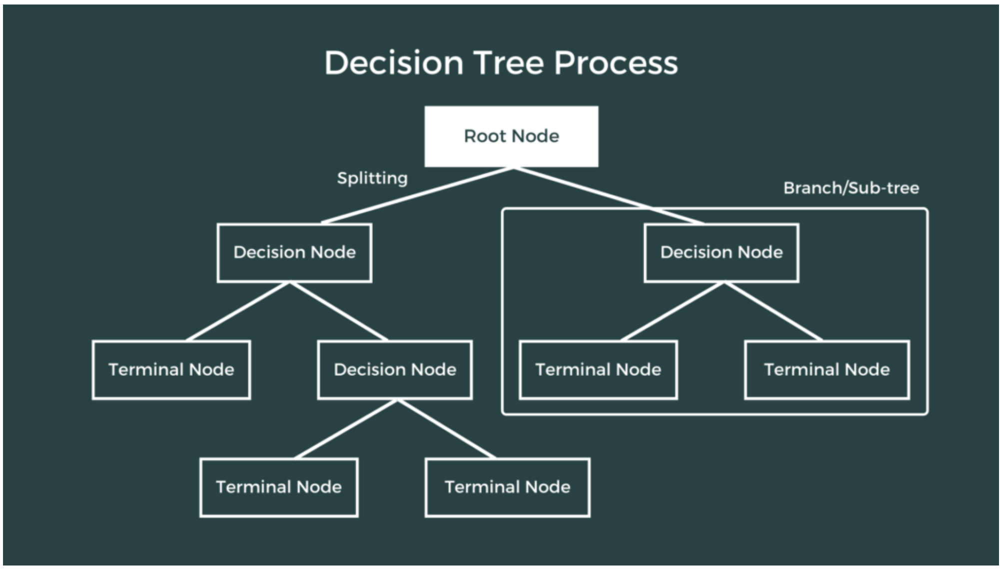



**2) Decision Tree Regression:**

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. A regression tree is basically a decision tree that is used for the task of regression which can be used to predict continuous valued outputs instead of discrete outputs. Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

The branches/edges represent the result of the node and the nodes have either:

- Conditions [Decision Nodes]
- Result [End Nodes]

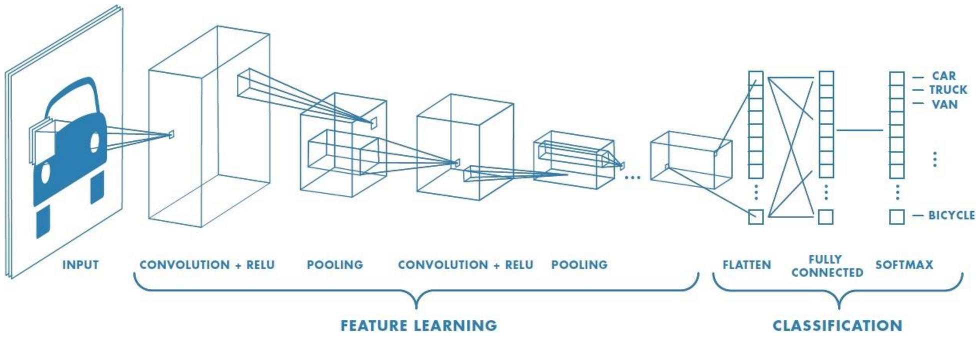



3) **Convolutional Neural Network:**

Convolutional Neural Networks also known as CNNs or ConvNets, are a type of feed-forward artificial neural network whose connectivity structure is inspired by the organization of the animal visual cortex. Small clusters of cells in the visual cortex are sensitive to certain areas of the visual field. Individual neuronal cells in the brain respond or fire only when certain orientations of edges are present. Some neurons activate when shown vertical edges, while others fire when shown horizontal or diagonal edges. A convolutional neural network is a type of artificial neural network used in deep learning to evaluate visual information.

Convolutional Neural Networks (CNNs) have an input layer, an output layer, numerous hidden layers, and millions of parameters, allowing them to learn complicated objects and patterns. It uses convolution and pooling processes to sub-sample the given input before applying an activation function, where all of them are hidden layers that are partially connected, with the completely connected layer at the end resulting in the output layer. The output shape is similar to the size of the input image.

Convolution is the process of combining two functions to produce the output of the other function. The input image is convoluted with the application of filters in CNNs, resulting in a Feature map. Filters are weights and biases that are randomly generated vectors in the network. Instead of having individual weights and biases for each neuron, CNN uses the same weights and biases for all neurons



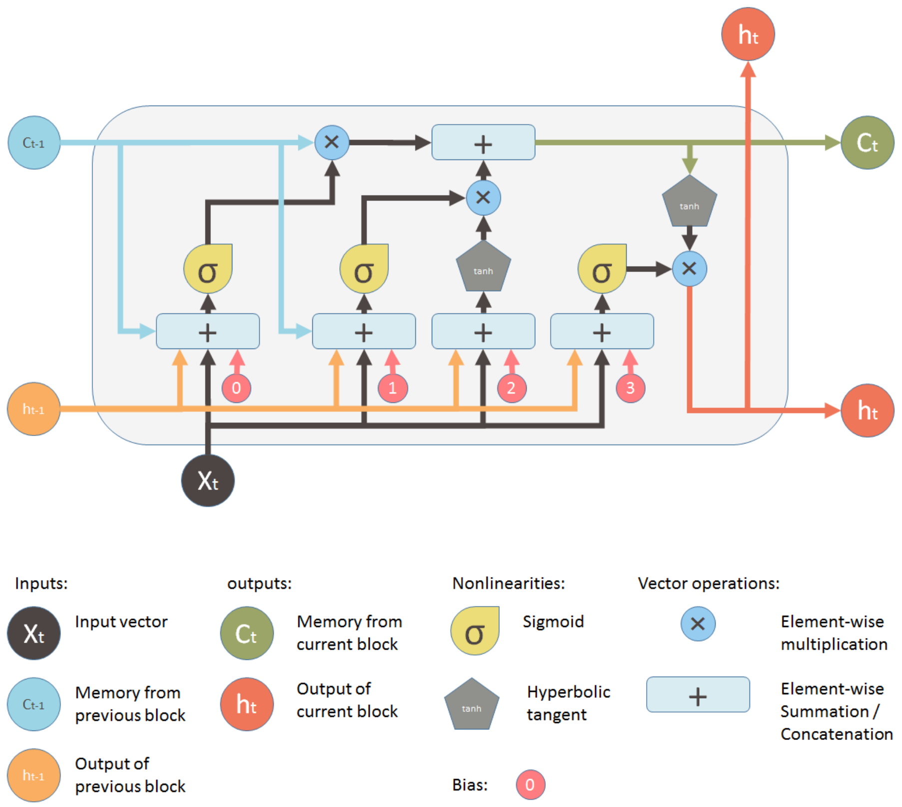

**4) LSTM:**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Time is a very important factor in stock market, as the golden rule of stock market tells that "History repeats itself", so with careful analysis on time, supported by huge volumes of data, one can be able to interpret the price of the stock in future and identify the reason behind the format of patterns.

The more time passes, the less likely it becomes that the next output depends on a very old input. This time dependency distance itself is as well as contextual information to be learned. LSTM networks manage this by learning when to remember and when to forget.

LSTM has three gates:

- The input gate: The input gate adds information to the cell state,
- The forget gate: It removes the information that is no longer required by the model,
- The output gate: Output Gate at LSTM selects the information to be shown as output.




## 5) KNN Regression

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood. The KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.

There are various methods for calculating this distance, of which the most commonly known methods are – Euclidian, Manhattan (for continuous) and Hamming distance (for categorical).

1. **Euclidean Distance:** Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).
2. **Manhattan Distance:** This is the distance between real vectors using the sum of their absolute difference.

**Distance functions**

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}\left|x_i - y_i\right|$

3. **Hamming Distance:** It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

$$D_H = \sum_{i=1}^{k}\left|x_i - y_i\right|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

Once the distance of a new observation from the points in our training set has been measured, the next step is to pick the closest points. The number of points to be considered is defined by the value of k. The next step is to select the k value. This determines the number of neighbours we look at when we assign a value to any new observation. Based on the k value, the result tends to change.

For a very low value of k (suppose k=1), the model overfits on the training data, which leads to a high error rate on the validation set. On the other hand, for a high value of k, the model performs poorly on both train and validation set.We can also use the grid search technique to find the best k value.

**f. Screenshot and Demo along with Visualization (For results):**

# 1) CNN Model for CIPLA data:

In [28]:

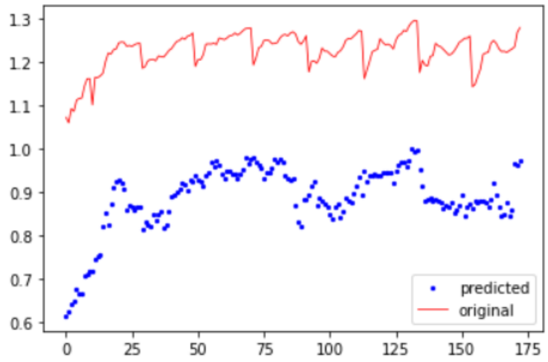
```python
import math

ypred = model.predict(X_test_new)
print("MSE: %.4f" % mean_squared_error(y_test, ypred))
trainScore = model.evaluate(X_train_new, y_train, verbose=0)
```

```
MSE: 0.1159
```

In [29]:

```python
x_ax = range(len(ypred))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend(["predicted", "original"], loc ="lower right")
```

Out[29]: <matplotlib.legend.Legend at 0x24f93072880>



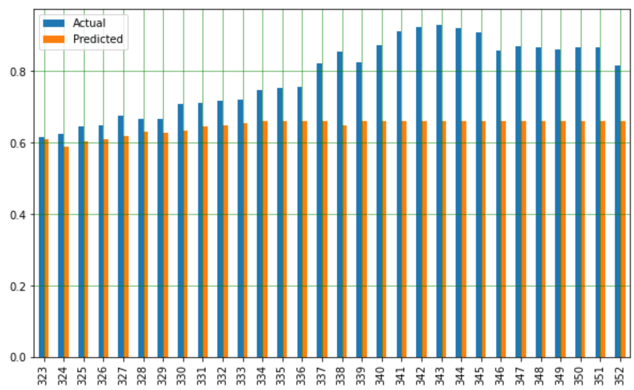

# 2) ANN Model for CIPLA data:

**ANN**

```python
from sklearn.neural_network import MLPRegressor
mlp_reg = MLPRegressor(hidden_layer_sizes=(5,1),
                       max_iter = 1000,activation = 'relu',
                       solver = 'adam')

mlp_reg.fit(X_train_new[['Open','High','Low','Volume']],y_train)
y_pred = mlp_reg.predict(X_test_new[['Open','High','Low','Volume']])
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

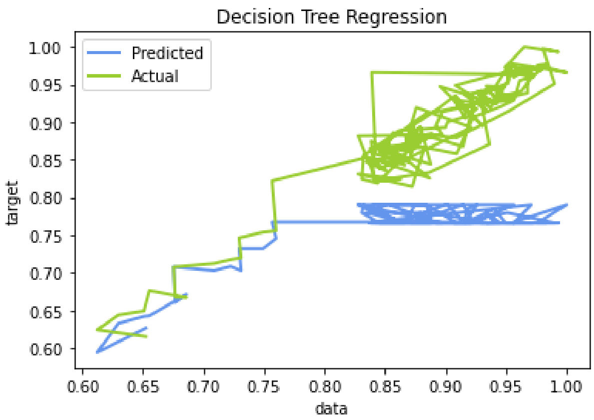
```
Mean Absolute Error: 0.22864940625103197
Mean Squared Error: 0.05683669349126584
Root Mean Squared Error: 0.2384044745621731
```

```python
df_temp2 = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

df_temp2 = df_temp2.head(30)
df_temp2.plot(kind='bar',figsize=(10,6))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

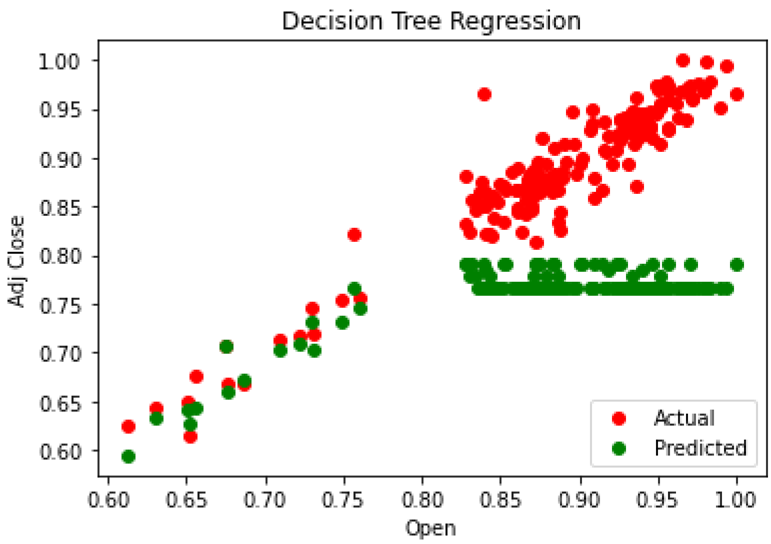


## 3) Decision tree Regression Model for CIPLA data:

```
plt.figure()
plt.plot(X_test['Open'], y_predicted, color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Open'], y_test, color="yellowgreen", label="Actual", linewidth=2)
plt.xlabel("data")
plt.ylabel("target")
plt.title("Decision Tree Regression")
plt.legend()
plt.show()

```



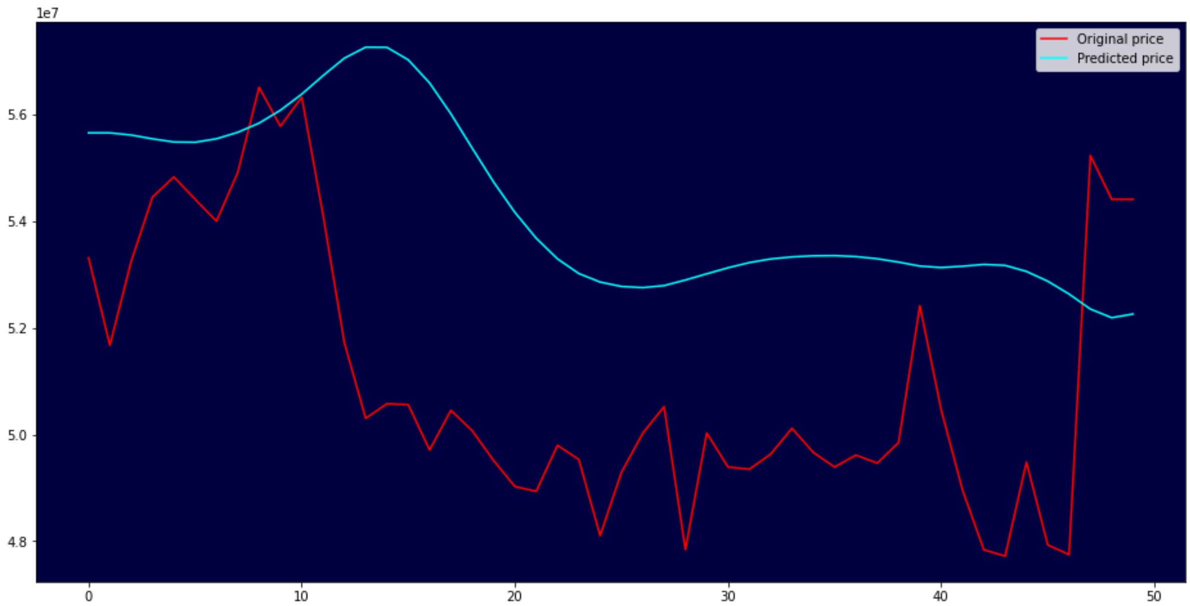

```
plt.scatter(X_test_new['Open'], y_test, color = 'red')
plt.scatter(X_test_new['Open'], y_predicted, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Open')
plt.ylabel('Adj Close')
plt.legend(["Actual", "Predicted"], loc ="lower right")
plt.show()

```




## 4) LSTM Model for CIPLA data:

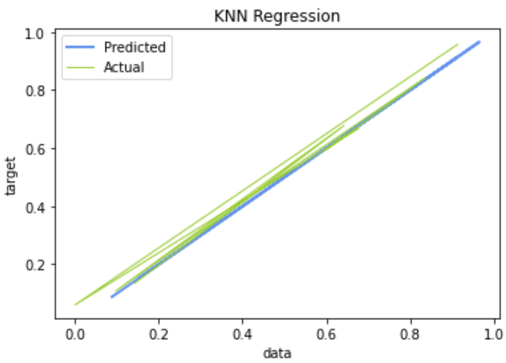
```python
from keras.models import Sequential, load_model
model = load_model('stock_prediction1.h5')
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
y_test_scaled = scaler.inverse_transform(y_test.reshape(-1, 1))

fig, ax = plt.subplots(figsize=(16,8))
ax.set_facecolor('#000041')
ax.plot(y_test_scaled, color='red', label='Original price')
plt.plot(predictions, color='cyan', label='Predicted price')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x24fa2d88700>
```




## 5) KNearest Neighbours Regression model for CIPLA data:

```python
plt.figure()
plt.plot(X_train['Adj Close'].values[:10], y_train[:10], color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Adj Close'].values[:10], preds[:10], color="yellowgreen", label="Actual", linewidth=1)
plt.xlabel("data")
plt.ylabel("target")
plt.title("KNN Regression")
plt.legend()
plt.show()
```




## 6) CNN Model for DR.REDDY data:

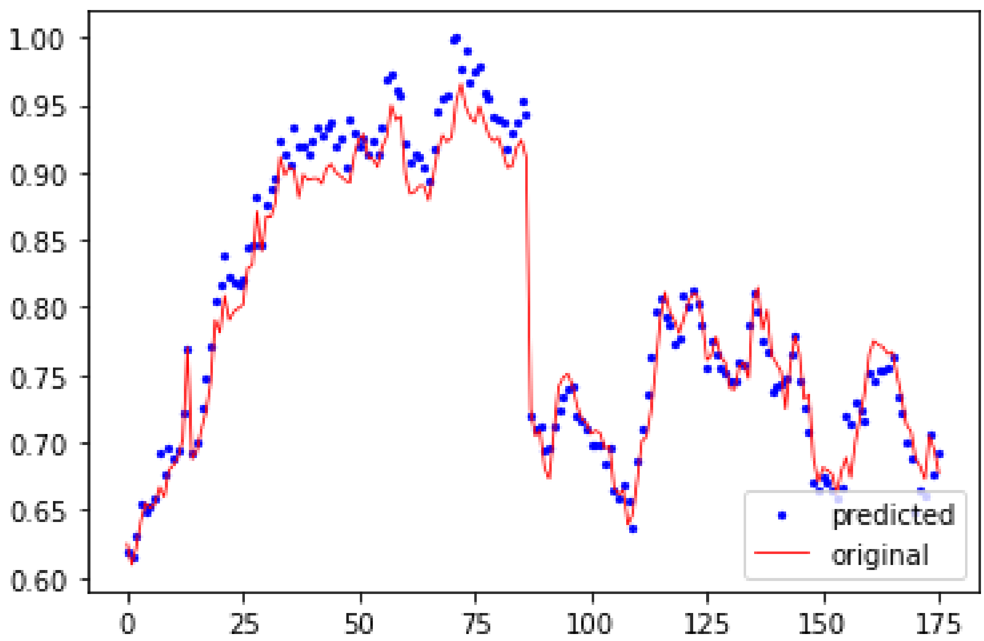
```python
model = build_model2()
model.fit(
    X_train_new,
    y_train,
    batch_size=12,epochs=200, verbose=0)
```

```
<keras.callbacks.History at 0x24fa24704f0>
```

```python
import math

ypred = model.predict(X_test_new)
print("MSE: %.4f" % mean_squared_error(y_test, ypred))
trainScore = model.evaluate(X_train_new, y_train, verbose=0)
```

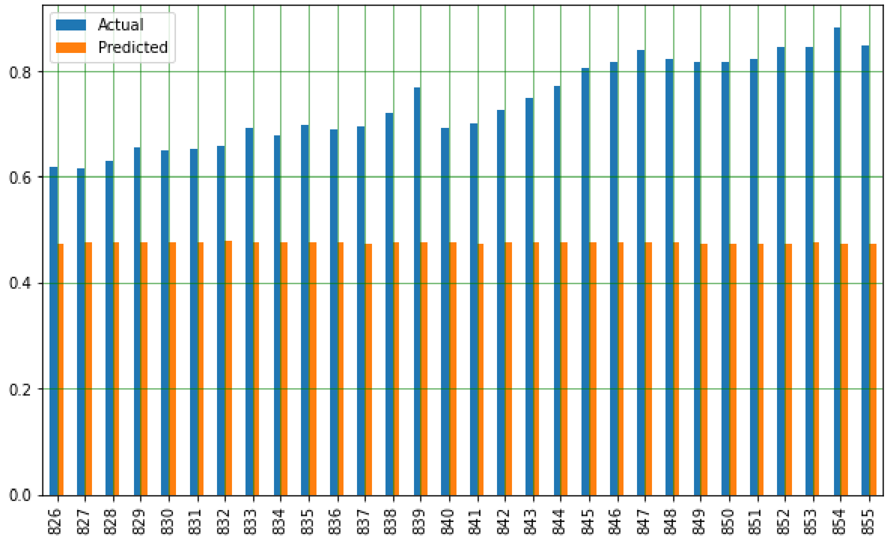
```
MSE: 0.0004
```

```python
x_ax = range(len(ypred))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend(["predicted", "original"], loc ="lower right")
```



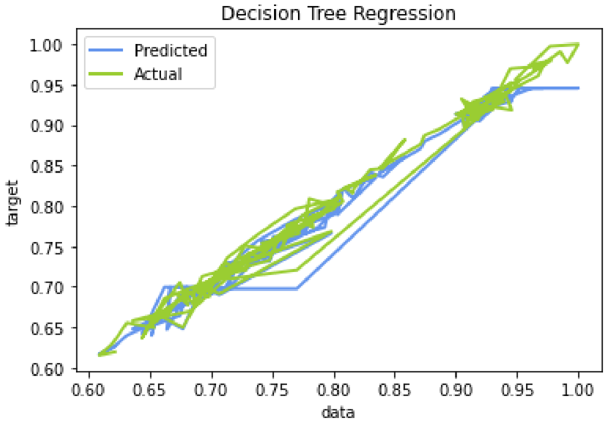

**7) ANN Model for DR.REDDY data:**

```python
df_temp2 = pd.DataFrame({'Actual': y_test, 'Predicted': y_predicted})

df_temp2 = df_temp2.head(30)
df_temp2.plot(kind='bar',figsize=(10,6))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

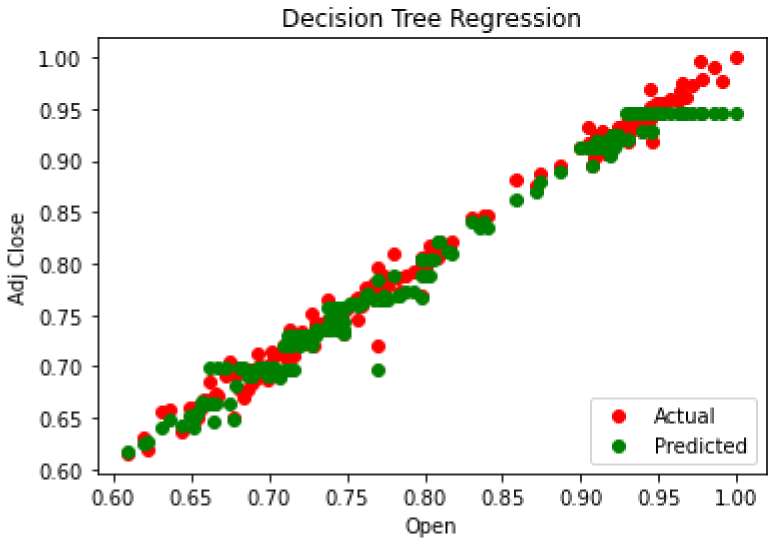



## 8) Decision tree Model for DR.REDDY data:

```python
plt.figure()
plt.plot(X_test['Open'], y_predicted, color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Open'], y_test, color="yellowgreen", label="Actual", linewidth=2)
plt.xlabel("data")
plt.ylabel("target")
plt.title("Decision Tree Regression")
plt.legend()
plt.show()
```



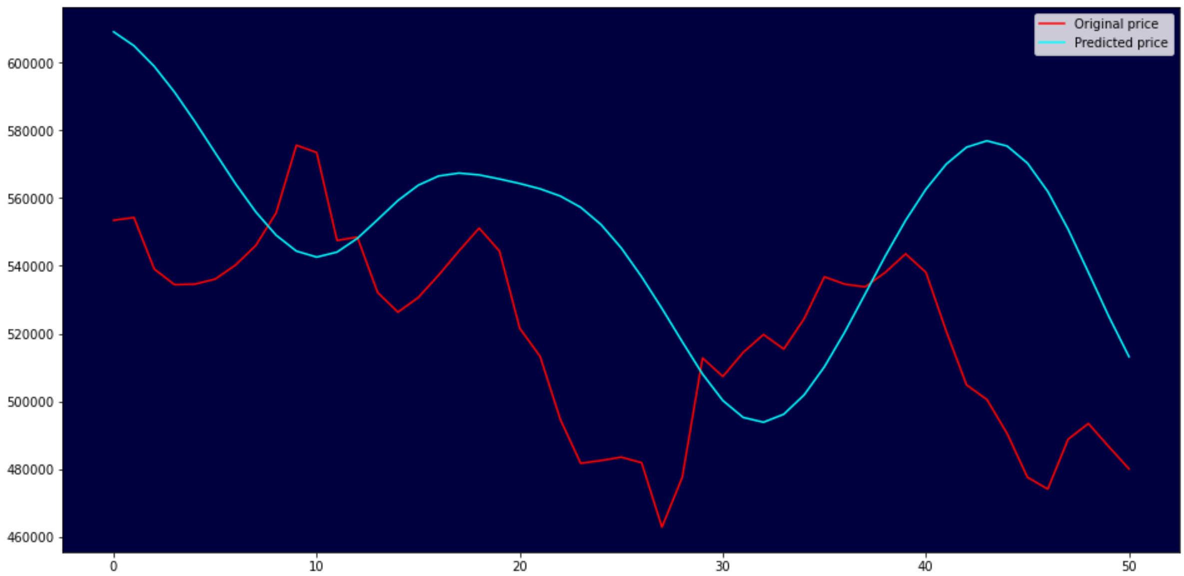
```python
plt.scatter(X_test_new['Open'], y_test, color = 'red')
plt.scatter(X_test_new['Open'], y_predicted, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Open')
plt.ylabel('Adj Close')
plt.legend(["Actual", "Predicted"], loc ="lower right")
plt.show()
```




## 9) LSTM Model for DR.REDDY data:

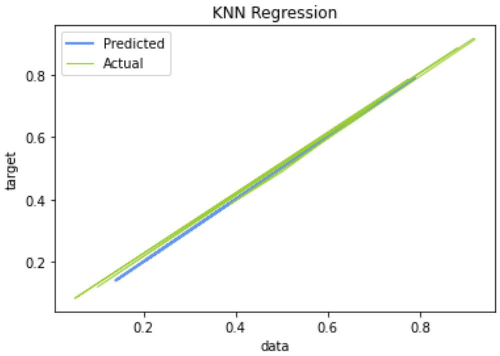
```python
from keras.models import Sequential, load_model
model = load_model('stock_prediction2.h5')
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
y_test_scaled = scaler.inverse_transform(y_test.reshape(-1, 1))

fig, ax = plt.subplots(figsize=(16,8))
ax.set_facecolor('#000041')
ax.plot(y_test_scaled, color='red', label='Original price')
plt.plot(predictions, color='cyan', label='Predicted price')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x24fa35a9670>
```




## 10) KNearest Neighbours Regression model for DR.REDDY data:

```python
plt.figure()
plt.plot(X_train['Adj Close'].values[:10], y_train[:10], color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Adj Close'].values[:10], preds[:10], color="yellowgreen", label="Actual", linewidth=1)
plt.xlabel("data")
plt.ylabel("target")
plt.title("KNN Regression")
plt.legend()
plt.show()
```



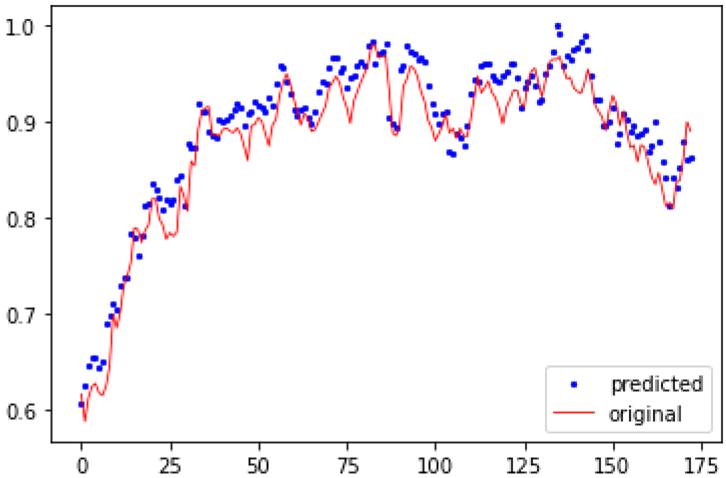

## 11) CNN Model for PHARMA NIFTY data:

```python
ypred = model.predict(X_test_new)
print("MSE: %.4f" % mean_squared_error(y_test, ypred))
trainScore = model.evaluate(X_train_new, y_train, verbose=0)
```

```
MSE: 0.0007
```

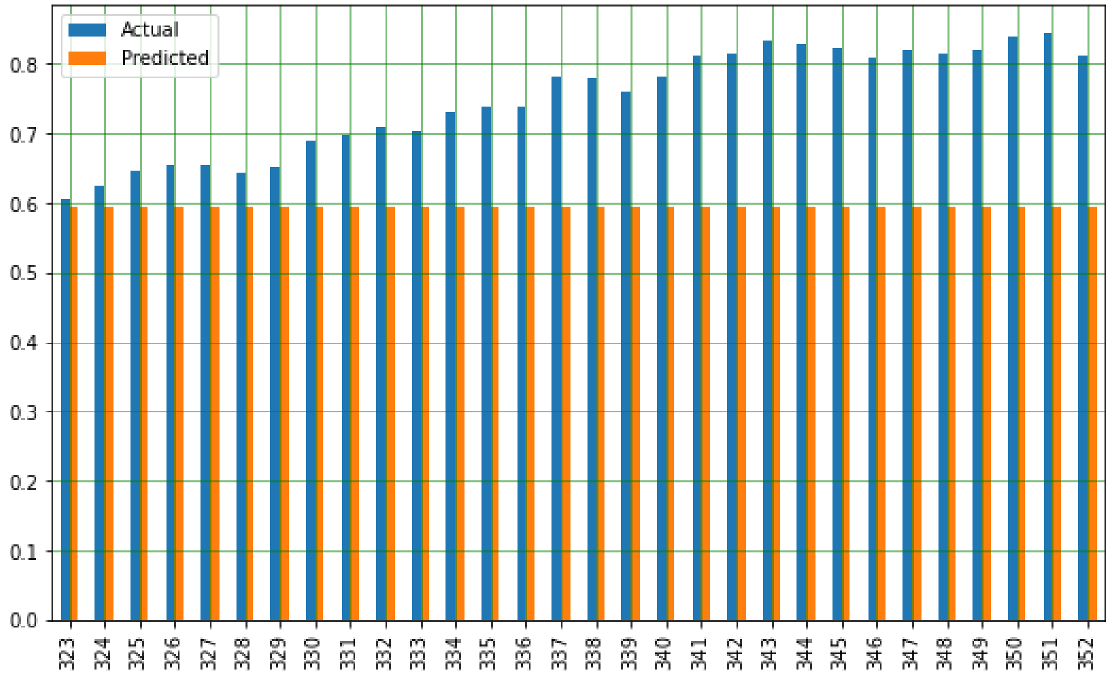
```python
x_ax = range(len(ypred))
plt.scatter(x_ax, y_test, s=5, color="blue", label="original")
plt.plot(x_ax, ypred, lw=0.8, color="red", label="predicted")
plt.legend(["predicted", "original"], loc ="lower right")
```

```
<matplotlib.legend.Legend at 0x24fa4097910>
```



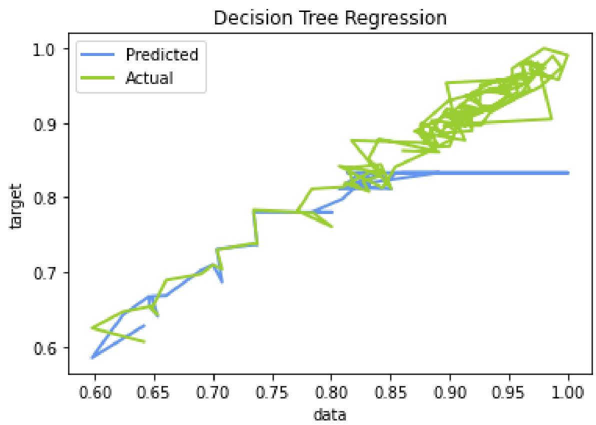

## 12) ANN Model for PHARMA NIFTY data:

```
df_temp2 = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})

df_temp2 = df_temp2.head(30)
df_temp2.plot(kind='bar',figsize=(10,6))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



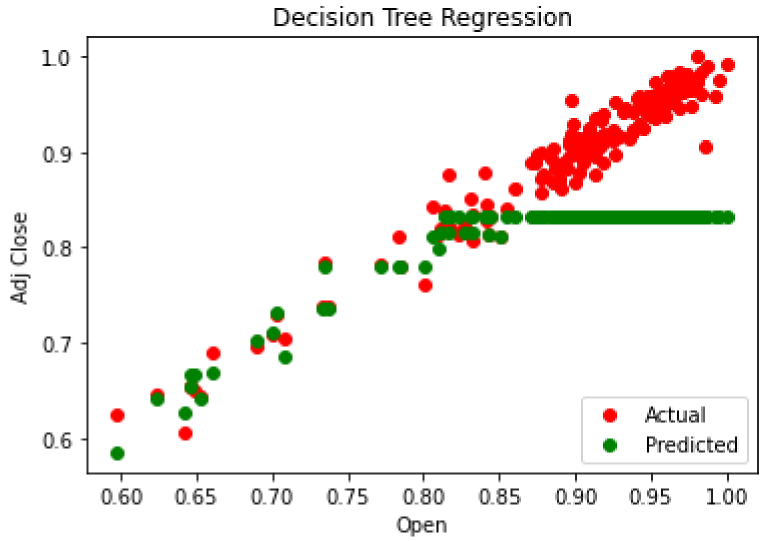

## 13) Decision tree Model for PHARMA NIFTY data:

```
plt.figure()
plt.plot(X_test['Open'], y_predicted, color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Open'], y_test, color="yellowgreen", label="Actual", linewidth=2)
plt.xlabel("data")
plt.ylabel("target")
plt.title("Decision Tree Regression")
plt.legend()
plt.show()
```



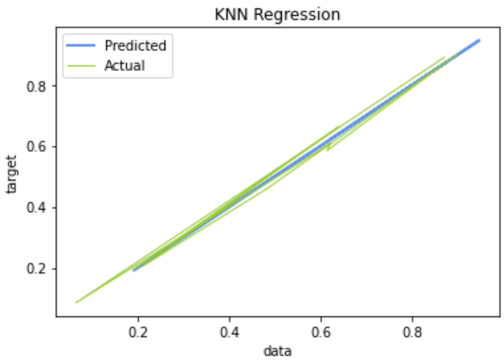

## 14) LSTM Model for PHARMA NIFTY data:

```python
plt.scatter(X_test_new['Open'], y_test, color = 'red')
plt.scatter(X_test_new['Open'], y_predicted, color = 'green')
plt.title('Decision Tree Regression')
plt.xlabel('Open')
plt.ylabel('Adj Close')
plt.legend(["Actual", "Predicted"], loc ="lower right")
plt.show()
```




### 15) KNearest Neighbours Regression model for PHARMA NIFTY data:

```python
plt.figure()
plt.plot(X_train['Adj Close'].values[:10], y_train[:10], color="cornflowerblue", label="Predicted", linewidth=2)
plt.plot(X_test['Adj Close'].values[:10], preds[:10], color="yellowgreen", label="Actual", linewidth=1)
plt.xlabel("data")
plt.ylabel("target")
plt.title("KNN Regression")
plt.legend()
plt.show()
```



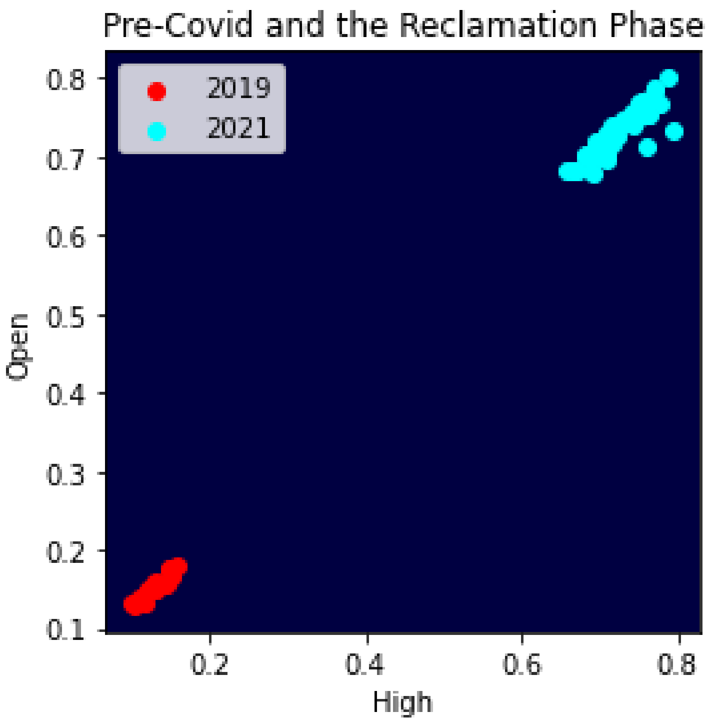

## 6. Comparison, Results and discussion along with Visualization

| Serial Number | Model Name | Output | Remarks |
| --- | --- | --- | --- |
| 1 | ANN | FOR CIPLA<br>Mean Absolute Error: 0.22864940625103197<br>Mean Squared Error: 0.05683669349126584<br>Root Mean Squared Error: 0.2384044745621731<br><br>FOR DR.REDDY<br>The model performance for testing set<br>--------------------------------------<br>MAE is 0.32345663981272393<br>MSE is 0.11616349809740925<br>R2 score is -9.243393269623084<br><br>FOR PHARMA NIFTY<br>Mean Absolute Error: 0.2995038335672276<br>Mean Squared Error: 0.09624189274944055<br>Root Mean Squared Error: 0.3102287748572665 | The ANN have performed quite well, with respect to all the three datasets. |
| 2 | CNN | FOR CIPLA<br>MSE: 0.1159<br><br>FOR DR.REDDY<br>MSE: 0.0004<br><br>FOR PHARMA NIFTY<br>MSE: 0.0007 | CNN have performed really well and a mean square error with such less magnitude means that this model can be preferred over other models. |
| 3 | Decision Tree Regression | FOR CIPLA<br>The model performance for testing set<br>--------------------------------------<br>MAE is 0.12454251436426464<br>MSE is 0.01879033262941087<br>R2 score is -2.3898922715306146<br><br>FOR DR.REDDY<br>The model performance for testing set<br>--------------------------------------<br>MAE is 0.010087318697405078<br>MSE is 0.00018648840083748157<br>R2 score is 0.9835552986842772<br><br>FOR PHARMA NIFTY | The R2 score of -2 indicates that model haven't been good enough . The R2 score if it is positive indicates that model is really good. |

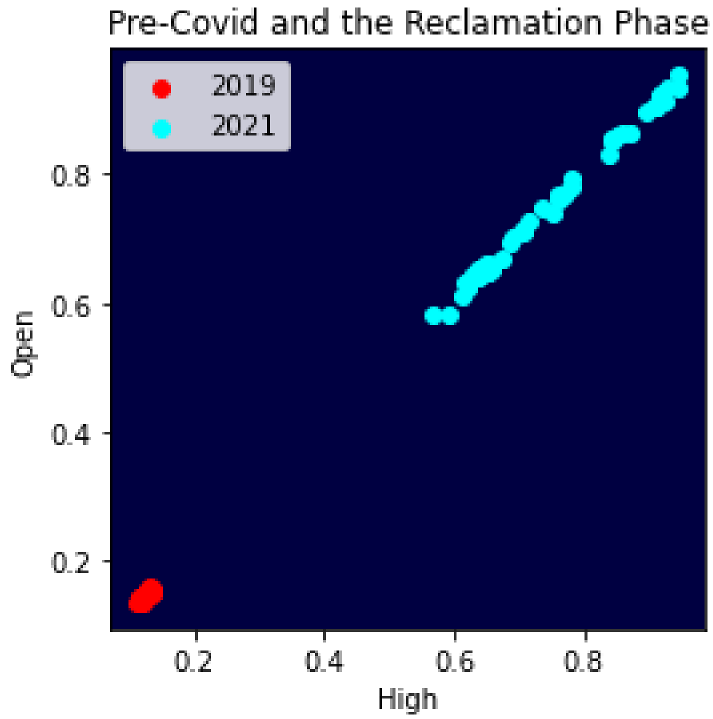
| | | | |
|---|---|---|---|
| | | The model performance for testing set<br>--------------------------------------<br>MAE is 0.07741137085497005<br>MSE is 0.008084828084084407<br>R2 score is -0.2363357979473213 | |
| 4 | LSTM | FOR CIPLA<br><br>Epoch 1/50<br>11/11 [==============================] - 21s 207ms/step - loss: 0.1038<br>Epoch 2/50<br>11/11 [==============================] - 2s 226ms/step - loss: 0.0227<br>Epoch 3/50<br>11/11 [==============================] - 2s 234ms/step - loss: 0.0202<br>Epoch 4/50<br>11/11 [==============================] - 3s 231ms/step - loss: 0.0121<br>Epoch 5/50<br>11/11 [==============================] - 2s 206ms/step - loss: 0.0093<br>Epoch 6/50<br>11/11 [==============================] - 3s 230ms/step - loss: 0.0081<br>Epoch 7/50<br>11/11 [==============================] - 2s 222ms/step - loss: 0.0088<br>Epoch 8/50<br>11/11 [==============================] - 2s 212ms/step - loss: 0.0084<br>Epoch 9/50<br>11/11 [==============================] - 2s 204ms/step - loss: 0.0075<br>Epoch 10/50<br>11/11 [==============================] - 2s 208ms/step - loss: 0.0081<br><br>FOR DR.REDDY<br>Epoch 1/50<br>11/11 [==============================] - 15s 177ms/step - loss: 0.1423<br>Epoch 2/50<br>11/11 [==============================] - 2s 163ms/step - loss: 0.0304<br>Epoch 3/50<br>11/11 [==============================] - 2s 155ms/step - loss: 0.0188<br>Epoch 4/50<br>11/11 [==============================] - 2s 168ms/step - loss: 0.0143<br>Epoch 5/50<br>11/11 [==============================] - 2s 150ms/step - loss: 0.0133<br>Epoch 6/50<br>11/11 [==============================] - 1s 132ms/step - loss: 0.0110<br>Epoch 7/50<br>11/11 [==============================] - 1s 126ms/step - loss: 0.0110<br>Epoch 8/50<br>11/11 [==============================] - 1s 109ms/step - loss: 0.0118<br>Epoch 9/50<br>11/11 [==============================] - 2s 155ms/step - loss: 0.0104<br>Epoch 10/50<br>11/11 [==============================] - 2s 139ms/step - loss: 0.0108<br>Epoch 11/50<br><br>FOR PHARMA NIFTY | The LSTM's performance was visualised using the graph and their effectiveness can be observed using the loss found during the epoch runs. It reveals that the model has been really good and is the best performing model among all other models. |

| | | | |
|---|---|---|---|
| | | Epoch 1/50<br>11/11 [==============================] - 26s 168ms/step - loss: 0.0971<br>Epoch 2/50<br>11/11 [==============================] - 2s 195ms/step - loss: 0.0244<br>Epoch 3/50<br>11/11 [==============================] - 2s 154ms/step - loss: 0.0149<br>Epoch 4/50<br>11/11 [==============================] - 2s 180ms/step - loss: 0.0113<br>Epoch 5/50<br>11/11 [==============================] - 2s 205ms/step - loss: 0.0086<br>Epoch 6/50<br>11/11 [==============================] - 2s 190ms/step - loss: 0.0075<br>Epoch 7/50<br>11/11 [==============================] - 2s 170ms/step - loss: 0.0089<br>Epoch 8/50<br>11/11 [==============================] - 2s 175ms/step - loss: 0.0075<br>Epoch 9/50<br>11/11 [==============================] - 2s 161ms/step - loss: 0.0075<br>Epoch 10/50<br>11/11 [==============================] - 2s 220ms/step - loss: 0.0073 | |
| 5 | KNN Regression | FOR CIPLA<br>5 rms<br>0.02832622131390678<br>FOR DR.REDDY<br>2 rms<br>0.017027592384664367<br>FOR PHARMA NIFTY<br>2 rms<br>0.022825857609070386 | The KNN regression has performed similar to decision tree regression. This should be the least preferred model to be used. |

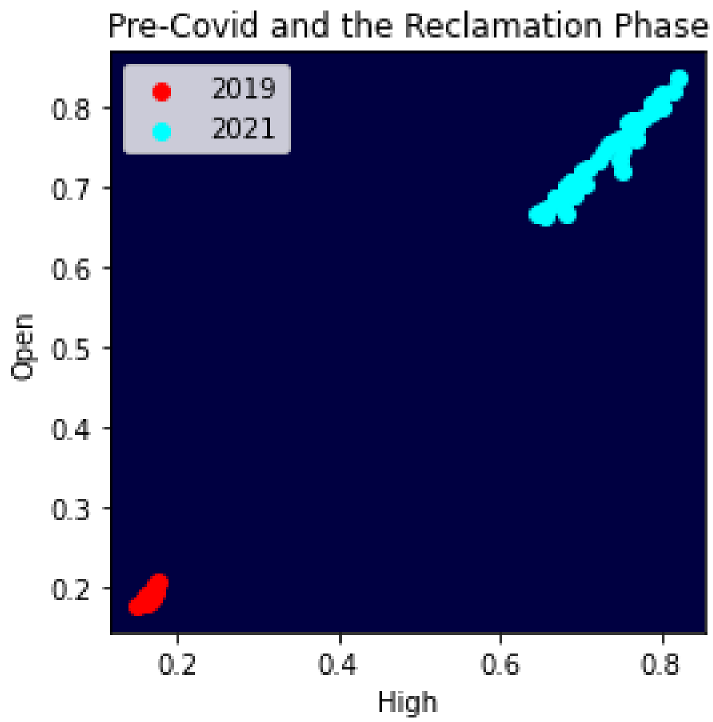
CIPLA



DR.REDDY




NIFTY PHARMA



We can clearly observe that there is a clear difference in the stock prices before and post the pandemic. We had a hypothesis that the stocks would have really progressed well after taking a 52 week low value in their prices and the COVID-19 pandemic had a direct influence on the stock's values , and several factors like governmental policies , vaccination status , the number of cases , the immigration of people might have influence the price of these stocks. If these three stocks are viewed together , we could conclude that they have performed very much identical to each other as expected. The dates in which the prices took a sharp turn were noted and it was expectedly the date where the company's had revealed about rolling out their new product which could address the concern of covid.

**7. References**