# CNNPred for Indian Stock Index

For this task of using CNNpred to predict Indian Stock Index, NIFTY50 Index was chosen. Open, High, Low, Close, Adjusted Close and Volume Data for the period between **2007-9-18** and **2022-7-27** was collected from NSE website.

Apart from 6 original features, 6 others were engineered based on popular technical indicators, namely ‘3 Days Moving Average’, ‘5 Days Moving Average’, ‘15 Days Moving Average’, ‘30 Days Moving Average’, Daily Trading Volume Difference and Weekly Difference in Closing Values.

Following hyperparameters were varied to find the optimal model-

1. Loss Function (mae, binary\_focal\_crossentropy, binary\_crossentropy, hinge)
2. Optimizer (SGD, Adam, Adagrad, Adamax),
3. Epochs (20, 30),
4. Batch Size (64, 32),
5. Dropout Rate (0.05, 0.1, 0.15, 0.2),

The whole experiment took takes 12h+ of training time despite using GPU. And yielded the following results.

CSV containing performance parameters for 64 possible combinations of above hyperparameters- <https://github.com/mandalnilabja/soc2022/blob/main/CNNpred_NIFTY50_performances.csv>

## Ensemble- Bagging

Bagging ensemble on 2dCNNpredNIFTY50 implementation using all 64 previous models significantly outperformed any single predictor. A hard voting strategy was manually implemented since each model by itself is a weak classifier.

Its evaluation metrics were:

Accuracy: 54.51%

Mean Absolute Error: 45.49%

Macro Averaged F1 Score: 70.50%

Compared to 2dCNNpred implementation on Wall Street Indices, even much fewer numbers of features were available and used, overall performance was evaluated to be better on NIFTY50 CNNpred.

Data Source:

<https://www1.nseindia.com/products/content/equities/indices/historical_index_data.htm>

Models:

<https://github.com/mandalnilabja/soc2022/tree/main/model>

Github Link to Notebook:

<https://github.com/mandalnilabja/soc2022/blob/main/Week10Assignment.ipynb>

<https://github.com/mandalnilabja/soc2022/blob/main/Week10AssignmentB.ipynb>

# Model Deployment using Flask API

Running the app.py initiates the Data Pipeline function that starts a locally hosted webpage using a html file index.html. The templates folder contains index.html which handles frontend for input button, flask requests to fetch file and creates a form for the prediction output.

User inputs NIFTY50 OHLCV data as csv (Use **NSEI.csv** for demo). The Next day’s movement of the stock index is predicted. If the index is predicted to increase then Output is “Up” and likewise for “Down”.

app.py: <https://github.com/mandalnilabja/soc2022/blob/main/app.py>

Templates folder: <https://github.com/mandalnilabja/soc2022/tree/main/templates>