In project3 folder

<https://github.com/Kulbear/stock-prediction.git>

<https://github.com/huseinzol05/Stock-Prediction-Models.git>

<https://github.com/kartik-joshi/Stock-predection.git>

<https://github.com/Rajat-dhyani/Stock-Price-Predictor.git>

<https://github.com/BenjiKCF/Neural-Net-with-Financial-Time-Series-Data.git>

<https://github.com/rosdyana/CNN-Financial-Data.git>

<https://github.com/rosdyana/Going-Deeper-with-Convolutional-Neural-Network-for-Stock-Market-Prediction.git>

<https://github.com/RajatSablok/EmoStock.git>

<https://github.com/eytanohana/LSTM-for-stock-price-prediction.git>

not in

<https://github.com/dark-0ne/StockPredictor.git>

<https://towardsdatascience.com/aifortrading-2edd6fac689d>

might be good, already folder of same name.

<https://github.com/UWFlex/stock-prediction.git>

<https://www.kaggle.com/magichanics/amateur-hour-using-headlines-to-predict-stocks>

<https://github.com/jangarong/StockMarketPredictions.git>

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0212320#sec002>

Output

Dense

LSTM

CNN

Input

CNN for feature extraction, LSTM for interpreting features across time steps.

Single CNN layer 🡪 sequence of LSTM models (one for each time step)

- wrap CNN model (n layers) within TimeDistribution layer

|  |  |
| --- | --- |
| 2  3  4  5  6  7  8 | model = Sequential()  # define CNN model  model.add(TimeDistributed(Conv2D(...))  model.add(TimeDistributed(MaxPooling2D(...)))  model.add(TimeDistributed(Flatten()))  # define LSTM model  model.add(LSTM(...))  model.add(Dense(...)) |

Use LSTM to learn long term relationships of features extracted with CNN

…best model with linear layer before and after LSTM