

Problem: The goal of this project is to predict whether to buy or sell Solana (SOL) cryptocurrency at a certain price point to maximize profit. By leveraging time series analysis and deep learning models, we aim to create a robust prediction system that can assist in making profitable trading decisions. Crypto currency trading is highly volatile and unpredictable. The problem with time series analysis on things such as stocks, foreign exchange, crypto is that predicting a price for the future would be near impossible if you want to achieve high accuracy. So instead, what if we make it a classification problem instead of a regression problem. How about instead of trying to predict what the stock price will be, the challenge is to develop a predictive model that can analyze historical price data of Solana and make accurate buy/sell Recommendations. That is what we try to do here.

Approach:

1. The approach was first downloading 4 years of cryptocurrency data from yahoo finance. After that, there wasn't any data missing, so there wasn't much data wrangling besides making sure that the data was not missing.
2. Now the only thing that was missing is the labels, and how to determine whether at that price point it was a buy or a sell label
 - a. The function would label each instance of our dataset, which would calculate whether the price would increase up to at least 5% within the next 5 to 30 days as buy, and a minimum decrease of 5% to the 5 to 30 days as sell.
3. After getting our labels, it was time to do some modeling, and I started off with the ARIMA model. For this one, I didn't do any classification, but forecasting actual values to see how it would perform
 - a. Performed Dicky-Fuller test to see if it was stationary
 - b. It was not so calculate the difference between each data point to make it stationary
 - c. Plotted the autocorrelation graph to see if past points would correlate
 - d. Then I fitted and trained the model and plotted the forecast
4. After testing that model, I decided to go with the LSTM. And I decided to add multiple LSTM layers to have a more deeper model, as well as adding Dropout as a loss function to reduce overfitting, and batch normalization in between each layer to keep the normalization in between each layer, with ReLu activation functions. I went with the Adam optimizer because it is the best in my opinion, and softmax activation for the output layer because it is classification
 - a. I first normalized all the values
 - b. I then created sequences into an np.array as my X and y
 - c. I then set up the Long Short-Term Memory Neural Network

Results:

What I found with the ARIMA model is that it had forecasted the mean to be wrong, but not far off. The mean squared error was ridiculously high at a value of 6512, which is strange given SOL's peak price is around 200 dollars. However, when printing the confidence interval of what the forecast is believed to be around, it matches the actual graph accurately. So my

interpretation is that this model may be bad for predicting short term trades, but good for predicting long term trades.

The accuracy after training the LSTM was around 44%. However this was at 2000 epochs. At a certain point from 1-400 epochs the average testing accuracy was 60%. Which is really good because you only need to be above 50% to have a profitable and working model. Despite the testing validation being so low, the training accuracy kept increasing which is a good sign meaning that it is learnable, however the model does not do well against generability, but could have potential at lower epochs.

Potential Research & Recommendations:

Some potential research that I could do is potentially improving the model to have better generality, possibly having a more detailed model with more data from other crypto currencies to possibly have a more accurate model. Maybe instead of having data of cryptocurrency prices per day, I can do per hour or per minute to get a even short term model for high frequency trading

1. My first recommendation is that if you were to use my ARIMA model, I would suggest only using it for long term trades if you are going to invest any money in SOL
2. My second recommendation if you are to use my dataset and my LSTM NN model, I suggest training it up to 500 epochs max, and this would best be for short term trading to find out when to sell and when to buy
3. My third recommendation is that through my findings, the training validation reached accuracy of 95%, and it will only go up meaning that the price movements for Solana is predictable, meaning there is hope for a more accurate model