Investment and Trading Project

Domain Background

This is an attempt to use information about past stock prices to predict future prices in the investment and trading domain. While there are multiple factors that can affect stock pricing – economic, political, social etc., we will use only the past stock prices themselves to solve a classification problem (buy or sell) or a regression problem (close price). My main motivation for this project is to learn how to effectively solve a time-series problem with a LSTM as the neural network. This is related to a side-project of mine whose data i cannot use in this project for proprietary reasons.

Problem Statement

All traders do wish to be able to foretell the price of a particular stock as it amounts to significant wealth (or loss). There have been many attempts to use statistical analysis and machine learning to predict stock prices. In this project, I wish to use deep learning to solve this problem. Specifically, I would like to tackle this as a time series problem and use LSTM to do the prediction. LSTM has a unique ability to be able to maintain state across sequences and hence I believe it would be a good fit for this problem. The main questions I would like to answer are:

1. A classification problem to predict whether to buy or sell a stock based on N days of data
2. A regression problem to predict Adj Close of a stock based on N days of data.

Datasets and Inputs

I plan to use the data sets that is publicly available using Yahoo finance or Quandl. The input (features) to the model will be Open, Close, High, Low prices and the volume on a per-day basis.

Solution Statement

For my current problem, I will end up with 2 models fed with similar inputs that will predict a class or a numeric value. In the classification problem, we will predict if to buy or sell. The buy or sell decision will be made based on the difference of Adj close price from the previous day. In the regression problem, I will train the model with Adj Close price as output. Subsequently, the model will attempt to predict the Adj close prices. [I can also try to make it more granular i.e. hourly or minute-by-minute may be applicable for an intra-day trader].

Benchmark Model

For the classification problem, i plan to use the random_classifier and set classifier to use most_frequent. Considering it is a time-series problem, i plan to predict the same value (rounded to closest decimal) as previous day as the baseline model for the regression problem.

Evaluation Metrics

Since this is a time-series problem, i want to ensure that the train and test data are represented as-is i.e. without shuffling. Hence, I will use a train/test split of 70/30. Since we want to evaluate on future values, the train split will contain the older 70% of the data. Accuracy would be a decent metric for the classification problem since there

would be sufficient labels on both classes in the training sample. I plan to use Root Mean Square Error (RMSE) as the metric for the regression problem. To ensure apple-to-apple comparision, we will use the same metric on both the benchmark and the solution model.

Project Design

This is the work flow i plan to use as part of the Project implementation

1. Data Preprocessing:

The input data has the following columns: Ticker, Date, Open, High, Low, Close, Volume, Adj Close. To convert it into a classification problem, we can add another column with a Buy/Sell label based on the previous day's close i.e. if the Adj Close on on day2 is higher than day1, we suggest a buy on day1. We can also use a moving average for 7 days into the future to predict if to buy or sell. Hence the inputs to the classifier is Open, High, Low, Close, Volume, Label for a particular ticker. For the regression problem, we will use Open, High, Low, Close, Volume as input and Adj Close as output.

2. Visualize data:

I hope to create some plots such as scatter plots to try to understand any relationship between the different features. A description of my input data set will also help better understand the stats underlying it.

3. Implement Baseline model

Will implement a baseline model using the ideas outlined earlier. This should help me evaluate the final solution model on its performance.

4. Train a Model

The data will then be re-scaled to a (0-1) range before passing it to a model. I plan to build both models using LSTM. Hence, i have to first define the input_dim (batch_size, timesteps, features). The LSTM layer will be followed by a dense layer. I may add dropout to reduce overfitting. I plan to use "mae" as the loss function and the "adam" optimizer to compile the model.

5. Evaluate the model

Model evaluation is primarily to understand if the solution model is overfitting or underfitting to the dataset. I would graph the runs for both the training and test data to understand the model characteristic. I may then have to iterate by tuning the model (its layers, hyperparameters etc) to ensure that it is suitable for prediction.

6. Prediction

We are now ready to use the model to predict if the stock is a buy or a sell and what can be the Adj close price. For the regression problem, we will need to rescale the output to the actual input values. We can now compare our solution against the baseline model

7. Improvement options:

Use a sliding window of past values to make a future prediction rather than on a day-on-day basis Use K-means to understand which features stand out best for predictions. Possibly use only the Principal components to improve accuracy.

I do want to create a frontend for the solution where a user can provide a stock ticker and a date-range. This info can be provided via a REST API and the server end would download the necessary data to train a model that can be used to give a prediction. I will give it a shot only if I don't have time/resource constraints. My main objective is to be able to create a decent model based on my learnings from the course.