Capstone Proposal

December 10, 2018

1 Capstone Proposal

1.1 ## Proposal

1.1.1 Abstract

The project proposal is to create a deeep learning model that predicts stock prices accurately using news about the stocks. My primary objective will be in top 10% of kagggle competation from 2 sigma

1.1.2 Domain Background

Predicting stock prices has been of interest for a large number of ML enthusiasts for a long time. There has been never a better time to do this, with availability for large data, news feeds and extremely large computers available to do the computation. There are two kinds of data available for this competation. 1. Market Data (2007 to 2018), which is provided by Intrino contains financial information such as OHLC, cumulated returns, trading volume etc. 2. News data (2007 to 2018) is provided by Thomson Reuters contains information about news articles, sentiment and commentry.

1.1.3 Problem Statement

The problem is forcasting stock prices. We get historical stock and news data from 2007 to present. We need to predict a signed confidence value

$$\hat{y}_{ti} \in [-1, 1]$$

which is multiplied by a market adjusted return of a given asset over a ten day window. If the algorithm thinks that the stock will go up and it is very confident, it will return a value close to 1.

1.1.4 Dataset and Inputs

It has two kinds of data. 1. Market data 1. News Data

Market Data Market data has following information * time(datetime64[ns, UTC]) - the current time (in marketdata, all rows are taken at 22:00 UTC) * assetCode(object) - a unique id of an asset * assetName(category) - the name that corresponds to a group of assetCodes. These may be "Unknown" if the corresponding assetCode does not have any rows in the news data. * universe(float64) - a boolean indicating whether or not the instrument on that day will be included

in scoring. This value is not provided outside of the training data time period. The trading universe on a given date is the set of instruments that are avilable for trading (the scoring function will not consider instruments that are not in the trading universe). The trading universe changes daily. * volume(float64) - trading volume in shares for the day * close(float64) - the close price for the day (not adjusted for splits or dividends) * open(float64) - the open price for the day (not adjusted for splits or dividends) * returnsClosePrevRaw1(float64) - see returns explanation above * returnsOpenPrevRaw1(float64) - see returns explanation above * returnsClosePrevMktres1(float64) - see returns explanation above * returnsOpenPrevMktres1(float64) - see returns explanation above * returnsClosePrevRaw10(float64) - see returns explanation above * returnsOpenPrevRaw10(float64) - see returns explanation above * returnsClosePrevMktres10(float64) - see returns explanation above * returnsOpenPrevMktres10(float64) - see returns explanation above * returnsOpenNextMktres10(float64) - 10 day, market-residualized return. This is the target variable used in competition scoring. The market data has been filtered such that *returnsOpenNextMktres10 is always not null.

News data

- time(datetime64[ns, UTC]) - UTC timestamp showing when the data was available on the feed (second precision)
- sourceTimestamp(datetime64[ns, UTC]) - UTC timestamp of this news item when it was created
- firstCreated(datetime64[ns, UTC]) - UTC timestamp for the first version of the item
- sourceId(object) - an Id for each news item
- headline(object) - the item's headline
- urgency(int8) - differentiates story types (1: alert, 3: article)
- takeSequence(int16) - the take sequence number of the news item, starting at 1. For a given story, alerts and articles have separate sequences.
- provider(category) - identifier for the organization which provided the news item (e.g. RTRS for Reuters News, BSW for Business Wire)
- subjects(category) - topic codes and company identifiers that relate to this news item. Topic codes describe the news item's subject matter. These can cover asset classes, geographies, events, industries/sectors, and other types.
- audiences(category) - identifies which desktop news product(s) the news item belongs to. They are typically tailored to specific audiences. (e.g. "M" for Money International News Service and "FB" for French General News Service)
- bodySize(int32) - the size of the current version of the story body in characters
- companyCount(int8) - the number of companies explicitly listed in the news item in the subjects field
- headlineTag(object) - the Thomson Reuters headline tag for the news item
- marketCommentary(bool) - boolean indicator that the item is discussing general market conditions, such as "After the Bell" summaries
- sentenceCount(int16) - the total number of sentences in the news item. Can be used in conjunction with firstMentionSentence to determine the relative position of the first mention in the item.
- wordCount(int32) - the total number of lexical tokens (words and punctuation) in the news item
- assetCodes(category) - list of assets mentioned in the item
- assetName(category) - name of the asset

- firstMentionSentence(int16) - the first sentence, starting with the headline, in which the scored asset is mentioned. 1: headline
  1: first sentence of the story body
  1: second sentence of the body, etc
  0: the asset being scored was not found in the news item's headline or body text. As a result, the entire news item's text (headline + body) will be used to determine the sentiment score.

- relevance(float32) - a decimal number indicating the relevance of the news item to the asset. It ranges from 0 to 1. If the asset is mentioned in the headline, the relevance is set to 1. When the item is an alert (urgency == 1), relevance should be gauged by firstMentionSentence instead.
- sentimentClass(int8) - indicates the predominant sentiment class for this news item with respect to the asset. The indicated class is the one with the highest probability.
- sentimentNegative(float32) - probability that the sentiment of the news item was negative for the asset
- sentimentNeutral(float32) - probability that the sentiment of the news item was neutral for the asset
- sentimentPositive(float32) - probability that the sentiment of the news item was positive for the asset
- sentimentWordCount(int32) - the number of lexical tokens in the sections of the item text that are deemed relevant to the asset. This can be used in conjunction with wordCount to determine the proportion of the news item discussing the asset.
- noveltyCount12H(int16) - The 12 hour novelty of the content within a news item on a particular asset. It is calculated by comparing it with the asset-specific text over a cache of previous news items that contain the asset.
- noveltyCount24H(int16) - same as above, but for 24 hours
- noveltyCount3D(int16) - same as above, but for 3 days
- noveltyCount5D(int16) - same as above, but for 5 days
- noveltyCount7D(int16) - same as above, but for 7 days
- volumeCounts12H(int16) - the 12 hour volume of news for each asset. A cache of previous news items is maintained and the number of news items that mention the asset within each of five historical periods is calculated.
- volumeCounts24H(int16) - same as above, but for 24 hours
- volumeCounts3D(int16) - same as above, but for 3 days
- volumeCounts5D(int16) - same as above, but for 5 days
- volumeCounts7D(int16) - same as above, but for 7 days

1.1.5 Solution Statement

This is a unique problem involving tabular data, time series data and natural language data in the same problem. The stock values is influenced by various factors. We will process time series data with recurrent neural networks and process news data with natural language processing with deep neural networks. We will also convert catogorical values such as 'firstMentionSentence' into one hot encoded features. Currently I am still in experimental mode, so the final architecutre of the model will be submitted along with the project submission.

1.1.6 Benchmark model

This is a Kaggle competition and my intention if to be in top 10% on the leader board. A benchmark score to get there with current scenario is 0.67 which would put our model in top 10%. A benchmark model that can achieve this would use gradient boosting calssifier (XGBoost) and a lightgbm classifier and ensemble the results. It drops the natural language part of news and only collects the categorical part of news data. Merging news catagories like sentimentClass, relevence etc into market data like OLHCV it trains a gradient boosting algorithm to predict the probability of stock moving up next day.

1.1.7 Evaluation Metrics

Given the above predicted $ _{ti} $, the return will be calculated as

$$x_t = \sum_i \hat{y}_{ti} r_{ti} u_{ti}$$

Where,
r_{ii} is the 10 day market adjusted return for day t for the instrument i
u_{ti} is a variable that controls whether a particular asset is included in scoring on a particular day.
The score for submission, hence the performance of the model is decided by the standard deviation for the daily x_t values

$$score = \frac{\bar{x}_t}{\sigma(x_t)}$$

1.1.8 Project Design

The initial part of the project is to do a exploratory data analysis to understand the correlation between inputs, how news affects stock price etc. Then we will move on to use a RNN to predict returns from the historical data. Once that is stable, we will incorporate the news data into the model so that we can get the desired results.

Exploratory data analysis Exploratory data analysis would include 1. Finding abnormal drops and rises in market, for example collapse on Lehman Brothers in 2008. 2. Finding huge fluctuations caused by splits or possible data errors and eliminating them 3. Finding standard deviations of the above cleaned data. 4. Finding cumulative returns of the cleaned data. 5. Find bollinger bands and other indicators and add them to dataset. 6. Exploring news to understand how it affects stock prices.

Once I have all this data encorporated into our training data, I will move on to designing the model.

Language model for news data I will build a language model that can classify the sentiment in the news better than positive, negative and neutral using neural network. Initially I will build language model based on other data set like IMDB reviews and then tune it to work on the stock news dataset. Once I have a reasonable model to understand news, I will move on the build a LSTM for stock prediction.

LSTM model for stock data I will first add the results of news into the historical stock prices dataset as catagorical variable. Then I will create a LSTM network to predict the stock price. An LSTM model is a form of Recurrent Neural Network, that can keep track of long and short term behaviour of the input data. This input data will also have categorical view of news data. I will train LSTM model on the data provided and evaluate the accuracy.

1.1.9 References

1. Kaggle: https://www.kaggle.com/c/two-sigma-financial-news#evaluation
2. LSTM: https://www.datacamp.com/community/tutorials/lstm-python-stock-market