**Something need to be handled:**

* 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.

How to handle this?

* **Stock market operating time:**

Two import questions:

1. How to handle the discrete values
2. How to merge the news data and the market data

**Explanation of these features:**

**Within the marketdata, you will find the following columns:**

* time(datetime64[ns, UTC]) - the current time (in marketdata, all rows are taken at 22:00 UTC)

2498

**the time of the information**

* assetCode(object) - a unique id of an asset

3780

**Use this to represent the company**

* 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.

3511

**Use this to represent the company**

* 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.

2

**This is used to calculate the score**

* volume(float64) - trading volume in shares for the day

**Trade volume, this definitely influence the trend of this stock. (numerical)**

* close(float64) - the close price for the day (not adjusted for splits or dividends)

**Close price. This combining the open price will influence the trend of the stock (numerical)**

* open(float64) - the open price for the day (not adjusted for splits or dividends)

**Open price. This combining the open price will influence the trend of the stock (numerical)**

* returnsClosePrevRaw1(float64) - see returns explanation above

**previous 1 day, close – close.(raw) (numerical)**

* returnsOpenPrevRaw1(float64) - see returns explanation above

**previous 1 day, open – open.(raw) (numerical)**

* returnsClosePrevMktres1(float64) - see returns explanation above

**previous 1 day, close – close.(mktre) (numerical)**

* returnsOpenPrevMktres1(float64) - see returns explanation above

**previous 1 day, open – open. (mktre) (numerical)**

* returnsClosePrevRaw10(float64) - see returns explanation above

**previous 10 day, close – close.(raw) (numerical)**

* returnsOpenPrevRaw10(float64) - see returns explanation above

**previous 10 day, open – open.(raw) (numerical)**

* returnsClosePrevMktres10(float64) - see returns explanation above

**previous 10 day, close – close.(mktre) (numerical)**

* returnsOpenPrevMktres10(float64) - see returns explanation above

**previous 10 day, open – open. (mktre) (numerical)**

* 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.

**Next 10 day, open – open.(mktre) (numerical)**

News data

The news data contains information at both the news article level and asset level (in other words, the table is intentionally not normalized).

* 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

**In the above three version of time, I think we just need one**

* sourceId(object) - an Id for each news item

**How to use it?**

* headline(object) - the item's headline

**Some people just see the headline. You know. (String)**

* urgency(int8) - differentiates story types (1: alert, 3: article)

3

**This may influence the importance of the news.( numerical) (discrete)**

* takeSequence(int16) - the take sequence number of the news item, starting at 1. For a given story, alerts and articles have separate sequences.

97

**The sequence number of the sequence which hold this news item ( discrete)**

* provider(category) - identifier for the organization which provided the news item (e.g. RTRS for Reuters News, BSW for Business Wire)

30

**The provider’s name. This may reflect the popularity of this news item. (discrete)**

* 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.

**It is like a very simple summarize. (discrete)**

* 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)

I need to handle this. This is very important.

**This is very import. It covers the information about who would be influenced by this news.**

* bodySize(int32) - the size of the current version of the story body in characters

**A little. This may influence whether people are willing to read it.**

* companyCount(int8) - the number of companies explicitly listed in the news item in the subjects field

**This just summarize the subject field. (discrete)**

* headlineTag(object) - the Thomson Reuters headline tag for the news item

163

Useful. (discrete)

* marketCommentary(bool) - boolean indicator that the item is discussing general market conditions, such as "After the Bell" summaries

Useful.

* 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.

**A little. This may influence whether people are willing to read it.**

* wordCount(int32) - the total number of lexical tokens (words and punctuation) in the news item

A **little. This may influence whether people are willing to read it.**

* assetCodes(category) - list of assets mentioned in the item

**Useful**

* assetName(category) - name of the asset

**Useful**

* firstMentionSentence(int16) - the first sentence, starting with the headline, in which the scored asset is mentioned.
  + 1: headline
  + 2: first sentence of the story body
  + 3: 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.

**Kind of.**

* 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.

Very important.

* sentimentNegative(float32) - probability that the sentiment of the news item was negative for the asset

Very important.

* sentimentNeutral(float32) - probability that the sentiment of the news item was neutral for the asset

Very important.

* sentimentPositive(float32) - probability that the sentiment of the news item was positive for the asset

Very important.

* 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.

**Import**

* 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.

**Very import**

* noveltyCount24H(int16) - same as above, but for 24 hours

Very important.

* 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

discrete value:

market data:

AssetName AssetCode

News data:

sourceId(object) , headline(object), urgency, takeSequence(int16), provider(category),

subjects(category) audiences(category) headlineTag(object) marketCommentary(bool)

sentimentClass(int8)

What should we do?

from kaggle.competitions import twosigmanews

env = twosigmanews.make\_env()

(market\_train\_df, news\_train\_df) = env.get\_training\_data()

train\_my\_model(market\_train\_df, news\_train\_df)

for (market\_obs\_df, news\_obs\_df, predictions\_template\_df) in env.get\_prediction\_days():

predictions\_df = make\_my\_predictions(market\_obs\_df, news\_obs\_df, predictions\_template\_df)

env.predict(predictions\_df)

env.write\_submission\_file()

These two functions:

Train\_my\_model(),

Make\_my\_predictions();

First step:

Numerize the discrete value:

AssetCode: 3780

AssetName: 3511

Urgency: 3

Take Sequence: 97

Provider: 30

Subjects: how ?

Audience: how?

HeadlineTag: 163

Market Commentary: 2

Sentimental Class: 2

The parameter that I am gonna to use:

??

Data preprocess:

For market data:

Replace the possible error data with the normal data

Leave the data since 2010.

For news data: