Exissting & New Insider Product Requirements vs. 1.0

Drafted by: Mark LoPresti
4/19/2014

Issue: Security Type Linkage

- Currently our Insider content on the insider beacon page aggregates trades from insiders without deference to security type meaning bonds, different classes of common stock, etc are aggregated as one because ***we cannot link the Forms data to the Price data.***
- The reason we cannot link it is because security type is not standardized (natural language).
- Despite warnings & a specific Terms & Conditions legalese that protects from users, our risks from the reporting companies lingers.
- Result: we may not launch with this content in version 1.0

Opportunity

- Due to me be considered the "insider guru" Stansberry has already said that they would add this to the portfolio. This is a "perfect fit" for their subscribers.
- A former manager of mine, who works in a senior role at Standard & Poors that has authority and influence to add product & content, said that if we already had a new insider trading "ranking model" they would either license it from us or engage in a revenue share arrangement.

4 Hurdles

1) For production purposes we need a process that can link the security on each form record – at the minimum- from the Non-Derivative section to the Price data.

2) While former employers have a team in Bangalore working on this manually every week, our process for the most part must be automated.

3) For testing & creating a model I need at least a good representative amount of companies (S&P500 list?) that ARE linked to conduct my transformation of the data

4) For prototype and production purposes of a new ranking model product, we'll need a strong understanding of our performance. Previous interactions with this data has been met with horrible query performance impacting development.

What Was Done

- Mark created an R process that did 3 things:
  - Prepped the data for matching
  - Obtained some additional meta data that may be useful later
  - Added a process that found the most common "cleaned up" phrases and manually matched them and a process to build this list over time.
  - Matches the prepped data from a list of identified common phrases
  - Matches the remaining unmatched data using an algorithm

What Went Well

- The majority of the codes matched better than 98%
- Non-Derivative & Deriv performed fairly well overall with over 90% match success on an extensive QC efforts totalling 6,000 randomly selected records built from all match codes. Many codes > 99%. (see "Sec_Title_Stndrdztn_Results.pdf")
- A1, the matched-common-phrase assignment code, matched 97.4%

Improvements Needed For Product Dependencies

- While A1, common phrase matching, worked well, there still were some not being assigned correctly and many of those errors were the "common stock". This means a "loss" of key data or possibly a "single but important" key data point. For the record, the text entered was not "common stock" but could have been "cmn stk", "commstk", etc...
- Z1, the algorithm assignment code was wrong about 80% of the time. However, the other codes took care of the majority of the data and this assignment code was only implemented about 1.3% of the time

Explanation of "Sec_Title_Stndrdztn_Results.pdf"

- 2 vertical sections
  - Error rate summary by Assignment Code
  - Assignment code break down after historical process run: Deriv & Non-Deriv
- Error rate summary reveals Z1 (algorithm), which deals with the final unassigned items after stripping out the "easiest" onces was incorrect 80% of the time.
- Assuming error rates from QC sample, over 25,000 records could be incorrect; most of which is in Non-Derive (key data set for our products). I'm not sure to what that translates on a Daily exception count.

Explanation of SecurityTitleAnalysis_02272012.ods

- 2 tabs
  - Tab 1 = Pivot tables that are found in the "..Results.pdf"
  - Tab 2 = 3 columns: a) orignal b) assigned from process c) correct after QC
  - Tab 2 = the 500+ errors that were found in the 6,000 record QC

Improvement Suggestions

- One of the reasons for the errors was "begun" in clean up. Insider names (for trusts, custodial accounts, etc..) were included in the natural text entered in the Security Title section. I did not account for this when I wrote the code. So when Z1 was used it would attempt to match on irrelevant words such as the insider's name. Perform this check first (last name & first name), remove before deploying process. I do not speculate on how much improvement can be gained here from this specific run.
- Perhaps a review of the error file would reveal some rules that need to be added to both Deriv & Non-Deriv.
- Perhaps an additional QC of a greater # should be undertaken.
- Perhaps this can be converted to a completely SQL process?