Textural Analytics Scoring and Validation Project

Principal financial Natural Language Processing

Sri Ram Sagar Kappagantula – G01084203
Chandrika Amarkhed – G01099774
Avneet Pal Kour – G01094149
Paras Sethi – G01110919
Merin Joy - G01118158

Volgenau School of Engineering, George Mason University, 4400 University Dr, Fairfax, VA, 22030

Abstract

Principal Financial Group (PFG), a global investment management leader, offers class retirement services, insurance solutions, and asset management services to businesses, institutional investors and individuals preparing for the future. To achieve the company's mission, management needs to take various decisions as per financial statements, company records and market conditions. However, reading these large number of financial and non-financial documents is a painstaking task. This project aims to build a 'Textural Analytics Classifier' that classifies financial unclassified text to give meaningful insights that the management can use to make important decisions on growth, opportunity and strategy.

The project utilizes the data derived from financial sources like analyst notes, stock narratives, transcribed announcements, emails, etc. synthesized by Decooda, a text analytics service partner of Principal Financial Group. The annotated data received from Decooda is preprocessed, followed by training of 4 different models on 8 different financial word lists, to create the baselines. The best contributing features from each trained model are then selected using three python machine learning feature selection methods. These top features are then combined and trained to give the overall best features and the best results. The results are visualized using PowerBI dashboards.

Results achieved using various natural language and feature selection techniques demonstrate a direct relationship between features and labels. These classifiers and the various machine leaning methods used in this project can be used by PFG as a baseline to create their knowledge base system. This system will help them to convert raw data to information and further to knowledge, which will aid them to generate insights and make better decisions.

Key Words: Feature selection, Machine Learning, Logistic Regression, Naïve Bayes, Support Vector Machine.

1. Introduction

1.1 Background and Rationale

Principal Financial Group generates large amount of unstructured data from their management communications in large number of sources like analyst notes, stock narratives, transcript announcements, emails, etc. This data could have a lot of useful information when uncovered could give the company's management group an edge in decision making.

1.2 Research

To identify the different textual analytics techniques for financial data we went through a set of research papers like News versus Sentiment: Comparing Textual Processing Approaches for Predicting Stock Returns after reading we found that we can use financial dictionary of Loughran and McDonald (2011) for sentiment analysis of our data set and neural networks is most effective technique for topic analysis without having to use very large set of corpora.

In the Barron's Red Flags research paper the best performance was shown by Logistic Regression with 10-Ks from SEC EDGAR (1994-2008) dataset. In this research paper, Logistic Regression is used but we do not find any new unique algorithms.

Research papers	Datasets	Models	Helpful for Project
News versus Sentiment: Comparing Textual Processing Approaches for Predicting Stock Returns	900,000 news stories financial dictionary of Loughran and McDonald (2011)	Bag of words Neural Networks	We can use Financial dictionary of sentiment analysis and neural networks
Barron's Red Flags: Do They Actually Work?	10-Ks from SEC's EDGAR (1994-2008)	Logistic Regression	No unique algorithm useful for the project.
The impact of narrative disclosure readability on bond ratings and the cost of debt	Mergent Fixed Income Securities Database (FISD)	Logistic Regression, Ordinary Least Square Regression	Performing hypothesis tests, examining uncertain language (UNCERTAIN), forward looking statements (FLS) and net optimistic language (TONE) can give us more insights.
When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks	positive (Fin-Pos).,uncertainty (Fin-Unc); litigious (Fin-Lit).,strong modal words (MW-Strong)., weak modal words (MW-Weak).	Bag-of-words, likelihood ratio classifier, vector distances.	Word dictionaries to generate word vectors for the use of neural networks.

Fig. 1.2.1: Research papers

The next research paper we read was about the impact of narrative disclosure readability on bond ratings and cost of debt. The researchers dealt with Mergent Fixed Income Securities Database (FISD) and used models like Logistic Regression and Ordinary Least Square

Regression. The interesting insight in this research was from the word lists used. Wordlists for examining uncertain language, forward looking statements and net optimistic language to better understand the tone of the sentence. These wordlists can help us in achieving our goal and give us more insights.

The research on "When is Liability not a Liability? Textual Analytics dictionaries and 10-Ks" had used a much complex approach using neural networks. The creation of word dictionaries to generate word vectors is used as an approach in neural networks. Also, with the neural networks, bag of words, likelihood ratio classifier and vector distances are used to improve the efficiency of the model. This gives us an idea on how to improve the model efficiency effectively.

1.3 Project Objective

The objective of this project is to build a textual analytics classifier that classifies text related to growth, opportunity, strategy and stability correctly in order save Principal Financial Group the tedious task of manually extracting textual data from multiple sources, interpreting text, and extracting information from it. Capturing how often the management discusses topics like opportunity, strategy, growth. Identifying management teams who frequently address the topic would help identify good management teams within the company.

2. Data Acquisition

2.1 Data Preparation

Before data could be used for analysis, it needs to be annotated. Principle Financial Group partnered with Decooda, their text analytics service research partner synthesizes data from various sources such as analytics notes, stock narratives, social media, news, blogs, transcribed announcements, websites, email messages, feeds it into a Knowledge Transfer Module (KTM), the Cognitive Analytics engine where data is annotated for its relevance and sentiments.

KTM presents the user with a statement and the user has four options to tag the statement - extremely relevant, very relevant, relevant and not-relevant. If the user selects the relevance of the statement as relevant, next step is selecting the sentiment of the statement - positive, negative or neutral. Text from three categories of financial management topics, growth, opportunity and strategy are tagged using KTM.

2.2 Data Description

Data from three categories of financial management topics were obtained after annotation of unstructured text using KTM. The first dataset was the growth data that consisted of 2224 rows and 31 columns, where column 1 is the given text and the next 30 columns are the important features derived by PFG. The second dataset was for the topic opportunity that consisted of 3153 rows and 31 columns, where column 1 is the text and 30 columns are the

features derived by the client and the third dataset for strategy consisted of 3232 rows and 32 columns, where column 1 is the given text and next 31 columns are the derived features from the client.

2.3 Data Filtering

We had four different types of relevance - relevant, extremely relevant, very relevant and not relevant. Similarly, we had three different types of sentiments i.e. positive, negative and neutral. The data we received had the percentages for different types of relevance and sentiments. The percentage described proportion of people choosing the particular option. Hence, the first step was choosing the final label for relevance and sentiment.

Filtering out the relevant sentences by allocating weights according to relevance. highest weights to Extremely Relevant having the highest weights to Relevant having the lowest weights. If same percent is marked, then choosing the response as per weights. Filtering the sentiment of relevant sentences by assigning weights. Positive is given the value as +1.Neutral is given as 0.Negative is given as -1. If same percentage is marked, then choosing the sentiment by adding the values. For example, if the same percent is marked as negative and neutral, so final sentiment will be (-1+0=-1) i.e. Negative.

2.4 Data Pre-Processing

Various techniques were utilized to pre-process the data. This included dealing with the special symbols and punctuations such as emojis and emoticons, dealing with the contractions, converting to lower case, removal of stop words followed by lemmatization. Dataset consisted of punctuations, special characters and repeated comments.

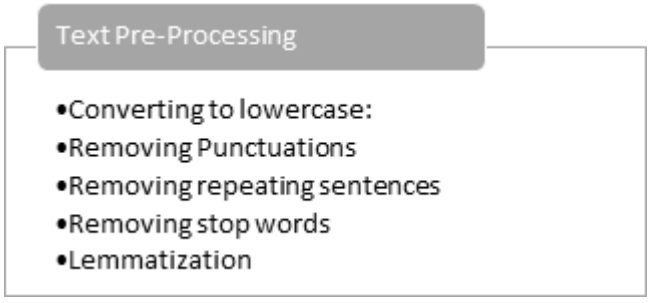


Fig: 2.4.1

In the datasets, we have data in mixed cases but models like wordnet are case sensitive i.e. they treat same word differently if the case for words are different. Hence, case needs to be changed to either lowercase or uppercase.

The data acquired was clean and structured, but our goal is to design models that can work on unstructured data such as tweets. Most of the tweets consists of special symbols such as emojis and emoticons and people write most of the sentences with contractions. It is important to deal with special symbols and contractions, so the data is reprocessed in such a way that it replaces the emojis and emoticons with a related word and expands the contractions. Apart from

the special symbols and contractions, data also contained lot of symbols like /, +, ~ etc. Hence removing the remaining punctuations was an important part of the cleaning process.

The dataset also consisted of words that don't have any meaning like - 'the', 'is', etc. Having lot of stop-words can also impact performance negatively, so removing them was also important step in the data pre-processing. Words like 'not', 'nor', 'up' and 'down' are important for sentiment analysis so we did not remove such stop-words. Also, abbreviations were not removed as dataset contained various financial abbreviations that could be useful while prediction. Important financial abbreviations were replaced with their full-forms.

In the next step we choose between stemming and lemmatization. In stemming, model learns similar representations for words of the same stem when the data suggests it and in such a process, meaning of many words are lost. Hence, we used lemmatization instead of stemming.

3. Modeling

3.1 Introducing New Features

After the data pre-processing step, few insights were drawn from the research papers related to our topic of research. Addition of new features to the datasets along with the features shared by the client should increase the model performance substantially. The new features obtained are wordlists used in the referred research papers that were successful in achieving the desired accuracy and precision. Making use of these features would help us in analyzing the datasets and increase the model performance. Hence, below are the eight wordlists that we implemented in our project.

1. **Client wordlist** – Using the word list provided by Principal Financial Group.
2. **Tagme word list** – Using the word lists from Tagme API wrapper for Python.
3. **Bag of words wordlist** – Using the bag of words method in Python.
4. **Wordnet wordlist** – Using the an NLTK corpus reader called WordNet.

Loughran – Mc Donald Sentiment word lists – Using the list provided by research papers.

5. **Litigious wordlist**
6. **Positive wordlist**
7. **Negative wordlist**
8. **Uncertain wordlist**

3.2 Evaluation Metrics

Evaluation metrics refers to the measures used in determining the model performance. In this project the metrics used for comparison and correlation between the models are as follows:

Metric	Formula and Description
True Positive Rates (TPR)	*TPR = TP / (TP + FN)*
False Positive Rates (FPR)	*FPR = FP / (FP + TN)*
Precision	*Precision = TP / (TP + FP)*
Recall	*Recall = TP / (TP + FN)*
F-Measure	*F-Measure = 2TP / (2TP + FP + FN)*
Accuracy	*Accuracy = (TP + TN) / (TP + TN + FP + FN)*

Fig. 3.2.1 Evaluation Metrics

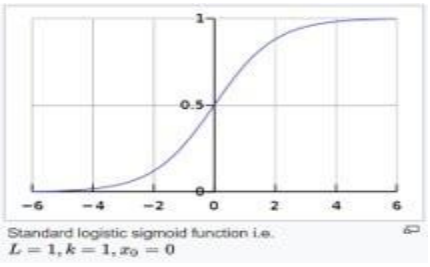
Also, the use of a Confusion matrix helps in identifying precision and recall in a much wider range. The factor that which of the two, precision or recall should be considered much case in this scenario is a factor to think about. A Type 2 error in this case states that it fails to reject the null hypothesis by stating that the sentence is not relevant when it is actually relevant. A Type 1 error rejects the null hypothesis by stating that the sentence is relevant when it is not. In this case, a Type 1 error or false positives are much worse that false negatives. Hence, we lookout for higher precision than recall.

3.3 Models

The baseline models - Logistic Regression, Random Forest, Naïve Bayes and Support Vector Machines are selected to perform modeling using the eight different lists. Evaluation metrics for each model is obtained and compared among each other.

1. **Logistic Regression**

A logistic function or logistic curve is a "S"(sigmoid curve) shape curve, with equation:



$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}}$$

Fig: 3.3.1

Logistic functions are used in logistic regression to model how the probability *p* of an event may be affected by one or more explanatory variables:

$$p = f(a + bx)$$

an example would be to have the model where x is the explanatory variable and a and b are model parameters to be fitted and f is the standard logistic function.

2. Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

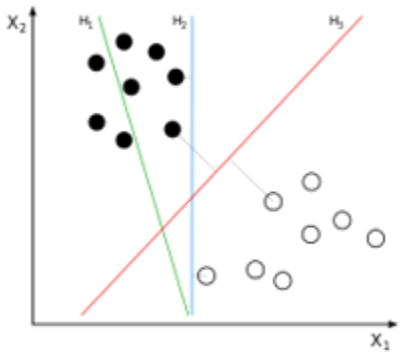


Fig: 3.3.2

3. Random Forest

Random forest is an ensemble learning method using multiple learning algorithms to obtain better predictive performance. Multiple decision trees use the Random subspace method to obtain the combined results from the decision trees. A series of decisions are made to segment the data into homogeneous subgroups. This is also called recursive partitioning. When drawn out graphically, the model can resemble a tree with branches.

A decision tree is comprised of nodes and splits of the data. The tree starts with all training data residing in the first node.An initial split is made using a predictor variable, segmenting the data into 2 or more child nodes. Splits can then be made from the child nodes. A terminal node is one where no more splits are made. Predictions are made based on the make-up of terminal nodes.

4. Naïve Bayes

This is a popular technique to classify text and documents based on a category (whether to classify a document as Relevant or not relevant based on the occurrence of certain words). It is a simple way to assign class or category labels to instances or cases.

Rather than being a single distinct algorithm, it is a set of algorithms that work on one underlying principle -- “the value of a given feature is independent of the value of any other feature”.

3.3.1 Baseline Results

The 4 models used are trained individually with the 8 word lists that were derived. Evaluation metrics for each model is obtained and compared among each other. The best accuracies achieved for each of the datasets are as follows:

1. **Growth Data**
   - Best Accuracy of 73.77% for relevance was achieved using Random Forest.
   - Best Accuracy of 68.06% for Sentiments was achieved using Logistic Regression.
   - Naive Bayes given least results for sentiments.

2. **Opportunity Data**
   - Best Accuracy of 77.06% for relevance was achieved using Random Forest.
   - Best Accuracy of 70.72% for Sentiments was achieved using Logistic Regression.
   - Naive Bayes given least results for sentiments.

3. **Strategy Data**
   - Best Accuracy of 92.27% for relevance was achieved using SVM.
   - Maximum of 60% accuracy was achieved on sentiments.
   - Naive Bayes given least results for sentiments.

4. **Stability Data**
   - Best Accuracy of 97.13% for relevance was achieved using Logistic Regression.
   - Maximum of 50% accuracy was achieved on sentiments.

3.4 Feature Selection

In order to increase the performance of the models, few feature selection techniques were applied as follows:

- **Mutual_info_classif:** Measures the mutual dependency between the variables. Estimate mutual information for a discrete target variable.
- **Chi-square:** Measures dependency between stochastic measures. “Weeds out” the features that are the most likely to be independent of class and therefore irrelevant for classification. Compute chi-squared stats between each non-negative feature and class.
- **SelectFromModel:** Meta -transformer for selecting features based on importance weights.

The above-mentioned feature selection methods were used to improve the model efficiency. From the eight wordlists, each model was run to obtain top attributes using the above-mentioned feature selection techniques. By looping over the top attributes, we select the best attributes using AUC values.

Below figure 3.4.1 is a plot with all ROC curves for bag of words with different number of features. AUC value is also given for each curve. And we see that AUC value is .811 for 3600 features as well as for 2600 features so we can choose from either 2600 or 3600 number of features depending on how much accuracy we get. In this case we went with dimensionality reduction and choose 2600 top attributes.

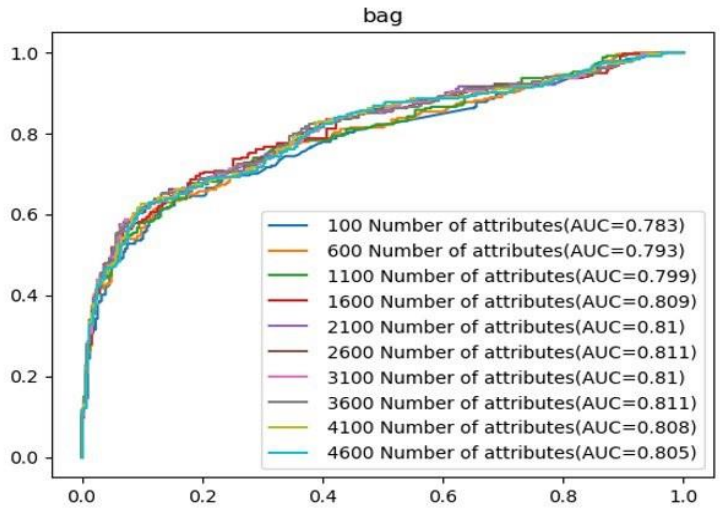


Fig. 3.4.1: ROC Example Based on Chi-Squared Values

3.4.1 Model Results after Feature Selection

1. **Growth Data**
   - Best Accuracy of 76.92% for relevance was achieved using Logistic Regression.
   - Best feature selection was given by mutual_Info_Classif.

2. **Opportunity Data**
   - Best Accuracy of 77.56% for relevance was achieved using Random Forest.
   - Best feature selection was given by SelectFromModel.

3. **Strategy Data**
   - Best Accuracy of 91.95% for relevance was achieved using Random Forest.
   - Best feature selection was given by SelectFromModel.

4. **Stability Data**
   - Best Accuracy of 95.10% for relevance was achieved using Random Forest.
   - Best feature selection was given by Mutual_Info_Classif.

4. Visualization

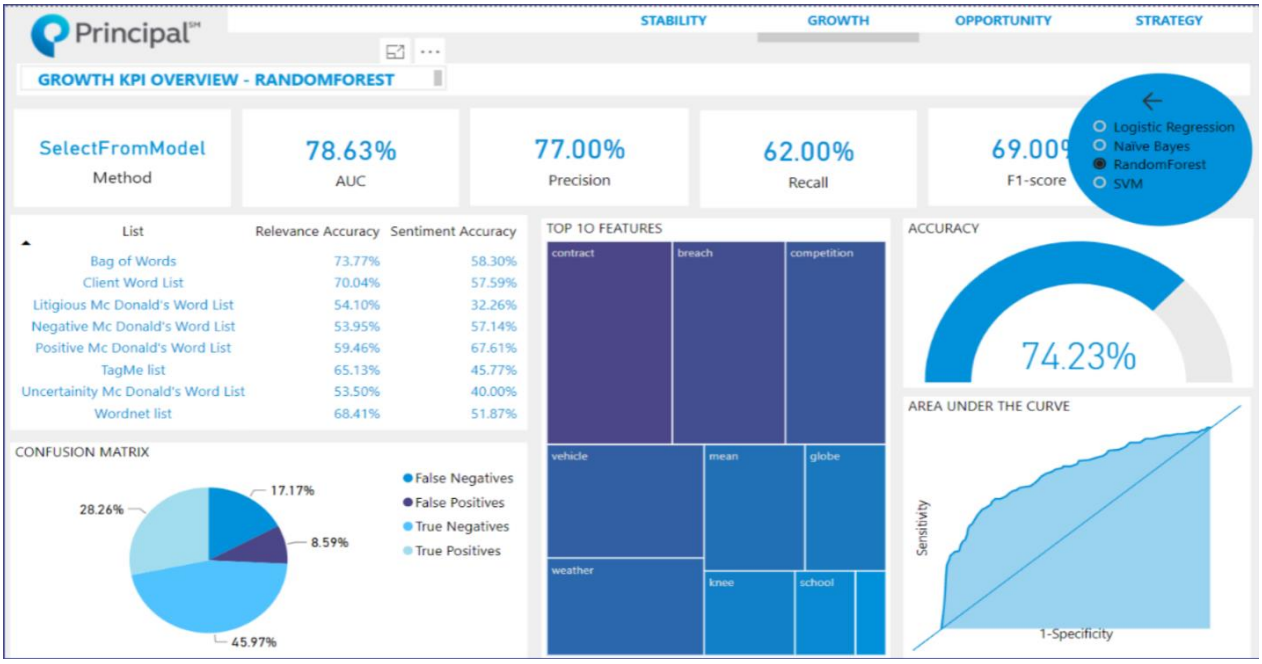


Fig. 4.1: Model results using PowerBI dashboard

The Key Performance Indicators (KPI) for growth, opportunity, strategy and stability datasets are shown using PowerBI dashboards. As discussed earlier in the paper, various evaluation metrics are used to determine the model performance and efficiency.

Different types of charts are used to give meaning to the model results and at the same time provide meaningful visualizations. Cards in PowerBI is used to display AUC, Precision, Recall and F1-Score. Confusion Matrix plays an important role in identifying false positives and false negatives and also, giving us an insight in what happens with the dataset. It is represented using a pie chart. Top 10 features from every model is shown using tree maps showing top attributes from every model using the feature selection techniques. A gauge chart is represented for accuracy as it displays how much more accuracy need to be achieved. ROC curves is an important plot for a classification model and hence is shown as it is using an area chart of how much the area under the curve constitutes. Also, ROC curves tell us how much a model has learned depending on flatness of the curve as it goes up. Baseline accuracies are explained using a simple table for eight different lists.

Some special features used in these dashboards are, menu button to toggle between different models that was created using selection and bookmarks pane. Also, actions were used to toggle between the four different dashboards representing the four datasets.

5. Risks and Assumptions

The accuracy of the classifier proposed in this paper could be hindered by two major factors: Insufficient data and data quality. Since the data generated after annotation was small, the models are trained on a limited training set. Data quality could be very crucial as the model relies on the quality of data it trains on. Since the data was annotated manually by a limited set of people, there could be a possible bias created based on the users' limited domain knowledge in financial management.

6. Conclusion

Including feature selection improves model results considerably compared to the baseline models. In this case, feature selection is also responsible for dimensionality reduction of more than thousands of features.

Also, by using lists like Word net and Bag of Words we were able to increase the interoperability of our models. These models are not limited to a specific data set anymore.

Results achieved using various natural language and feature selection techniques demonstrate a direct relationship between features and labels.

We can also say that in most cases, Random Forest model performs better compared to other modeling algorithms.

7. Future Work

Models could be trained and tested with much larger datasets as the data available after annotation was not large and diverse enough. Word embeddings such as glove and word2vec can be used for feature engineering. Also, neural networks and advanced deep learning models can be implemented.

References

[1] Steven L. Heston, and Nitish Ranjan Sinha, April 23, 2014," News versus Sentiment: Comparing Textual Processing Approaches for Predicting Stock Returns."
[2] Samuel B. Bonsall IV & Brian P. Miller, 20 March 2017," The impact of narrative disclosure readability on bond ratings and the cost of debt."
[3] TIM LOUGHRAN and BILL MCDONALD April 23, 2014," When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks."

Appendix A

Code reference : https://github.com/sriram161/DAEN690_NLP.git