# Business Problem

The principle objective of Brand Reputation tracker is to benchmark ASI and its peers across various industry themes and understand the key topics driving the buzz in various social and financial channels.

To analyze impact of marketing efforts and identify key ideas being associated with the brand on a regular basis. To deep-dive into specific areas of interest to business and provide relevant insights.

# Data

## 2.1 Data Extraction Prerequisites:

Keywords are generated based on the secondary research performed on the ASI and its peer group, which are given as input to a data aggregator (Sysomos) to extract the required data.

For Example: The following keywords were used to reference ASI consistently across channels (Twitter, News Blogs):

*@ASInvestmentsSE‏, @ASInvestmentsNO , @SLA\_plc ‏, @ASInvestmentsAE‏, @ASInvestmentsUK, @ASInvestments‏, @ASInvestmentsCH‏, @ASInvestmentsAU‏, Aberdeen Standard Investment, AberdeenStandardInvestment, Aberdeen Asset Management/Standard Life, Aberdeen Standard Life, Standard Life Investment, Aberdeen Asset Management, StandardLifeAberdeen, Standard Life Aberdeen, AberdeenStandardInvestment, aberdeen standard investments, aberdeen standard, aberdeen standard life, AberdeenStandardLife, StandardLifeInvestment, AberdeenAssetManagement, aberdeen asset management, aberdeen asset manager, aberdeen asset mgmt, aberdeen asset, standard life investment, standardlifeinvestment, standardlifeinvestment.com, @sli\_global*

The keywords used for the extraction are in the file below:



## 2.2 Data Extraction:

Once a set of keywords were finalized, a Data Aggregator (Sysomos) was used to scrape the data. Sysomos was chosen over other aggregators (Sprinklr, Social Studio) due to the quality(low on spam) and completeness of the conversations(Contents in News Blogs) being consistently better. The data aggregator chosen provides data for a time period of 13 months from the current date. Data extracted from Sysomos contains *Source, Host, Link, Date(ET), Time(ET), LocalTime, Category, Author ID, Author Name, Author URL, Authority, Followers, Following, Age, Gender, Language, Country, Province/State, City, Location, Sentiment, Themes, Classifications, Entities, Alexa Rank, Alexa Reach, Title, Snippet, Contents, Summary, Bio, Unique ID, Post Source.* The following data sources were used for extraction:

1. **Social Media**, which include Twitter, Facebook, Instagram, etc. (Twitter constitutes 90% of the social media data extracted)

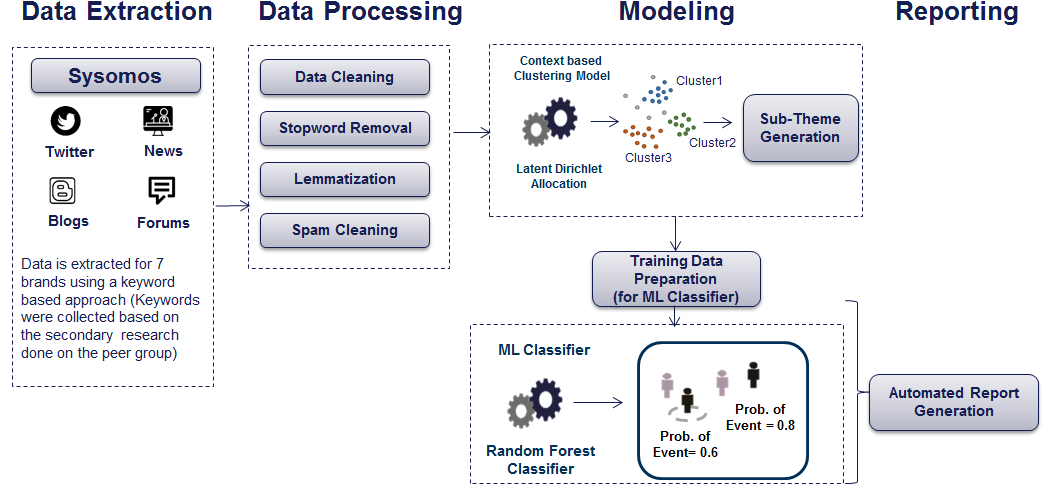
Columns used for Analysis

1. **Financial Media,** which include Financial News, Blogs, Forums, etc.

Columns used for Analysis:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Source** | **Host** | **Date(ET)** | **Language** | **Country** | | **Title** | | **Contents** |
| TWITTER | http://www.twitter.com | 3/30/2019 | English | UK |  | | RT @KezWarner: Listen to the latest @Lipper\_Alpha #FundManagerChat #podcast w/ UK Property #Fund Manager @mandgprof Fiona Rowley https://t.co/7tysm1IInk https://t.co/vTYblhV1Oe | |

# Process Flow



# Models and Refresh Process

Models and Refresh Process involves following steps:

* [Data Extraction and Merging](#_4.1_Data_Extraction)
* [Data Preprocessing](#_Data_Preprocessing)
* [Model Iteration-1 (Clustering model)](#_Model_Iteration-1_–)
* [Sub-Theme Generation](#_4.4_Sub-Theme_Generation)
* [Model Iteration-2 (Classification model)](#_4.5_Model_Iteration-2)

## 4.1 Data Extraction and Merging

Data is extracted from Sysomos (data aggregator) for both social and financial media. Each of these data files are to be read and merged to a single file in order to access it easily. Column names are altered prior to merging the Data.

NOTE TO USER:

Queries used to extract data from Sysomos are in the file below.



The following script is used to merge the extracted data.



## Data Preprocessing

Data Preprocessing involves the following steps:

* [Data Cleaning](#_4.2.1_Data_Cleaning)
* [Stopword Removal](#_4.2.2_Stopword_Removal)
* [Lemmatization](#_4.2.3_Lemmatization)
* [Spam Removal](#_4.2.4_Spam_Removal)

### 4.2.1 Data Cleaning

Before diving into the data that we merged in the previous section, first step is to clean the data in order to obtain better results. The process of data cleaning includes:

* Lower case all contents
* Remove numeric data
* Remove punctuations
* Remove special characters

Data Cleaning ensures all the data to have a uniform format.

NOTE TO USER:

Script for data preprocessing is attached at section 4.3.2

### 4.2.2 Stopword Removal

Removing stopwords (commonly occurring words) can be eliminated from the cleaned data as the frequent occurrence of these words may skew the analysis. We can create a custom list of stopwords or it can be downloaded from nltk package (base NLP package in python).

NOTE TO USER:

Script for data preprocessing is attached at section 4.3.2

### 4.2.3 Lemmatization

Normalizing text before storing or processing allows it to be consistent before operations are performed on it. Lemmatization is a useful normalization technique for text. It aims to remove inflectional endings and returns the base or dictionary form of a word.

NOTE TO USER:

Script for data preprocessing is attached at section 4.3.2

### 4.2.4 Spam Removal

Since the data downloaded from Sysomos is purely based on keyword approach, there is a high possibility that the data contains some irrelevant data.

For example, “*SLI*” as a keyword is given as input assuming it stands for “*Standard Life Investments*” whereas some of the conversations are related to “*Scalable Link Interface*” (graphic cards from Nvidia)

Such conversations are to be identified and tagged as spam so that the data we input to model will be relevant to track the brand reputation.

Performance measures for spam classifier are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Spam Classifier** | | | | |
| Total Records | 13756 | Train | 9629 |  |
| Technique Used | Random Forest | Test | 4127 |  |
| Accuracy | 0.9539617155 |  |  |  |
| Confusion Matrix |  |  | Metrics |  |
|  | *Predicted: No* | *Predicted: Yes* | Precision | 0.9540531544 |
| *Actual: No* | 2024 | 86 | Recall(Sensitivity) | 0.9538399904 |
| *Actual: Yes* | 104 | 1913 | F1 Score | 0.9539283874 |

NOTE TO USER:

The following script is used to run spam classifier:



## Model Iteration-1 (Clustering Model)

In order to study association of brands at a theme level, we have to identify the underlying themes in the data. Since the data we have is not structured or labelled, an unsupervised model (Clustering) can be useful to make a preliminary study to understand the possible themes in the asset management industry in financial and social space.

This task of identifying themes (topics) that best describes a set of documents is known as Topic Modelling. One such popular topic modelling technique is ***Latent Dirichlet Allocation*** (LDA)

### Feature Generation

Any machine learning algorithm cannot directly deal with text data. So text data should be converted to a fixed-length vectors of numbers. There are multiple ways to convert text to vectors in which count vectorization is one of the simple and useful technique. Count Vectorization involves counting the number of occurrences each word in a document.

NOTE TO USER:

Script for topic modelling is attached at section 4.3.2

### 4.3.2 Topic Modelling and Tuning

Latent Dirichlet Allocation is an unsupervised clustering model which divides the entire corpus (input set of documents) into a fixed number of clusters (topics) where all the documents in any specific topic are semantically related.

Since user need to predefine the number of topics that the corpus has to be divided into, it is very important to identify the optimal topic count. This can be done by testing with all possibilities of topic counts and then choosing the optimal count.

Optimal number of topics can be decided based on the perplexity score and log likelihood score of each model.

NOTE TO USER:

The following script is used to run topic modelling:



## 

## 4.4 Sub-Theme Generation

The above LDA model helps us to cluster all the conversations into multiple groups of similar data. Analyzing the data in these groups helps us identify multiple topics related to the asset management industry in social and financial space.

10 Sub-Themes are identified after discussing with the business experts and these sub-themes are majorly categorized into 4 Themes. The Theme and Sub-Theme categorization is as follows:

* + Corporate News
    - Business Results
    - Key Management Announcements
    - Partnerships and Mergers
    - Sponsorships
  + Macro Themes
    - Market Commentary
    - Economics/Politics
  + Emerging Themes
    - Active Vs Passive Investing
    - Innovation and Technology
    - ESG (Environmental Social Governance)
  + Product/Fund

NOTE TO USER:

More sub-themes can be added as and when required

Now that the sub-themes that can be present in asset management industry in financial and social space are identified, we can have a supervised learning model so that most of the data in the future can be tagged automatically if the conversations are related to the identified sub-themes.

## 4.5 Model Iteration-2 (Classification model)

Since the sub-themes are identified using the LDA model, tagging conversations to the sub-themes can be automated by using supervised models. A binary classifier can be built for each of the identified sub-themes. Multiple models are created after noticing that each conversation can be tagged to multiple sub-themes.

For Example: A conversation related to “*Launch of a crypto currency ETF”* can fall under multiple sub-themes such as *“Innovation and Technology”, “Active Vs Passive Investing” and “Product/Fund”*

### 4.5.1 Training Data Preparation

To create a supervised model, we have to train the model with a subset of the existing data. A balanced data set chosen to train the models as some of classifiers are sensitive to the proportions in an imbalanced dataset. So, for each of the sub-themes identified in section 4.4, balanced data sets are prepared.

NOTE TO USER:

The amount chosen for training each of the sub-themes can be varied based on the amount of data available with respect to that sub-theme

### 4.5.2 Feature Creation

Any machine learning algorithm cannot directly deal with text data. So text data should be converted to a fixed-length vectors of numbers. To convert text to vector, Term Frequency-Inverse Document Frequency (TfIdf) Vectorizer is used as it not only considers the ocuurence of the word in a document but also considers the occurrence of the word across all the documents in the corpus.

Each of the training data sets created in the section 4.5.1 are provided as input to a TfIdf Vectorizer which returns matrices of TfIdf Vectors.

NOTE TO USER:

Script for Classifier is attached at section 4.5.3

### 4.5.3 Building Classifier & Measuring its Performance

Divide each of the training sets created for 10 sub-themes in section 4.5.1 into training and test data sets and fit the train data set to a Random Forest Classification model. Save the TfIdf Vectorizers and the classifier models for all ten sub-themes into pickle objects.

To measure the performance of the models, score each of the 10 test data sets on the respective saved model objects. Also record measures such as accuracy, precision, recall and fscore. Performance of models that are recorded after scoring them with test data sets are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bussiness Results** | | | | |
| *Total Records* | 16401 | *Train* | 11480 |  |
| *Technique Used* | Random Forest | *Test* | 4921 |  |
| *Accuracy* | 0.8914854704 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.8913710239 |
| *Actual: No* | 2264 | 326 | *Recall(Sensitivity)* | 0.8924495924 |
| *Actual: Yes* | 208 | 2123 | *F1 Score* | 0.8913963092 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Management Announcements** | | | | |
| *Total Records* | 20600 | *Train* | 14420 |  |
| *Technique Used* | Random Forest | *Test* | 6180 |  |
| *Accuracy* | 0.9593851133 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9604940929 |
| *Actual: No* | 3052 | 53 | *Recall(Sensitivity)* | 0.9592702565 |
| *Actual: Yes* | 198 | 2877 | *F1 Score* | 0.9593525196 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Partnerships And Mergers** | | | | |
| *Total Records* | 8860 | *Train* | 6202 |  |
| *Technique Used* | Random Forest | *Test* | 2658 |  |
| *Accuracy* | 0.9635063958 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9639179464 |
| *Actual: No* | 1298 | 28 | *Recall(Sensitivity)* | 0.9635410297 |
| *Actual: Yes* | 69 | 1263 | *F1 Score* | 0.963500067 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sponsorships** | | | | |
| Total Records | 2490 | Train | 1743 |  |
| Technique Used | Random Forest | Test | 747 |  |
| Accuracy | 0.983935743 |  |  |  |
| Confusion Matrix |  |  | Metrics |  |
|  | *Predicted: No* | *Predicted: Yes* | Precision | 0.9841712687 |
| *Actual: No* | 363 | 10 | Recall(Sensitivity) | 0.9839213775 |
| *Actual: Yes* | 2 | 372 | F1 Score | 0.9839334108 |
| **Market Commentary** | | | | |
| *Total Records* | 21126 | *Train* | 14788 |  |
| *Technique Used* | Random Forest | *Test* | 6338 |  |
| *Accuracy* | 0.9605553802 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9605500724 |
| *Actual: No* | 3112 | 123 | *Recall(Sensitivity)* | 0.9605251138 |
| *Actual: Yes* | 127 | 2976 | *F1 Score* | 0.96053721 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Economics and Politics** | | | | |
| *Total Records* | 13138 | *Train* | 9196 |  |
| *Technique Used* | Random Forest | *Test* | 3942 |  |
| *Accuracy* | 0.9822425165 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9822569374 |
| *Actual: No* | 1987 | 33 | *Recall(Sensitivity)* | 0.9822072526 |
| *Actual: Yes* | 37 | 1886 | *F1 Score* | 0.9822310817 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Active Vs Passive Investing** | | | | |
| *Total Records* | 12046 | *Train* | 8432 |  |
| *Technique Used* | Random Forest | *Test* | 3614 |  |
| *Accuracy* | 0.9916989485 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9916933289 |
| *Actual: No* | 1839 | 15 | *Recall(Sensitivity)* | 0.9916933289 |
| *Actual: Yes* | 15 | 1745 | *F1 Score* | 0.9916933289 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Innovation and Technology** | | | | |
| *Total Records* | 12400 | *Train* | 8680 |  |
| *Technique Used* | Random Forest | *Test* | 3720 |  |
| *Accuracy* | 0.9688172043 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9689324707 |
| *Actual: No* | 1832 | 47 | *Recall(Sensitivity)* | 0.9687535322 |
| *Actual: Yes* | 69 | 1772 | *F1 Score* | 0.9688090901 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ESG** | | | | |
| Total Records | 11746 | Train | 8222 |  |
| Technique Used | Random Forest | Test | 3524 |  |
| Accuracy | 0.9787173666 |  |  |  |
| Confusion Matrix |  |  | Metrics |  |
|  | *Predicted: No* | *Predicted: Yes* | Precision | 0.9791025866 |
| *Actual: No* | 1771 | 18 | Recall(Sensitivity) | 0.9785427436 |
| *Actual: Yes* | 57 | 1678 | F1 Score | 0.9787025339 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product/Fund** | | | | |
| *Total Records* | 11448 | *Train* | 8013 |  |
| *Technique Used* | Random Forest | *Test* | 3435 |  |
| *Accuracy* | 0.9735080058 |  |  |  |
| *Confusion Matrix* |  |  | *Metrics* |  |
|  | *Predicted: No* | *Predicted: Yes* | *Precision* | 0.9734541851 |
| *Actual: No* | 1697 | 59 | *Recall(Sensitivity)* | 0.9736709737 |
| *Actual: Yes* | 32 | 1647 | *F1 Score* | 0.9735023916 |

NOTE TO USER:

Script for Classifier is in the following file:



### 4.5.4 Testing the Model on Actual Data

Now that all the model objects for 10 sub-themes are built and tested, complete data set created after spam removal (section 4.2.4) can be scored on these models.

To score the data on these models, fit the data into each of the TfIdf Vectorizer object and pass these vectors to the corresponding classifier object. The result of the classifier will be binary which is appended to the original data. Each of the data conversation will be tagged to one or more sub-themes.

There is also a possibility that a conversation may not be tagged to any of these 10 sub-themes. All such conversations are to be analyzed and one of these two actions are to be taken:

1. If the conversation can be tagged to any of the 10 sub-themes that are defined, alter the existing training data set by adding few more records and train the model again.
2. If the conversation doesn’t fit under any of these defined sub-themes, create a new sub-theme and create a model for it.

NOTE TO USER:

Script for scoring the final data set on model objects is in the following file:



### 4.5.5 Model Refresh

NOTE TO USER:

Steps to be followed for scoring models:

Step1: Extract all data from Sysomos for a specific time period and merge all the files downloaded into a single file as shown in section 4.1

Step2: Preprocess the merged data by cleaning, removing stopwords, lemmatizing and spam cleaning as shown in section 4.2

Step3: For each of the 10 models created, score the model on the existing data as described in section 4.5.4

The model files that are to be used for each of the sub-themes are as follows:

|  |  |  |
| --- | --- | --- |
| Sub-Theme | Classifier File | Vectorizer File |
| Business Results |  |  |
| Management Announcements |  |  |
| Partnerships and Mergers |  |  |
| Sponsorships |  |  |
| Market Commentary |  |  |
| Economics and Politics |  |  |
| Active Vs Passive Investing |  |  |
| Innovation & Technology |  |  |
| ESG |  |  |
| Product/Fund |  |  |

Step4: Analyze the unclassified data and prepare the next version of training data accordingly. Provide the new trained data to Random Forest Classifier and train it as shown it section 4.5.3.

Step5: Iterate steps 3&4 till all the data gets classified.

# Reporting and Packaging

NOTE TO USER:

The above data is used to update the following report:

**### Recent report to be attached**