Academic Article

Big Data in Financial Research: Applications in Equity Trading

Received: *January 11, 2019*
Revised: *June 14, 2019*
Accepted: *July 4, 2019*

Dr.Pantisa Pavabutr
Associate Professor of Finance,
Thammasat Business School, Thammasat University

Siraprapa Watakit
Data Scientist, Center of Expertise, Ageas, Hong Kong

ABSTRACT

The internet of things (IoT) has put us all under the radar of the world wide web of information resulting in profound changes in the marketplace. In the world of finance, Big Data generated by IoT provide rich real time information prompting new approach towards predictive modeling. The purpose of the article is to familiarize readers with terminology used in data analytics today as well as to demystify the “black box” of machine learning algorithm. We provide schematic comparisons between analytical tools used in traditional econometrics and in financial machine learning. Users of the traditional approach will find that these methods are not new. We then provide examples of applications in equity trading using information in the limit order book and sentiment data extracted from Reuters. An indicator for limit order book imbalance (ILOB) and sentiment index on news feed are created. ILOB generates signals that decay substantially when the holding period is increased from 15 to 30 minutes. A high sentiment index score is associated with higher stock prices but the correlation is weak.

Keywords: Big Data, Financial Machine Learning, Supervised Learning, Unsupervised Learning, Limit Order Book, Text Mining, Sentiment Index

บทความวิชาการ

การใช้บิ๊กดาต้าสำหรับการวิจัยทางการเงิน และการประยุกต์ใช้ในการซื้อขายหลักทรัพย์

วันที่ได้รับต้นฉบับบทความ : *11 มกราคม 2562*
วันที่แก้ไขปรับปรุงบทความ : *14 มิถุนายน 2562*
วันที่ตอบรับตีพิมพ์บทความ : *4 กรกฎาคม 2562*

ดร.พันทิศา ภาวบุตร
รองศาสตราจารย์ประจำภาควิชาการเงิน
คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์

ศิรประภา วาทกิจ
Data Scientist, Center of Expertise, Ageas, Hong Kong

บทคัดย่อ

ยุคของ “อินเทอร์เน็ตในทุกสิ่ง” (Internet of Things หรือ IoT) เป็นยุคที่อุปกรณ์รอบตัวเราเป็นตัวเชื่อมโยงทุกสิ่งทุกอย่างสู่โลกอินเทอร์เน็ตและนำมาซึ่งการเปลี่ยนแปลงในตลาดการบริโภคสินค้าและตลาดการเงินการลงทุน ในโลกของการเงินนั้น IoT เปิดโอกาสให้นักลงทุนได้รับข่าวสารและข้อมูลในระดับจุลภาคที่ทันท่วงทีจึงเป็นปัจจัยสำคัญที่ผลักดันให้เกิดแบบจำลองเพื่อการพยากรณ์ที่ต่างไปจากวิธีเดิม บทความนี้มีวัตถุประสงค์แนะนำศัพท์บัญญัติที่เกี่ยวกับการจัดการข้อมูลชุดใหญ่หรือที่เรียกกันว่า “บิ๊กดาต้า” และสร้างความเข้าใจเกี่ยวกับกระบวนการวิเคราะห์ข้อมูลชุดใหญ่ที่หลาย ๆ คนเข้าใจว่าเป็นเรื่องลึกลับและยากที่จะเข้าใจ ผู้เขียนนำเสนอเปรียบเทียบกระบวนการทางเศรษฐมิติแบบดั้งเดิมกับอัลกอริทึ่มหรือกระบวนการเรียนรู้ของเครื่องคอมพิวเตอร์ และยกตัวอย่างการนำกระบวนการดังกล่าวไปในการซื้อขายหลักทรัพย์โดยอาศัยข้อมูลการซื้อขายระหว่างวันของตลาดหลักทรัพย์แห่งประเทศไทยและข้อมูลการวิเคราะห์อารมณ์และความรู้สึกจากข่าว Reuters ผู้เขียนได้ประมาณตัวชี้วัดความไม่สมดุลระหว่างปริมาณซื้อและขาย และ ดัชนีชี้วัดทัศนคติของผู้ลงทุนจากข่าว พบว่าอัตสหสัมพันธ์ของตัวชี้วัดความไม่สมดุลระหว่างปริมาณซื้อและขายลดลงอย่างเห็นได้ชัดเมื่อระยะเวลาในการถือครองหลักทรัพย์เพิ่มจาก 15 นาที เป็น 30 นาที ส่วนดัชนีชี้วัดทัศนคติของผู้ลงทุนจากข่าวมีความสัมพันธ์ในทางเดียวกับราคาหลักทรัพย์แต่มีค่าสหสัมพันธ์ที่อ่อนค่า

คำสำคัญ : ข้อมูลชุดใหญ่ ระบบที่สามารถเรียนรู้ของคอมพิวเตอร์ ระบบการบันทึกคำสั่งซื้อขาย การเรียนรู้โดยมีผู้สอน การเรียนรู้โดยไม่มีผู้สอน การวิเคราะห์อารมณ์และความรู้สึก

1. INTRODUCTION

Today's internet of things (IoT) put us all under the radar of the world wide web of information resulting in profound changes in the marketplace. The public life of Big Data as we read in the popular press conjures up visions of complicated neural network prescriptions for machine learning, advanced and incomprehensible quantitative models, and unfamiliar computer architecture of programming languages. Such preconceived ideas often places technology at the forefront of humans in understanding data. In the field of finance, Big Data has brought about changes with finance professionals increasingly adopting quantitative techniques particularly in investments. With advances in trading platforms and supporting technology of machine learning and AI, there is growing popularity in algorithmic trades, sentiment extraction from text analysis, and merging of fundamental and quantitative investment styles.

The purpose of the article is to familiarize readers with terminology used in finance data analytics today as well as to demystify the "black box" of machine learning algorithm via our samples and programming snippets. Users of the traditional approach to data analysis will find that these technology driven approaches are not new. As a matter of fact, the lessor known aspect of the Big Data is that as intriguing and plausible machine learning concepts may sound, it may not always lead to strong investment signals and sustainable alphas (excess return from investment management) as we will show. We begin by providing background discussion on Big Data terminology, describe some common applications of Big Data in finance and provide schematic comparisons between analytical tools used in traditional econometrics and in financial machine learning. We demonstrate two types of strategies that employ Big Data in equity markets: a market microstructure strategy that exploit profitable opportunities from transaction level trading flows, and a text mining strategy of news articles for predicting stock price movement. For programmers, we provide sample codes for our empirical work in the appendix.

2. BIG DATA: DEFINITIONS, RESOURCES, AND METHODOLOGIES

2.1 Definitions

Finance people who have long dealt with large amounts of data would probably see Big data" as just data. No matter what the data size; we deal with it the same way: Collect, analyze, and extract value. In the context of today's technology and application, the popular lexicon "Big Data" contains features widely described as the five V's of "Big" data. Figure 1 maps the stages of data work to "Big data" characteristics. Under stage 1, advancement of technology allows more speedy collection and storage such that the term "Big" these days consist of three prominent characteristics: Volume: The size of data is very large with subjective lower bounds continuously being revised upwards.[1] Velocity: The speed with which data is generated, sent, and received is virtually instantaneous. Variety: Data come in variety of formats: structured data are tabulated in Structured Query Language (SQL) tables or Comma separated value (CSV) files; semi-structured are mix of text and tabulation often found on websites are in Hypertext markup language (HTML) or Javascript; and unstructured data are in form of images, voice and video. In stage 2, data analytics, the key tool in screening relevant data depends on the key term: Veracity: The degree of accuracy or reliability of data must be assessed as the noise component in the data collected may be disproportionately higher than modes of traditional data collection. Finally, stage 3; the extraction process using algorithms to evaluate business value.

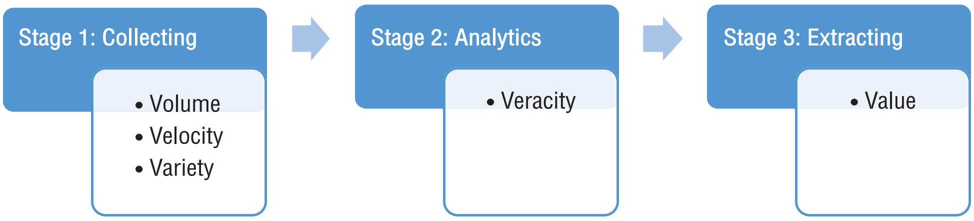

Figure 1: Big Data Steps and Characteristics

It is also important to recognize that technological advances in computing environment:- disk, central process unit (CPU), memory, network, and specialized software lead to an increasing capacity to handle data volume and lowered processing time. Other important area of progress is in data storage and retrieval system from a relational database management system (RDMBS) to a distributed file system computing. The most popular platform is called “Hadoop.” Unlike the RDMBS, which is a self-organizing collection of integrated records accessible by structured queries, the current trend is to store data in a distributional system like the Hadoop. The system works by breaking up data into small pieces and distributed to many computers such that overall calculation time is reduced before fragmented results from data truncation are then consolidated into one single dataset.

2.2 Big Data Resources for Financial Research

Two types of data are widely used in finance; (i) transaction data from market trading, corporate announcements, and government/institutional agencies data; and (ii) data generated by individuals typically available on social media websites (Facebook, Renren, and Twitter), product review websites, (Amazon, CNET, or even Pantip), web search trends (Google trends, and Alexa). Transaction data is now readily available on- line for free if it is economic type variables collected by central banks, governments,

[1] The terms gigabyte has given way to larger measures such as Petabytes (1 million Gigabytes), Exabytes (1 billion Gigabytes), and Zetabytes (1 trillion Gigabytes).

or international agencies like International Monetary Fund (IMF) or World Bank. Transaction databases for equities, derivatives, and commodities are usually available by direct contact with relevant exchanges, sometimes for a fee. In other cases, a number of firms such as Euromonitor and Nielsen emerge to track commercial transactions and make datasets available only to members. Among individual activity data, social media sentiment analysis is probably the most widely used alternative data for prediction of asset price movement. Besides popular web trend engines, sentiment analysis is provided by specialized financial data provider like RavenPack and SentimenTrader. (Kolanovic and Krishnamurai (2017) provides excellent review of Artificial Intelligence (AI) investment strategies and list of data analytics resources for financial services).

From a plethora of internet base sources, finance professionals must sieve through datasets that provides either the “alpha” content for investing or for better serving the product needs of clients. The screening process comes at two levels: the first level is to make sure that data is accurate and care must be taken to remove replications, outliers, and identify errors. The second level is the data analytics process to search for relationship among variables or predict certain variables as a function of others using the classic econometric techniques such as regressions, factor analyses, and regime change identification. Machine learning tools can be used to aid search for best prediction variables and functional relationships as well as provide new kinds of data for traditional questions; for example, measuring economic activity with satellite images or extracting sentiment from text analysis.

2.3 Financial Machine Learning Methods

2.3.1 Terminology

Machine learning is a method of data analysis where the computer is given a set of inputs (predictor variables or datasets) and outputs (predicted variables). The computer then finds the rules or “learns” the rules that best links inputs and outputs. Clearly, machine learning is a field that integrates concepts from two fields, statistics and mathematics, and computer science. As it relates to finance, financial machine learning incorporates concepts of financial economics. For how else can the user differentiate spurious correlations from random chance draws of data from true economic relations or resolve the problem of screening relevant economically grounded inputs to obtain target predicted outputs. The diverse nature of machine learning has led to some confusing terminology. We summarize common terms that are used interchangeably across fields in Table 1.

Table 1: Financial Statistics vs Computer Science Machine Learning Terminology

Financial Statistics Terminology	Computer Science Financial Machine Learning Terminology
In-sample data	Training data, training set
Out-of-sample data	Test data, test set, validation set
Dependent variable	Target, output
Independent variable	Attribute, input,
Variable	Feature
Model parameters	Model weights
Categorical response	Label
Regression	Supervised learning
Cluster analysis and dimensionality reduction (Factor analysis and Principal component analysis)	Unsupervised learning

2.3.2 Machine learning approaches

There are various ways to classify machine learning approaches. From econometrics view, machine learning can be divided based on method of input-output identification. Supervised machine learning begins with the researcher identifying variable sets that are inputs or outputs. The researcher then “supervise” the machine to find a rule, an equation, to predict the output variable. In contrast, unsupervised machine learning, allows the machine to learn correlations and linkages among input and output variables with the goal to reduce dimensionality of the data. Statistically speaking this is akin to factor or cluster analyses. Some terminology like deep learning, developed by computer science is used to analyze unstructured data such as pictures, videos, and texts to find non-linear relationship and patterns that generates output forecasts from credit card fraud, winning stock of the day, or regime shifts in financial time series.

Alternatively, data scientists may use special modeling technique or method of conditioning information numeric or text to classify machine learning. The terms neural network, algorithmic trading, and text mining are fairly standard approaches used today. We describe them more explicitly below. In the following discussion, we take more care describing text analysis as it entails fairly unfamiliar territory from usual quantitative methods.

Neural network

Despite the hype around neural network as they are described as the ability that machines can learn as "humans" would, the problem can be framed as solving a composite non-linear function. Figure 2 and equation (1) depict a vector of inputs $X \in (x_1, x_2, x_3, x_4)$ for which we seek to detect linkages to output $\mathrm{y} = \mathrm{y}(\mathrm{Z}, \mathrm{W}, \mathrm{X})$. Let's say, a quantitative analyst is trying to train a model to select from a set of inputs which determines a winning stock (output). The model we are trying to fit can look like this,

$$y(Z, W, X) \; = \; \Theta\,(Z{\cdot}g(W{\cdot}X)); \qquad (1)$$

where,

W = Vector of parameters (weights) connecting input variables to hidden layers

$g(.)$ = Function of the hidden layers

Z = Vector of parameters (weights) connecting hidden layers to out layers

Θ = Activation function connecting inputs, hidden layers, to output layers.

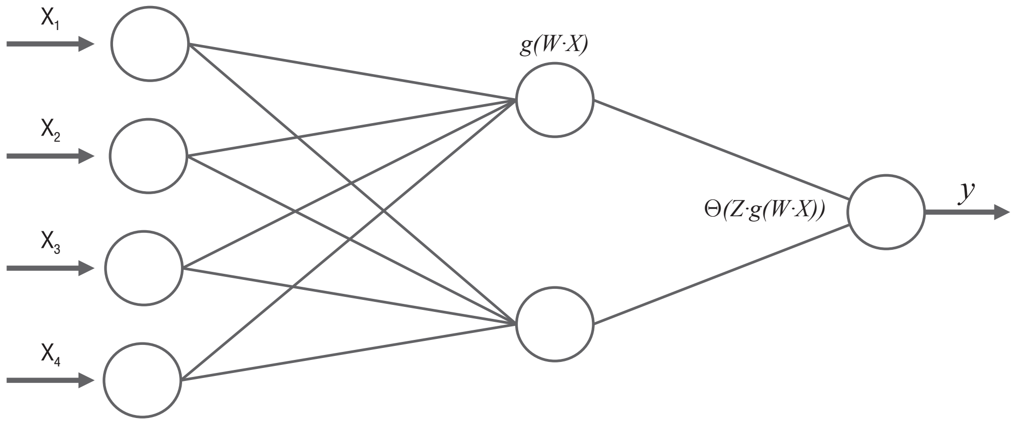

Figure 2: Simple Three Layer Neural Network

In the above example, there are two possible classes of the link function $g(W{\cdot}X)$ which ultimately represents output y depending on model selection of weight vector Z. More explicitly, the activation function Θ and an additional layer of parameters Z determine the course of action whether to apply a link function that places more weight on firm accounting information relative to trading information or vice versa.

Algorithmic Trading

The term algorithmic trading often replaces technical trading rules after the advent of high frequency trading and increased program trading. These technologies allow more complicated conditioning rules on trade data information to determine buy sell signal and utilizes automated market monitoring system to send out automated orders to minimize price impact. A set of rules to buy a stock can include, for example, rule (1) stock is included in top 20% momentum rank, rule (2) stock is trading above 100-day moving average, and rule (3) stock is added to constituent index. Rather than discussion of mining for variables that should best explain equity price, we approach the prediction problem by putting economics first and hence begin our focus motivating the significance of order flow as main conditioning information. This is because order flows are derived from the represent investor demand functions in market microstructure models.

A long list of theoretical and empirical research supports the view that order flow is highly responsible for information impounded in stock price. Early modeling in market microstructure literature also acknowledges the usefulness of technical trading rules. Kyle (1989) shows how informed orders effect asset price in a market with informed and noise traders. Under the assumption that traders receive noisy signals, Blume, Easley, and O'Hara (1994) show how volume, information precision, and price movements are related, and introduce a model to prove that sequences of volume and prices can be informative. With support of modern day high frequency trading, Easley, de Prado, and O'Hara (2016), empirically find that tick rule approaches and bulk volume classifiers are relatively good predictors of the aggressor side of trading. Extending analysis to limit order markets, Hollifield, Miller, Sandas, and Slive (2007) argue that buy and sell orders can be thought of market participants exposing beliefs of their own forecasts of equity value. In addition, market orders are irrevocable commitments to buy or sell and therefore carry the most powerful information. For modern limit order markets, the standing orders on the limit order book (LOB) is proxy for buy and sell interest and the queue imbalance which describes the difference between best bid and ask quotes could constitute a powerful predictor of price. (Rosu, 2009; Gould et al. 2013). Schulmeister (2009) finds that daily technical trading tools are no longer profitable on the S&P500 modern day trading and conjectures that stock price trends are shifting to higher frequencies. This is confirmed by Gao, Han, Li, and Zhou (2018) who use intraday trading momentum signals on S&P 500 exchange traded fund (ETF) from 1993–2013 and document predictability within half-hour holding period. We provide an example and describe the steps of using trading rule with LOB as conditioning information in section 3.

Text Mining

From information theory, investor make investment decision based on news and information which can either be public or private. Sentiment analysis is a process of gauging general opinion towards topics by aggregating infomation from various sources such as company news, company annoucement and analyst reports. In recent years, many empirical study show that sentiment data have material impact in predicting market direction. (see Da, Engleberg, and Gao (2014), Antweiler and Frank (2004), Bollen, Mao, and Zheng (2011), and Tantaopas, Padungsaksawasdi, and Treepongkaruna, (2016)). To construct a sentiment index of a certain topic, we simply collect information with regards to the topic and derive a sentiment value by using machine learning algorithm. In the following section, we demonstrate sentiment extraction from text mining of Thomson Reuters news

Text mining otherwise known as Natural language processing (NLP) is the use of computational and statistics tools to analyze text. The process involves a large scale automated processing of plain text in digital format that are converted into quantitative formats that we can analyze and extract business value. From behavioral economics standpoint, text provide information regarding market sentiment. Barberis, Shleifer, and Vishny (1998) and Admati and Pfleiderer (2001) have built theoretical models where sentiment effects asset prices. Antweiler and Frank (2004) and Tetlock (2007) are early studies to use internet message boards and news media to extract investor sentiment to predict markets. Hoberg and Phillips (2016) use machine learning to analyze text in corporate 10-K filings in order to classify industries.

There are five main steps to text mining:

Step 1 Data acquisition: A web crawler[2] is created taking on a starting date, days to look back, and search parameters. The software will crawl sites looking for news articles fitting the parameters.

Step 2 Pre-processing data set: Text must be simplified by removal of unnecessary contents (punctuations and high frequency words that do not contain significant information), and transformation into root forms (for example, increased, increasing, increasingly to increase).

Step 3 Document representation: In this step, the contents of the documents must be structured for computing predictive models. To do so, the analyst creates a frequency term matrix (see Table 2) that converts the documents or collection of sentences into a $D \times T$ matrix, where D is the number of documents in the study sample (corpus), and T is a number of unique terms or tokens. For example, suppose we have the following documents with these unique key terms:

Document 1: Earnings by far meet expectations ➔ Earnings far meet expectations

Document 2: Earnings far below expectations

Document 3: Earnings far above expectations

Document 4: Earnings is so far above expectations ➔ Earnings is far above expectations

[2] A software application often known as “bots” that scrapes text from specified web pages.

Table 2: Frequency Term Matrix

Doc Num	Earnings	Expect	Meet	Below	Above	Far	Class
Doc 1	1	1	1	0	0	1	Neutral
Doc 2	1	1	0	1	0	1	Negative
Doc 3	1	1	0	0	1	1	Positive
Doc 4	1	1	0	0	1	1	Positive

Another way of counting words is to create vector representations of text (see Table 3), also known as "bag of words."

Table 3: A Vector Representation of Document 3

Document Terms	Words Frequency (Weights)
Earnings	1
Expect	1
Meet	0
Below	0
Above	1
Far	1

There are three common ways to delineate important words that convey sentiment in a document:

Word count: Sum of each token appearance in a document.

Word proportion: Percentage frequency of token in a document.

Term frequency inverse document frequency (TF.IDF): A simple counting approach of positive and negative words may not convey sentiment in a document given linguistic subtleties from use of idioms and irony. Furthermore, if a word is used in many documents within the corpus, the power to discriminate content classification is reduced. In Table 2, the token "far" is used in all three documents, each with different content classification. The TF.IDF takes care of these issues and is derived from

$$tf.idf_{t,d} = (1 + \log f_{t,d}) \cdot \log\left(\frac{D}{Df_t}\right) \tag{2}$$

where $\mathbf{D}$ is the total number of documents in a corpus, df_t is number of documents in which the term $\mathbf{t}$ appears and $f_{t,d}$ is frequency of term $\mathbf{t}$, in document d. For example, a corpus contains 10 million documents. One thousand documents within the corpus contains the word "performance." A document containing 100 words where "performance" appears twice;- $tf.idf_{t,d} = (1 + \log\ 0.02) \cdot \log \frac{10M}{1000}$ $= -2.79$. The first factor in the term gives more weights to words that appear more frequently in a document. The second term gives greater weight to words that appear less frequently in entire corpus.

Step 4 Predictive modeling

Next we are interested in identifying document type positive or negative, $y \in [1,0]$ from frequency term $\mathbf{x}$, as in $y = f(\mathbf{x})$ from a set of training examples $D = [(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,(\mathbf{x}_n,y_n)]$. To do so, we need to estimate the probability that frequency term $\mathbf{x}$ belongs to type $\mathbf{y}$ in document (equation 1). The goal is to model the conditional probability that the i^{th} document belongs to a classification with link function ψ;

$$p(y \mid \beta, \mathbf{x}_i) = \psi(\beta^T \mathbf{x}_i); \tag{1}$$

A logistic link on the corpus D can be written as

$$L(\beta) = p(\beta \mid D) \propto \left(\prod_{i=1}^{n} \frac{1}{1 + \exp(-\beta^T \mathbf{x}_i y_i)} \, p(\beta) \right) \tag{2}$$

Where $p(\beta)$ is the prior on β and i^{th} document indexes the training example in the body of documents D. The initial classification in training data that gives rise to the prior can be derived from classification dictionaries to class words associated with optimism or pessimism (Tetlock 2007) or complicated computational linguistics approach (Hansen, McMahon, and Prat 2014).

Besides logistic regressions, decision tree classifiers can be used to predict document type y. The analyst begins by evaluating the initial entropy denoted $H(y)$ in equation (3). Entropy is a measure of categorization difference. Here p_c is the probability that a document belongs to class y (positive or negative; 1, 0). The initial entropy is the sum of the probability of each document type times the log base 2 of that probability. Usually, the first node entropy value lies above 0.9 but less than 1.0.

$$H(y) = -\sum_{c=1}^{2} p_c \log_2 pc = -[p_1 \log_2 p_1 + p_0 \log_2 p_0] \tag{3}$$

Moving down the nodes of the tree, the analyst finds a feature in the data to partition it in ways that reduces the noise (entropy). Referring to the four documents earlier in this section, we can first partition documents by a feature that contain or not contain the adjective "*above*." This partitions the documents into two classes which enables us better separate positive documents from negative ones resulting in reduced entropy because in the probability of finding positive documents in segments that contain (not contain) the feature "*above*," goes up (down). In a more complicated document structure, the decision tree algorithm searches for the next best feature after the initial division that helps to partition the documents at more refined levels. The search continues until it can no longer find a reduction between the entropy between previous nodes to the next nodes. We would expect that as we move from root node (initial node) to leaf node (predicted category), the entropy score at each evaluation point would go down. The reduction in entropy at each level is called "information gain." A variation of the decision tree is the use of a random forest classifier. A random forest consists of multiple decision trees. Rather than selecting the best feature divisor from the root, the algorithm randomly selects random subsets of factors at each node for each split. The result is usually a varying number of leaf nodes which can then be averaged. In sum, a random forest classifier is an average result of multiple random decision trees, and hence the risk of overfitting a model is substantially reduced usually yielding superior predictive power over simple decision tree or traditional regression type model (Krauss and Huck; 2017).

Step 5 Evaluation Metrics

Evaluation metrics of predictive models in text mining have special terms. Table 4 lays out the contingency table of text classification label. True positives (TP) and true negatives (TN) are samples that the model classifies correctly in each of their own class. False positives (FP) and false negatives (FN) are examples that are incorrectly specified as positive and negative. Thus, the strength of the model's accuracy can be calculated from the proportion of TPs and TNs. We list varying measures below Table 4.

Table 4: Text Classification Label for Model Evaluation

Model Classification	Correct Positive Label	Correct Negative Label
Model classified positive	True positive (TP)	False positive (FP)
Model classified negative	False negative (FN)	True negative (TN)

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F-1 score} = \left(\frac{recall^{-1} + precision^{-1}}{2}\right)^{-1} = 2 \cdot \left(\frac{recall \cdot precision}{recall + precision}\right)$$

The evaluation metric provides the analyst with guideline of best model performance which can then be applied to market prediction. A sentiment score derived from number of optimistic to pessimistic document can then be created to predict asset prices. Tetlock (2007) and Aase (2011) use the strategy of word count modeling and correlated them to asset prices.

3. APPLICATION EXAMPLES: EXTRACTING “ALPHA” SIGNALS FROM TRANSACTION AND SENTIMENT DATA

One of the most widespread applications of Big Data in finance today is undoubtedly in the trading equities and commodities.[3] In this section, provide two examples of Big Data application in equity trading. In the first example, we use the rich transaction data provided by the Stock Exchange of Thailand to form algorithms for intraday trading conditioning on order flow. In the second example, we describe the procedures of text mining to extract sentiment from Reuters news.

3.1 Order Flow and Market Direction

Like most markets of the world today, trading on the Stock Exchange of Thailand (SET) is predominantly based on an order driven system. The process begins when the buyer or seller submits their orders via the brokerages. These orders are then electronically submitted from the brokerages to the SET's computerized order matching system which automatically queues orders and matches them according to a single price that generates the greatest trading volume at opening and close and according to price-then-time priority during opening hours. In late 2012, the SET adopts “CONNECT” a new trading engine to accommodate high speed, high frequency transactions. Table 5 tabulates a small sample of a fifteen minute interval limit order book we constructed from a full limit order book. The column bid and ask is the standing bid-ask price at the i^{th} 15 minute interval (from 1–5) of the day’s trade. Sumbid and sumask columns add up all queued bid and ask orders in each interval and in general is a good indicator of trading interest. We use a scaled difference between sumbid and sumask to determine the net buying (selling interest) for positive (negative) differences. A positive (negative) imbalance suggests more buying (selling) interest, which has been shown theoretically and empirically to have ability to predict short-term returns (Resu, 2009; Gould et al. 2013). The price column is the last transacted price within the chosen 15 minute interval.

[3] Aldridge (2013) provides clear introduction to main approaches to algorithmic trading strategies.

Table 5: Sample Limit Order Book of a Single Stock

Date	Interval	Ticker	Bid	Sumbid	Ask	Sumask	Price
4/1/2016	1	ADVANC	144	6432700	147	6217500	146.5
4/1/2016	2	ADVANC	146	1022300	146	927500	146
4/1/2016	3	ADVANC	140	2723100	145.5	2558700	145.5
4/1/2016	4	ADVANC	145.5	1998600	146	1844800	146
4/1/2016	5	ADVANC	145	757800	145	1133600	145.5

We provide data analytics of intraday order queue imbalance using order files of trade in 2016 provided by the SET. We use a total of 55 firms on SET50 lists including 5 firms on reserve list for the same year. Brokers can directly observe order flows of their clients and broker networks. Retail investors can see the best 5 bids and offers on their trading applications. By monitoring the quantities that are available for purchase or sale at specified prices, investors can deduce current state of market demand and supply. We create an indicator variable for the limit order book or ILOB(t) for normalized bid ask queues aggregated every 15 minutes. At a given 15 minute interval during market continuous auction trade, let

$$ILOB(t) = \frac{n^b(b_t, t) - n^a(a_t, t)}{n^b(b_t, t) + n^a(a_t, t)} \qquad (4)$$

The quantity ILOB indicates the relative strengths of buying and selling pressure. When the number is 0, close to 1, and close to –1 we deduce that buying and selling is approximately balanced, very strong buy, and very strong sell. In Figure 3, we plot the distribution of ILOB for the sample period at 15 minute trading interval. The mean and standard deviation of ILOB is 0.0417 and 0.3949 respectively. The fairly wide imbalance range is typical for large tick stocks in our sample as large and liquid stock are exposed to active trades all day and more frequent arrival of information flows leading to dynamically changing limit order book shape.

Table 6 shows that ILOB are autocorrelated up to the fourth lag suggesting that net buy or sell queues are persistent up to an hour of trading. ILOB has the most significant impact on return at the contemporaneous time interval followed by a reversal of returns at the first and third lags. The alternate signs is to be expected from a consequence of bid-ask bounce. It is no surprise to observe significant positive autocorrelation between ILOBs. Today's trading algorithms chop large orders into numerous small orders, so it is order flow rather than individual orders that relate to trade motivation. Furthermore autocorrelation in ILOBs is expected, since trading is also done dynamically as brokers allow clients to strategically to place multiple orders at various price levels in the book, monitor the progression of their limit orders in the queue, and cancel and replace orders at different levels on their trading applications.

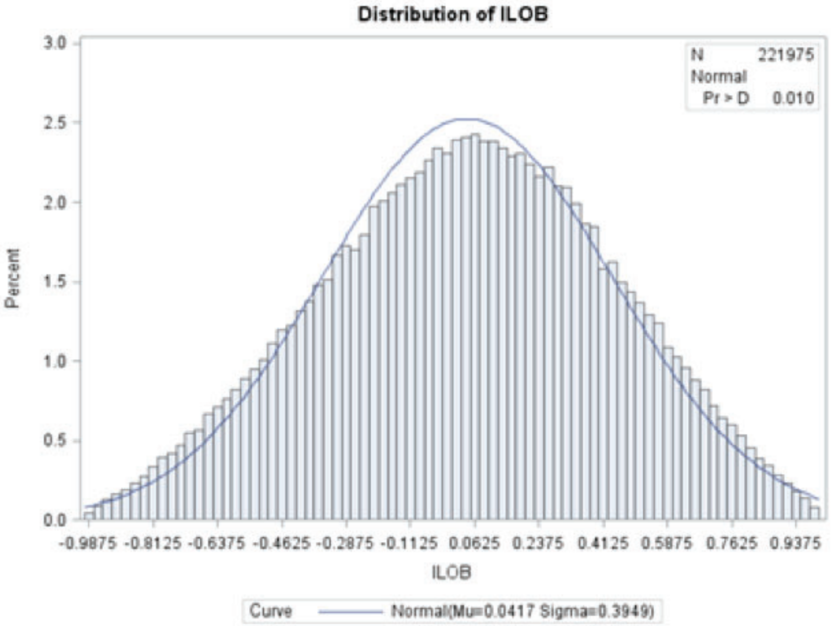

Figure 3: Distribution of ILOB of SET 50 Stocks in 2016

Table 6: Correlation between 15 Minute Return and Contemporaneous ILOB and Lags of ILOB

	RET	**ILOB**	**LILOB**	**L2ILOB**	**L3ILOB**	**L4ILOB**
RET	1.0000	0.3709 <.0001	–0.0583 <.0001	0.0458 <.0001	–0.0005 0.8542	0.0185 <.0001
ILOB	0.3709 <.0001	1.0000	0.1354 <.0001	0.1051 <.0001	0.0807 <.0001	0.0699 <.0001

To closely examine how queue imbalance affects future price changes, we run a logistic regression of function $y_{it}(ILOB_{it})$ which is an indicator variable with 0 for no change or downward price movement, and 1 for upward price movement. Both single stocks and aggregate panel regressions are estimated. To save space, only the aggregate panel logistic regressions are using contemporaneous and various lag lengths of ILOB. It appears that concurrent ILOB has the strongest effect on price direction whereas the second and fourth ILOB lag have predictive power on returns. For example, model 1 suggests that ILOB value of 1.0 compared to 0 raises the odds of observing contemporaneous prices up 2.6 times or in probability of observing prices go up is 92%. In model 3, with lags 1 and 2 of ILOB included, the odds of observing prices go up when all indicator ILOB variable is 1.0 is 86%. With contemporaneous terms removed, we find that the lags of ILOB from 1 to 4 is related to odds of price going up by 48%. The analysis suggests that ILOBs are significantly strong predictors of price direction, and that the size of ILOB at contemporaneous and different lags have differential impact on return movement. The estimation uses a step-wise logistic estimation.[4] The Wald test reveals that all model specification point to joint significance in all parameter estimates. Our simple trading rule based on intraday LOB appear to support recent empirical work (Gould et al, 2013; Easley, de Prado, and O'Hara, 2016) that bulk volume are better linked to proxies of information-based trading.

Table 7: Logistic Regressions of Order Queue Imbalance Indicator as Predictor of Stock Price Direction

Parameters	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Intercept	–0.1503 <.0001	–0.1305 <.0001	–0.1339 <.0001	–0.1277 <.0001	–0.1279 <.0001	–0.0216 0.0007
ILOB	2.6105 <.0001	2.8024 <.0001	2.7933 <.0001	2.8101 <.0001	2.8102 <.0001	
Lag 1 ILOB	–0.9093	–0.9224 <.0001	–0.9066 <.0001	–0.9074 <.0001	–0.4318 <.0001	<.0001
Lag 2 ILOB		0.1112	0.1329 <.0001	0.1325 <.0001	0.3074 <.0001	<.0001
Lag 3 ILOB			–0.2097	–0.2104 <.0001	-0.0252 <.0001	0.1241
Lag 4 ILOB				0.00641	0.1175 0.729	<.0001
Wald Test	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

We now use information from ILOB to construct a trading strategy. We first sort sample firms into ILOB decile groups. The equal weighted return (EWR) and return to volatility (SD of return) is reported under two strategies: We begin investment 15 minutes after each ranking period, then hold each decile group for the following 15 and 30 minutes. Table 8 reports the holding period returns which more or less rise monotonically in ILOB levels confirming that the higher the strength of order imbalance, the stronger the information signal since the persistent bidder (seller) will be forced return in that direction. The difference between top and bottom ranked ILOB portfolio EWR and return to volatility are significant in favor of top ranked ones.

[4] See programming snippet 1 in appendix for sample code.

Table 8: Return and Reward to Variability Performance

ILOB Decile	15 minutes		30 minutes	
	EWR	Ret to SD	EWR	Ret to SD
Bottom 1	–0.031%	–0.046	–0.012%	–0.022
2	–0.019%	–0.027	–0.0005%	–0.112
3	–0.023%	–0.026	0.006%	0.001
4	0.006%	–0.006	0.008%	0.006
5	–0.006%	–0.018	0.005%	0.011
6	0.011%	0.013	0.007%	0.000
7	0.011%	0.022	0.011%	0.006
8	0.028%	0.035	0.010%	0.006
9	0.045%	0.051	0.012%	0.007
Top 10	0.052%	0.057	0.016%	0.007
Diff Top - Bottom	0.083%	0.102	0.028%	0.011
T-stats	6.87	7.130	4.01	2.360
p-value	<.0001	<.0001	<.0001	0.0181

3.2 Sentiment Data Analytics

We follow the steps described in section 2.3.2 to create a sentiment index from news coverage. We first collect a total of 5,462 top-story news related to SET50 from January 2014 to June 2018 from Reuters English news. We remove irrelevant and duplicate news, and read through each news piece to categorize news into 3 groups: good news, bad news, and neutral. Initial screening brings down the total news to 1,914: 1,125 (neutral), 527(good), 262(bad). The amount of news coverage is increasing in stock market capitalization and liquidity. Next in the data pre-processing and feature engineering process, we begin with the collection of good and bad news, we create two types of feature matrices from word proportion and TF-IDF. 6 We have a total of size 6,926 vocabulary. That is, the feature matrix dimension is [789 × 6,926] where each element represents the appearance of each word (column) in each document (row). Figure 4 shows the wordcloud visualization for good news and bad news.[5] Prominent words for good news are: profit, group business, and investment. Frequent terms found in bad news are: *profit*, *loss*, and *fall*.

[5] See Snippet 2 in appendix for Python code to word clouds. More codes are available on the authors' web portal.

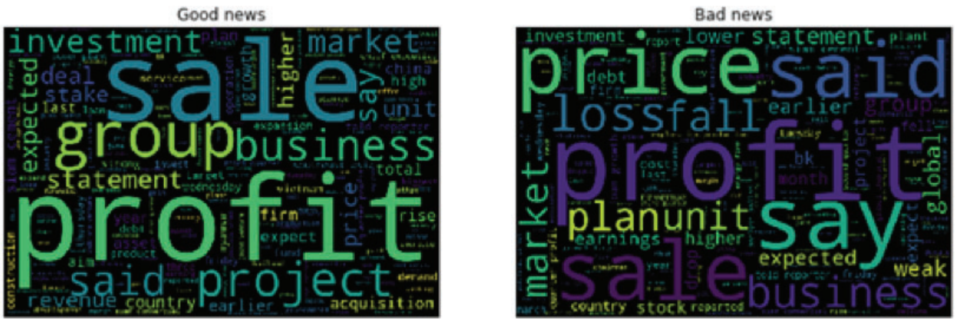

Figure 4: Data Visualization with Word Cloud

Next, we model the classifiers. The ratio split between training and test data is 80:20. The following classifers; a) Logistic regression (benchmark) b) Decision Tree, and c) Random Forests are estimated. The analysis begins with a logistic regression since the outcome of the model will provide numerical identification of important key words associated with good or bad news. As starting point, these “key” words are then used as reference nodes (parent nodes) of our decisions trees. Hence, we would expect overall accuracy of decision trees to be similar to the logistic approach. In random forest estimation, we select 80% tokens appearing in each row of the feature table randomly 300 times, producing 300 trees on average with depths of 41 which is substantially larger than the original tree. The trade-off is parsimony produced by simple trees compared to reduction of single model bias sampling error gained from forests.

Table 9: Words Associated with Good News and Bad News from Logistic Regression Model

Top 10 Words Associated with Good News and Bad News				
Good News			Bad News	
Word	Weight		Word	Weight
Contract	0.941109		Loss	–1.0088
Construction	0.616432		Profit	–0.965329
Signed	0.55993		Fall	–0.855269
Invest	0.557905		Versus	–0.543321
Development	0.474248		Drop	–0.529812
Group	0.468504		Production	–0.442947
Sign	0.422119		Shutdown	–0.438845
Agreement	0.406751		Weak	–0.406618
Project	0.395911		Quarterly	–0.388442
Venture	0.388836		Service	–0.367711

Table 9 reports the odds ratios of words that are associated with good news are: contract, construction, signed, investment, development; whereas words related to bad news are: loss, fall, drop, shut down. For example, the odds ratio of 0.94 on the word contract means that it is 0.94 times more likely that if the word appears in a document, it will be classified as good news. Out of our 6,926 vocabulary, we report only the most significant features associated with good or bad sentiment. Features that do not significantly contribute to information, receive negligible allocation weights. Table 10, evaluates the three models with four different metrics described earlier. We estimate each model first by using word proportion, and then by TF-IDF. For each modeling approach, all metric points to higher performance score using TF-IDF term frequency scheme. Our estimation validates the critique of using simple word proportions (percentage frequency of token in document) tend to deliver higher errors in term identification since words that are most often used in a document may not necessarily be the one conveying the sentiment message.

Table 10: Evaluation Metrics of Validation Data

Classifier	Word Proportion				TF-IDF			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.7083	0.5000	0.7100	0.5900	0.7639	0.8800	0.7600	0.8000
Decision Tree	0.7083	0.7400	0.7300	0.7300	0.7569	0.7600	0.7600	0.7600
Random Forest	0.7708	0.7600	0.7700	0.7400	0.8472	0.9000	0.8500	0.8600

From earlier discussion, the metric accuracy measures an overall model performance ie. the percentage of correct identification out of all identifications. Precision measures the percentage the model identifies correctly under positive classifications compared to sum of true and false positives. Recall is a ratio of true positive relative to true positive and false negative identifications. A model with high precision but low recall score suggests that it has lower ability to delineate between true and false negatives but better ability to differentiate true and false positives. For a scenario where false negative is more critical such as classifying a financially distressed company as highly liquid, the user would prefer to see a recall measure higher than precision measure. The F1-score can be considered a harmonic mean of precision and recall and is the preferred metric when the user deem identification of false negatives and false positives equally important. In Table 9, all three models can determine true positives above 75% of the time. Under logistic and random forest models, precision ratios are higher than recall suggesting that the models are better at minimizing false positives than minimizing false negatives. As expected, random forest classifiers outperforms logistic and decision tree approaches since the method employs a diverse set of classifiers in the root node, then reports averaging of multiple decision trees in the process reducing potential bias from any single root node. This is akin to bootstrapping to find the average value of random variable to reduce the sampling error.

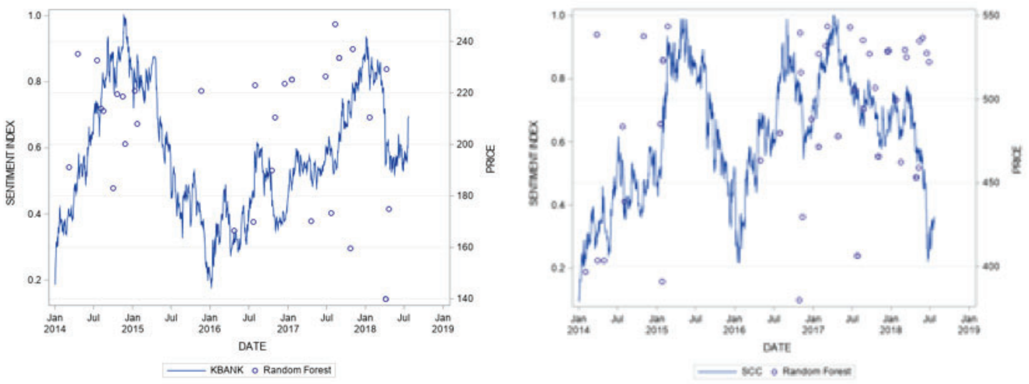

Figure 5: Time Series of Stock Price Plotting Against Sentiment Index

Using the random forest classifier, we allow the model to automatically generate sentiment score from the number of good to bad news that it can identify. Figure 7 plot two selected stock prices: Kasikorn Bank (KBANK) and Siam Cement (SCC) against a sentiment index. A high sentiment index score tends to be associated with relative higher prices. Given the limitation in the number of news story for any single stock, we find the strength of the correlation quite weak, less than 0.2 but with p value of only 0.07. In any case, the positive but weak association suggests that sentiment analysis is likely to be useful as a complementary signal of stock selection rather than a main signal. We decided against performing a long-short strategy designed follow the sentiment given too sparse data points to be able to deliver any valid statistical tests.

5. CONCLUSION

This article gives an introduction to the concept of Big Data and its computing environment. The transition to Big Data finance comes with both benefits and pitfalls. We describe key financial machine learning methods and provide examples of applications in equity trading using information in limit order book and sentiment data extracted from internet news. Technology provide access to ample real time dataset and speedy machines to clean up data noise and analyze key contents.

We draw similarity between traditional financial econometrics and terminology in data science. By and large, working with data:- big or small, requires the process of collection, analysis and extraction of value. Financial econometrics with big or small data need the user to train a model and fit a model that reduces the error of model fit. Big data may better capture patterns of non-linearity in social science data. However, in Big Data strategies, machines can continue to search and include a number of large variables in regressions to seek the model with the best fit. The procedure can lead to overfitting problems as well as irrational economic models from possible spurious correlations (Kolanovic and Krishnamachari, 2017). From their experiences, Aldridge (2013) and De Prado (2018) warn practitioners of hasty applications of Big Data techniques in trading. The authors note the length of time in years to develop profitable short-term trading strategies and that most attempts to develop them often lead to false positives and overfitting back tests.

For this reason, having teams of analysts who are trained both in financial economics and computing is preferable. Big data analytics often use abduction to seek the best explanation of a specific event or anomaly. The analysis may not be portable to generalize to other cases unlike the method of deduction in classical economics which begins with general theory then uses datasets to test theory. This observation is demonstrated in our empirical techniques in Big Data applied to equity trading. In the market microstructure strategy, a 15-minute buy and hold strategy of previous 15 minute top ranking limit order book intensity produces 0.052% return performance. When the holding period is extended to 30 minutes, the holding period return is further reduced to 0.016%. In creating a sentiment scores from text mining, we find weak albeit statistically significant correlation at 10% between the index and price levels. While confining our search to Reuters reduces the risk of noise, it also reduces the amount of available news observations subsequently leading to low correlation between sentiment and return index. Another possible explanation is the sentiment index created from stocks receiving high news coverage or reporters’ attention may not necessary be stocks with dominant events affecting value. On the contrary, stocks with low news coverage will generate insignificant or no change in sentiment index even if the paucity of events has substantial impact on value. Further analysis to measure the noise to signal ratio in the decision tree entropy process is required.

REFERENCES

Aase, KG., (2011). *Text mining of news articles for stock price predictions*, MSc Thesis. Norwegian University of Science and Technology.

Admati and Pfleiderer, (2001). *Noisytalk.com: Broadcasting opinions in a noisy environment*. Working Paper. Stanford University.

Aldridge, I., (2013). *High frequency trading: A practical guide to algorithmic strategies and trading systems*. Wiley.

Antweiler, W., Frank, M.Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance 59*(3), 1259–1294.

Barberis, N., Shleifer, A., Vishny, R. (1998). A model of investor sentiment. *Journal of Financial Economics, 49*, 307–343.

Blume, L., Easley, M., and O'Hara, M., (1994). Market Statistics and Technical Analysis: The Role of Volume, *Journal of Finance, 49*(1), 153–181.

Bollen, J., Mao, H., Zeng, X., (2011). Twitter mood predicts the stock market. *Journal of Computational Science 2*(1), 1–8.

Da, Z., Engelberg, J., and Gao, P., (2014).The sum of all FEARS investor sentiment and asset prices, *Review of Financial Studies, 28*(1), 1–32.

De Prado, M.L., (2018). *Advances in financial machine learning*, Wiley.

Easley, M., De Prado, M.L., and O'Hara, M., (2016), Discerning information from trade data, *Journal of Finance and Economics, 120*(2), 269–286.

Gao, L., Han, Y., Li, S.Z., and Zhou, G., (2018). Market intraday momentum, *Journal of Financial Economics, 129*(2), 394–414.

Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J, and Howison, S.D., (2013). Limit order books, *Quantitative Finance, 13*(11), 1709–1742.

Hansen, S., McMahon, M, and Prat, A., (2014). Transparency and deliberation within the FOMC, a Computational Linguistics Approach, *CEP Discussion Papers DP 1276*, Centre for Economic Performance, London School of Economics.

Hoberg, G., and Phillips, G., (2016). Text-based Network Industries and Endogenous Product Differentiation. *Journal of Political Economy 124*(5): 1423–65.

Hollifield, B., Miller, RA., Sandas, P., and Slive, J., (2006). Estimating gains from trade in limit order markets, *Journal of Finance, 61*(6), 2753–2804.

Kolanovic, M., and Krishnamachari, R.T., (2017). Big Data and AI Strategies: Machine learning and alternative data approach to investing, *Global Quantitative and Derivatives Strategy*, JP Morgan.

Krauss, C., Do, XA., Huck., N., (2017). Deep Neural networks, gradient boosted trees, random forests: Statistical arbitrage on the S&P500, *European Journal of Operational Research, 259*(2), 689–702.

Kyle, A., (1989). Informed speculation with imperfect competition. *Review of Economic Studies 56*, 317–355.

Mullainathan, S., and Spiess, J., (2017). Machine learning: An applied econometric approach, *Journal of Economic Perspectives, 31*(2), 87–106.

Rosu, I., (2009). A dynamic model of the limit order book. *Review of Financial Studies, 22*(11), 4601–4641.

Schulmeister, S. (2009), Profitability of technical stock trading: Has it moved from daily to intraday data?, *Review of Financial Economics, 18*(4), 190–201.

Tantaopas, P., Padungsaksawasdi, C., Treepongkaruna, S., (2016). Attention effect via internet search intensity in Asia Pacific stock markets, *Pacific Basin Finance Journal, 38*, 107–124.

Tetlock, PC, (2007). Giving content to investor sentiment: The role of media in the stock market, *Journal of Finance, 62*(3), 1139–1168.

Appendix of Programming Snippets

There are a number of specialized software that can handle data analytics. In this article we provide snippets for two popular ones: Python, and SAS. Python is an open source program with multi-threading capability. The program has gained popularity in recent years from the availability and growing number of free software machine learning library known as scikit-learn, which is under active development. SAS is licensed software with comprehensive data mining and statistics facilities and high computational performance. It has recently launched SAS Viya providing simple codes to handle algorithms used in text mining and big data analytics processing through its cloud analytic server (CAS). See more of our codes available on https://github.com/swatakit.

Snippet 1 Machine learning procedure in SAS Viya

```
proc logselect data=mycas.LOB_;
  model Class(up='1')=ILOB LILOB L2LILOB L3ILOB L4ILOB;
  fraction test = 0.75 validate = 0.25 seed = 12345678 ;
  selection method=forward;

run;
```

A number of machine learning statistical procedures are available in SAS Viya. For example, LOGSELECT procedure fits and performs model selection for logistic regression models in SAS Viya. The forward option selects the best model based on training observations. Other methods described in section 3.2.2 such as the least absolute shrinkage and selection operator (LASSO) method can be replaced.

Snippet 2 Visualizing data with word cloud

```
#convert word tokens to feature vector: proportion method
def tokens_to_vector(tokens,label):
  x=np.zeros(len(word_index_map)+1)
  for t in tokens:
  i = word_index_map[t]
  x[i] += 1
  x = x/x.sum()
  x[-1] = label
  return x
```

```
# visualize the data
def visualize(data,title):
  words = ''
  for row in data:
  for msg in row:
  words += msg + ' '
  wordcloud = WordCloud(width=600, height=400).generate(words)
  plt.imshow(wordcloud)
  plt.title("{}".format(title))
  plt.axis('off')
  plt.show()
```

After the feature matrix is created, the code above produces word clouds as in Figure 4. The feature matrix is then used for approximation of logistic, decision tree, and random forests models.