COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

STOCK PRICE CHANGE PREDICTION USING NEWS TEXT MINING

Marcelo Beckmann

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientadores: Nelson Francisco Favilla Ebecken
Beatriz de Souza Leite Pires de Lima

Rio de Janeiro
Janeiro de 2017

STOCK PRICE CHANGE PREDICTION USING NEWS TEXT MINING

Marcelo Beckmann

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

____

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

____

Prof^{a}. Beatriz de Souza Leite Pires de Lima, D.Sc.

____

Prof. Elton Fernandes

____

Prof. Helio José Corrêa Barbosa

____

Prof^{a}. Solange Guimarães

____

Prof^{a}. Regina Celia Paula Leal Toledo

RIO DE JANEIRO, RJ, BRASIL
JANEIRO/2017

Stock Price Change Prediction Using News Text Mining/Marcelo Beckmann. - Rio de Janeiro: UFRJ/COPPE, 2017.

IX, 143 p.: il.; 29,7 cm.

Orientadores: Nelson Francisco Favilla Ebecken
Beatriz de Souza Leite Pires de Lima

Tese (doutorado) - UFRJ/COPPE/ Programa de Engenharia de Civil, 2017.

Referências Bibliográficas: p. 120-135.

1. Finanças. 2. Mineração de Textos. 3. Mineração de Dados. I. Ebecken, Nelson Francisco Favilla *et al*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Acknowledgments

To God.

To my mother, who taught me to love the people and the world, and to be who I am.

To my wife, my love, my life.

To my sons, my light, my inspiration.

To my friends, Eduardo Nicodemos and Julia M. R. F. Santos, which devote their careers to high frequency algorithmic trading, and since the beginning they helped me to have a clear understanding and good insights in this area.

To Andrea Mussap, who kindly volunteered to review the English grammar and textual structure of this thesis.

To my friends from COPPE/Federal University of Rio de Janeiro. In special Mauricio Onoda and Cristian Klen, without their friendship, competence and clever ideas, it would never have been possible to reach this point.

To my teachers.

A special thanks to the giants whose shoulders I stood on all these years. They showed me a way to go, self-learning, research, think ahead, and propagate good science. These giants are my professors from COPPE/Federal University of Rio de Janeiro. Thank you very much, Nelson Francisco Favilla Ebecken, Beatriz de Souza Leite Pires de Lima, and Alexandre Gonçalves Evsukoff.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

PREDIÇÃO DA VARIAÇÃO DE PREÇOS DE AÇÕES UTILIZANDO MINERAÇÃO DE TEXTOS EM NOTÍCIAS

Marcelo Beckmann

Janeiro/2017

Orientadores: Nelson Francisco Favilla Ebecken
Beatriz de Souza Leite Pires de Lima

Programa: Engenharia Civil

Com o advento da Internet como um meio de propagação de notícias em formato digital, veio a necessidade de entender e transformar esses dados em informação.

Este trabalho tem como objetivo apresentar um processo computacional para predição de preços de ações ao longo do dia, dada a ocorrência de notícias relacionadas às companhias listadas no índice Down Jones. Para esta tarefa, um processo automatizado que coleta, limpa, rotula, classifica e simula investimentos foi desenvolvido. Este processo integra algoritmos de mineração de dados e textos já existentes, com novas técnicas de alinhamento entre notícias e preços de ações, pré-processamento, e assembleia de classificadores. Os resultados dos experimentos em termos de medidas de classificação e o retorno acumulado obtido através de simulação de investimentos foram maiores do que outros resultados encontrados após uma extensa revisão da literatura. Este trabalho também discute que a acurácia como medida de classificação, e a incorreta utilização da técnica de validação cruzada, têm muito pouco a contribuir em termos de recomendação de investimentos no mercado financeiro.

Ao todo, a metodologia desenvolvida e resultados contribuem com o estado da arte nesta área de pesquisa emergente, demonstrando que o uso correto de técnicas de mineração de dados e texto é uma alternativa aplicável para a predição de movimentos no mercado financeiro.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor in Science (D.Sc.)

STOCK PRICE CHANGE PREDICTION USING NEWS TEXT MINING

Marcelo Beckmann

January/2017

Advisors: Nelson Francisco Favilla Ebecken
Beatriz de Souza Leite Pires de Lima

Department: Civil Engineering

Along with the advent of the Internet as a new way of propagating news in a digital format, came the need to understand and transform this data into information.

This work presents a computational framework that aims to predict the changes of stock prices along the day, given the occurrence of news articles related to the companies listed in the Down Jones Index. For this task, an automated process that gathers, cleans, labels, classifies, and simulates investments was developed. This process integrates the existing data mining and text algorithms, with the proposal of new techniques of alignment between news articles and stock prices, pre-processing, and classifier ensemble. The result of experiments in terms of classification measures and the Cumulative Return obtained through investment simulation outperformed the other results found after an extensive review in the related literature. This work also argues that the classification measure of Accuracy and incorrect use of cross validation technique have too few to contribute in terms of investment recommendation for financial market.

Altogether, the developed methodology and results contribute with the state of art in this emerging research field, demonstrating that the correct use of text mining techniques is an applicable alternative to predict stock price movements in the financial market.

Table of Contents

Chapter 1 – Introduction

With the advent and popularization of the Internet during the '90s, the news articles that before were available in the day after, printed in paper, started to be available as soon as possible, in digital format, at the velocity that financial market needs. During the same decade, the developments in computing, inferential statistics, artificial intelligence, machine learning, information retrieval, natural language processing, and linguistics culminated in the creation of data mining and text mining as emerging technologies.

The advances in data mining and text mining, allied with the velocity and the way the news articles are published, created opportunities to use text mining applied to financial market prediction (TMFP). Nevertheless, to make possible computers to interpret news articles at the right time and generate profit in financial markets, an interdisciplinary field of research has been created. The Venn diagram in Figure 1 describes the three disciplines involved in this emerging field.

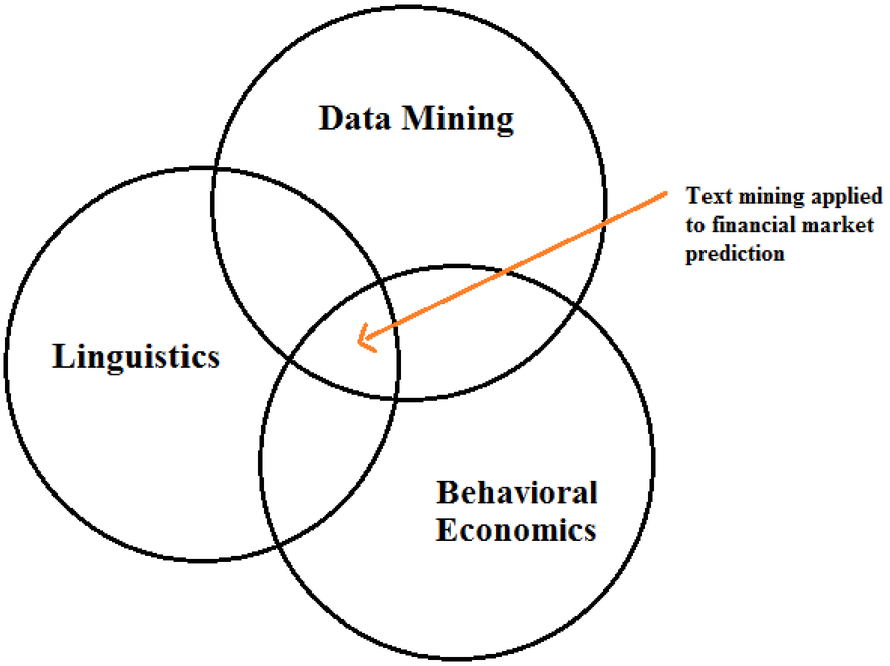


Figure 1 - Venn diagram describing the intersection of disciplines involved in this work.

TMFP is supported by Behavioral Economics (BE) theories, which analyse the psychological, social, cognitive, and emotional aspects of human behaviour when taking investment decisions. BE claims that humans can make irrational decisions that lead to discrepancies and market inefficiencies. Due to this inefficiency, the stock prices cannot reflect in real time the changes in the world, creating an opportunity for predictive techniques like data mining and text mining.

The main objective in this work is to prove that data mining and text mining can be used to automatically interpret news articles and learn patterns to predict the movement in the stock markets, providing in this way investment recommendations to be used by traders and automated trading systems to achieve returns. To accomplish this objective, a complete process of data mining and text mining was developed to predict the price movements in the stock market for the 30 companies listed in the Down Jones Industrial Index (DJIA) along the day (intraday). Due to the complex and unstable nature of financial markets, the traditional data mining algorithms were not enough to make correct predictions, and then a new data preparation technique to deal with imbalanced class problem, and a classifier ensemble technique to remove class overlapping were proposed in this work.

The experiment results are demonstrated in terms of classification measures such as Accuracy, Precision, Recall, AUC, G-Mean, and F-Measure; and an investment simulator was developed to validate the predictions generated by the classifier. The classifier measures and the cumulative return obtained with the investment simulation outperformed the results existing in the reviewed literature.

In this work, an extensive review of the literature related to TMFP was conducted, and problems like the use of Accuracy as classification measure, lack of information about the model evaluation, and incorrect use of cross validation were identified and will be discussed.

This thesis is organized as follows: Chapter 2 presents the data mining techniques used in this work. Chapter 3 presents the financial economics background that supports the application of text mining in financial market prediction, followed by an extensive review of the literature on this subject. Chapter 4 introduces the proposed methodology, followed by the experiments and discussion in Chapter 5. Finally, Chapter 6 concludes the thesis and proposes new developments in the future. The meaning of acronyms and financial terms can be found in Table 19 Appendix A.

Chapter 2 – Data Mining

The great volume of data generated nowadays and the expected growth in the next years bring new challenges to explore, understand, and transform all this data in useful information. The use of data mining techniques, also known as Knowledge Discovery in Databases (KDD), play an important role to deal with these new challenges.

Data mining is an interdisciplinary field of computer science. This term was coined in the ‘90s, and mainly involves inferential statistics, artificial intelligence, machine learning, and database systems techniques that were developed in the previous decades. As demonstrated in Figure 2, data mining is divided in supervised and unsupervised learning tasks. The supervised tasks are related to analyzing and learning from examples (also known as rows, records, or instances) with a previous identification (the class), and they aim to classify the new examples, using the concepts learned previously. In regression tasks, the learning algorithm uses a numerical value (integer or continuous) instead of a class, and the outcome of this algorithm is also a numerical value. In unsupervised learning, there is no class or numerical value associated with the examples from the dataset under analysis, and the clustering and association are exploratory tasks looking for unknown patterns, groups, and similarities.

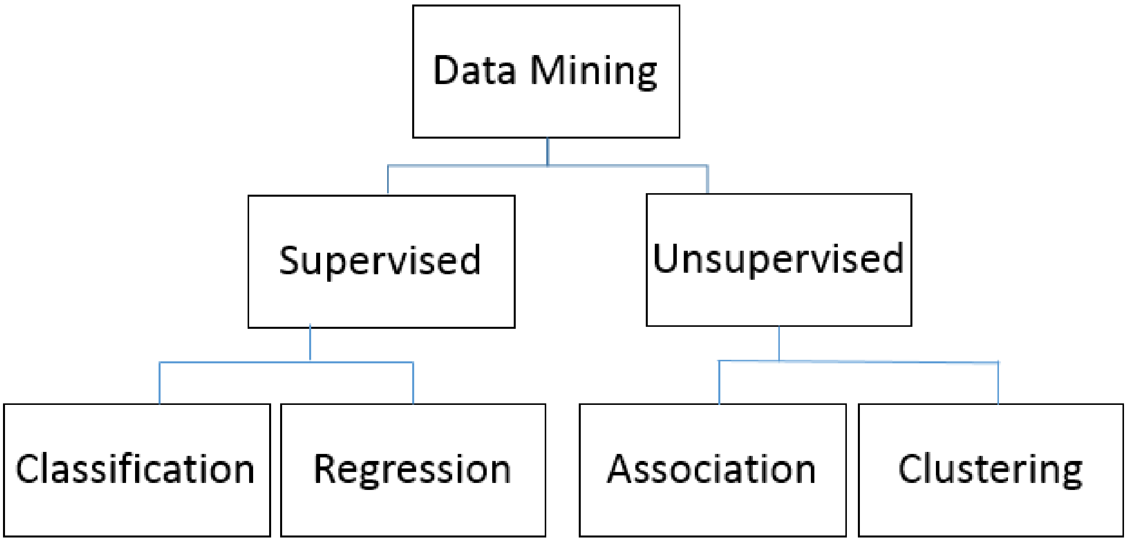


Figure 2 - Taxonomy of data mining tasks

This work focuses on supervised learning and classification tasks. In the next sections, the pre-processing, classification algorithms, evaluation measures, and text mining techniques will be described.

2.1 Pre-Processing Techniques

The proper use of classification algorithms is not enough to deliver a data mining product. In fact, there are several initiatives for an effective data mining process, the most well-known is the Cross Industry Standard Process for Data Mining (CRISP-DM), with its main process depicted in Figure 3.

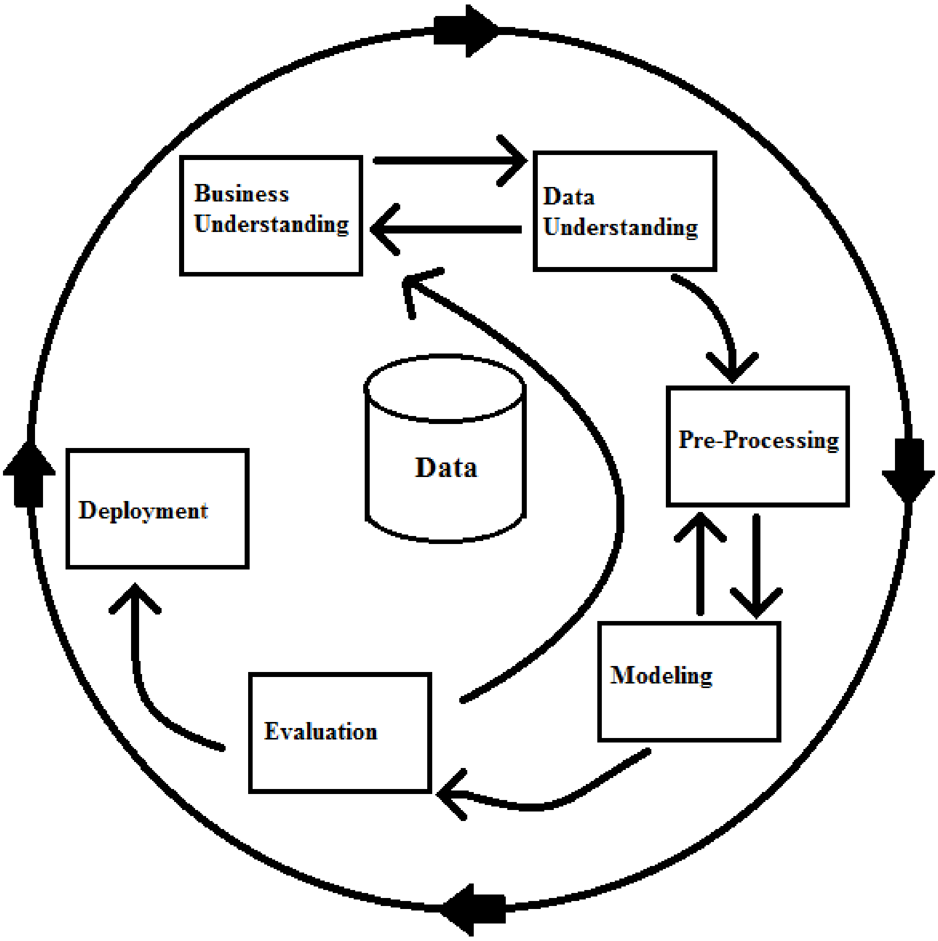


Figure 3 - The CRISP-DM process.

One of the most important phases in CRISP-DM is the pre-processing, or data preparation. The main goal of this phase is to transform and adjust the data for the modeling phase, in accordance with the input of data understanding phase, which generated descriptive statistics through exploratory analysis.

2.2 Classification Algorithms

Within the data mining context, classification is the capacity to identify objects and predict events. It is a modeling activity that uses machine learning algorithms, and it is considered a supervised learning task, because each example in a dataset must be labeled according to its features.

One of the first proposals of automatic classification was inspired in the capacity of live individuals to identify objects and events in their environments. This algorithm implemented a series of weighted connections that replicate a neurological system to create a network of artificial neurons (McCulloch & Pitts, 1943), and the term Artificial Neural Networks (ANN) was coined. Since then, the classification algorithms evolved and diversified, with application in all areas of human knowledge.

The classification process is divided into two phases: training and testing. For the training phase, a machine-learning algorithm is used on a dataset, entitled training set, which consists of examples. Each example consists of one or more attributes and a specific attribute, which contains the label that associates the example to a pre-defined class. The need for a pre-defined label in the training stage makes the classification a supervised learning activity.

The training algorithm will generate a predictive model based on the relationship between the attribute values and the class the instance belongs, that is, the algorithm infers that certain values are associated with certain classes. For this task, there are multiple learning techniques, and according to (Frank, et al., 2016), they are mainly categorized as Bayes, Functions (e.g., SVM, Neural Networks), Lazy Algorithms (e.g., KNN), Meta Classifiers (e.g., Bagging and Boosting), Decision Rules, and Decision Trees.

During the test phase, new examples of unknown class will be identified (labeled), using the predictive model generated in the training phase to decide which class the new example belongs, given its attribute values, thus completing the process of machine learning and classification. At this stage, it is necessary to compute some measures that assert the quality of the predictive model obtained during the training phase. These measures are known as classification measures and they will be described in section 2.3.

The training and testing process is also known as predictive model selection. Among the various model selection methods, the simplest technique consists in the separation of a portion (normally 70%) of the dataset for training, and the remaining for test.

Another model selection method is the cross-validation, which aims to improve the assessment of the predictive model by testing the classifier performance in unknown instances. The operation consists in partitioning the data set in *f* equal parts (usually *f = 10*), and separate *f-1* parts of the dataset for training and one part for testing. The process is repeated *f* times, and on each iteration a different part from the dataset is separated for testing, and the remaining for training. At the end, the average of the classification measures obtained on each iteration is taken.

Nowadays, there is a great number of classification methods and several variations of them. The objective of this section is to describe the classification methods used in this work. For further information about other methods, see (Wu, et al., 2007)

2.2.1 Support Vector Machine

The Support Vector Machine algorithm (SVM) is a supervised learning technique applicable for classification and regression tasks. With its bases initially launched by (Vapnik & Lerner, 1963) and enhanced in the ‘90s at AT&T Bell Labs, it is grounded in the theory of statistical learning and the principle of minimization of structural risk, which argues that the less complex the model, the better the ability of generalization this model will have.

Originally, the SVM was developed for linear classification problems with two classes separable by a margin, where the margin means the minimum distance of two hyperplanes separating the classes. The SVM learning algorithm searches for an optimal hyperplane separation where it maximizes the width of the margin of separation, which minimizes the structural risk, giving the model a great ability to generalize.

The solution or predictive model of SVM is only based on the data points that delimit the margins’ edge. These points are called support vectors. Another important feature is that the calculation of the structural risk does not take into account the dimensionality of the training set (2), which allows the SVM to be applied to high-dimensional problems such as image recognition and text mining. The SVM also does not take into account how the data is distributed. However, the algorithm did not solve nonlinear problems, until

(Boser, et al., 1992) suggested a way to make a nonlinear SVM classifier using the kernel trick (Aizerman, et al., 1964).

The vast majority of classification problems have not separable classes, which also prevented the acceptance and application of SVM, which was initially designed to deal with completely separable classes. The solution was proposed by (Cortes & Vapnik, 1995), who introduced a relaxing constraint variable (12), allowing hyperplanes with flexible margins and finally making the SVM a viable and successful algorithm.

Model Formulation

Different of other methods based on the error minimization, SVM searches for a model structure less complex as possible, in order to not trespass a pre-fixed error level, and with the aim to minimize the structural risk, represented by the functional equation:

$$R(f) = R_e(f) + R_s(f) \tag{1}$$

Where:

- $R(f)$ is the total risk;
- $R_e(f)$ is the empiric risk relative to the errors and noise from the training set;
- $R_s(f)$ is the model's structural risk, which is calculated as:

$$R_s(f) = \frac{\sqrt{h\left(\ln(\frac{2N}{h})+1\right)} - \ln(\frac{\eta}{4})}{N} \tag{2}$$

Where:

- N is the number of examples in the training set;
- $(1-\eta)$ is the statistical confidence from the result of (2), generally 5%;
- h is an integer called Vapnik-Chervonenkis dimension, which measures the predictive capacity of a family of functions applied in one model. For example,

for binary classification problems, *h* is the number of required points to perform the separation, according to the family of functions used in the problem.

Definition of structure and model parameter

Given the training set $T = \{(x(t), y(t)) \mid x(t) \in \Re, y(t) \in \{-1,1\}\}_{t=1}^{N}$, where *y(t)* can be *1* or *-1*, indicating to which class the point *x(t)* belongs, and each *x(t)* is a p-dimensional vector, the SVM searches for a hyperplane with a maximized margin among an infinity of existing hyperplanes (Figure 4), which splits the points that belong to *y(t) =1* and *y(t) = −1*. This separation surface is defined by the hyperplane:

$$d(x) = \langle w, x(t) \rangle + b = 0 \qquad (3)$$

Where:

- $x(t)$ is the input vector;
- w is the normal vector to the hyperplane, which defines the width of margin, because the bigger the angle, the bigger the margin (Figure 5), since the restrictions in (4) do not be violated
- b is a bias
- $\frac{-b}{\|w\|}$ is the distance from the hyperplane to the origin (Figure 5)

For the linear algorithm, the problem must be separable, considering the following constraints:

$$\langle w, x(t) \rangle + b \geq 0,\ if\ \ y(t) = 1 \qquad (4)$$
$$\langle w, x(t) \rangle + b < 0,\ if\ \ y(t) = -1$$

The linear decision surface is calculated as $\frac{2}{\|w\|}$, and the minimization of the 2-norm of *w* will make the separation margin to be maximized. The data points on the edge of separation margin are called support vectors α. The support vectors are the final product from the SVM algorithm, also known as model parameter or simply model. The support vectors α are calculated from b and w, through a quadratic optimization (Figure 6), which will be detailed in the next section.

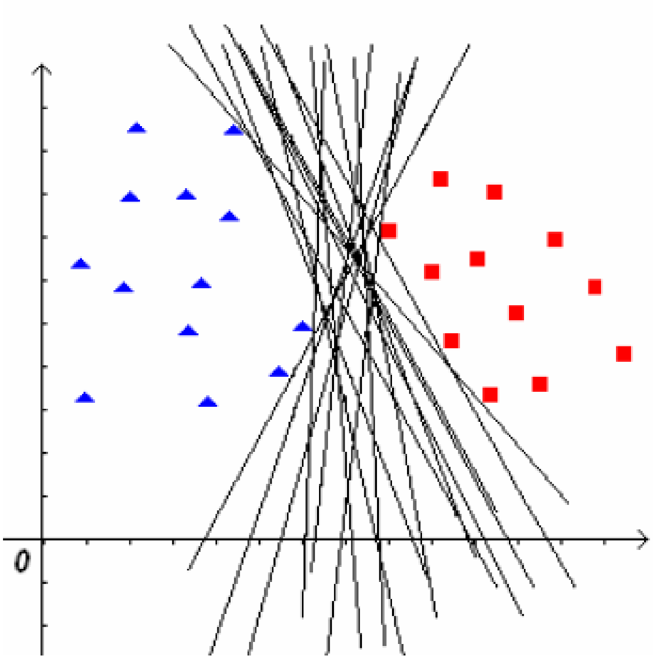
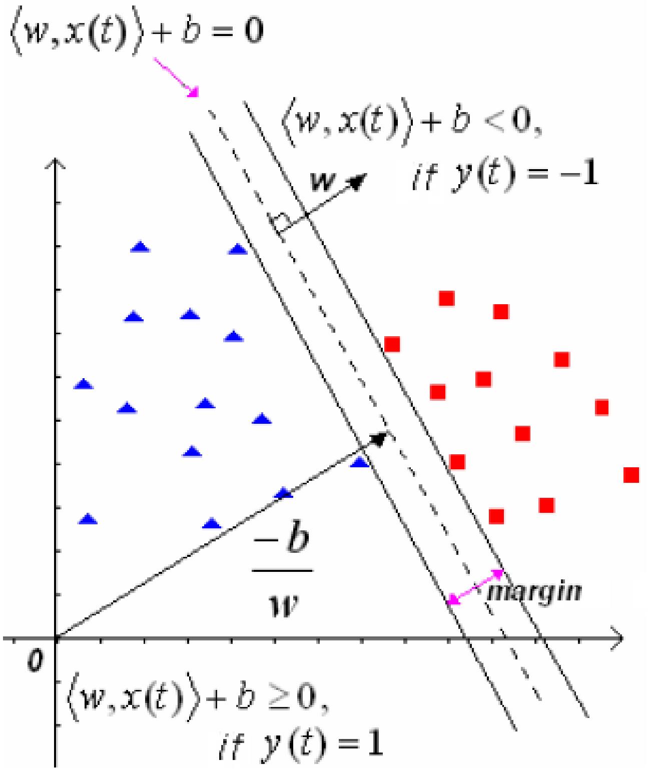
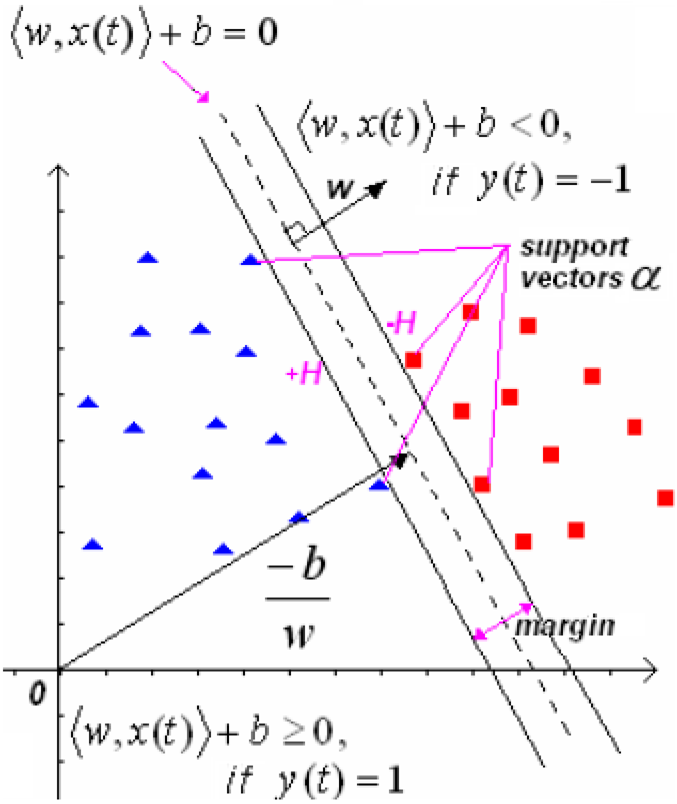


Figure 4 - Infinite separation surfaces in a binary classification problem.



$\langle w, x(t)\rangle + b = 0$
$\langle w, x(t)\rangle + b < 0,$
$if\ y(t) = -1$
w
$\frac{-b}{w}$
margin
0
$\langle w, x(t)\rangle + b \geq 0,$
$if\ y(t) = 1$

Figure 5 - Searching for a separation surface with a maximized margin and less structural risk. The higher *w*, the larger the margin.



$\langle w, x(t)\rangle + b = 0$
$\langle w, x(t)\rangle + b < 0,$
$if\ y(t) = -1$
w
support
vectors α
-H
+H
$\frac{-b}{w}$
margin
0
$\langle w, x(t)\rangle + b \geq 0,$
$if\ y(t) = 1$

Figure 6 - The result of ||w|| minimization are the support vectors.

Parameter adjusting algorithm

To obtain an optimal hyperplane $|| w ||$ and b must be minimized, causing the maximization of margin *w*, but subject to the constraints (4). This problem is n^p difficult because it is a non-convex optimization, but substituting $|| w ||$ for w (5), this will not change the solution, but at least now there is a convex optimization to be solved, which is a quadratic optimization problem with constraints.

$$\min_{w} J(w) = \frac{1}{2} || w ||^2, subjectto \ (4) \tag{5}$$

The primal form of Lagrange is obtained after applying the constraint on the equation above:

$$L_p(w,b,\alpha) = \frac{1}{2} || w ||^2 - \sum_{t=1}^{N} \alpha_t (y(t)(\langle w, x(t) \rangle + b) - 1) \tag{6}$$

To solve the problem above, the derivative of L_p relative to *w* and *b* must be equalized to zero:

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{t=1}^{N} \alpha_t y(t) x(t) \text{ ,and}$$
$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{t=1}^{N} \alpha_t y(t) = 0 \tag{7}$$

Replacing *w* and *b* in the primal Lagrangean **(6)**, the dual formulation is obtained, this time maximizing the margins:

$$\max_{\alpha} L_D(\alpha) = \sum_{t=1}^{N} \alpha_t - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y(i) y(j) \langle x(i), x(j) \rangle),$$

$$\text{subject to } \alpha_t \geq 0 \text{, e } \sum_{i=1}^{N} \alpha_i y(t) = 0 \qquad (8)$$

Finally, the discriminatory function starts to be calculated from the support vectors α :

$$d(x) = \sum_{t=1}^{N} \alpha_t y(t) \langle x, x(t) \rangle + b \begin{cases} \geq 0, if\ x \in \varpi + \\ < 0, if\ x \in \varpi - \end{cases} \qquad (9)$$

Parameter adjusting for nonlinear problems

The algorithm seen so far is only applicable to linear problems. For surfaces with nonlinear separation (Figure 7), the kernel trick was proposed by (Boser, et al., 1992) to maximize the margins of hyperplanes. The resulting algorithm is formally similar, except that all inner product is replaced by a nonlinear kernel function:

$$\max_{\alpha} L_D(\alpha) = \sum_{t=1}^{N} \alpha_t - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y(i) y(j) K(x(i), x(j))$$

(10)

$$d(x) = \sum_{t=1}^{N} \alpha_t y(t) K(x(i), x(j)) + b \begin{cases} \geq 0, if\ x \in \varpi + \\ < 0, if\ x \in \varpi - \end{cases}$$

Being, $K(x(i), x(j)) = \langle \Phi x(i), \Phi x(j) \rangle$

This allows the algorithm to adjust the hyperplane with a maximized margin in a transformed space of infinite dimensionality. Several functions can be applied to Φ , the most common are:

- Linear: $\Phi = x_i{}^T x_j$
- Homogeneous Polynomial: $\Phi = (\gamma x_i{}^T x_j)^d , \gamma > 0$
- Inhomogeneous Polynomial: $\Phi = (\gamma x_i{}^T x_j + 1)^d , \gamma > 0$
- Radial Basis: $\Phi = \exp(-\gamma \| x_i - x_j \|^2)), \mathit{for}\ \gamma > 0$ (11)
- Gaussian Radial Basis: $\Phi = \exp(\frac{-\gamma \| x_i - x_j \|^2}{2\sigma^2})$
- Sigmoid: $\Phi = \tanh(\gamma x_i^T x_j + r), \mathit{for\ any}\ \gamma > 0, \mathit{and}\ r < 0$

Here, γ and *d* are parameters provided by the user. The *d* parameter denotes the degree of a polynomial. The most common degree is *d=2* (quadratic), since larger degrees tend to over fit on Natural Language Processing (NLP) problems. The γ parameter (Gamma) is present in most functions in (11), and it is used to control the shape of peaks where the points raise. For example, in a problem with no linear separability between the classes, as shown in the 2-dimension plot from Figure 7, if the green points raise, the 2-dimension figure is transformed in a 3-dimension figure, then it is possible to separate green and red points with another plane (a hyperplane).

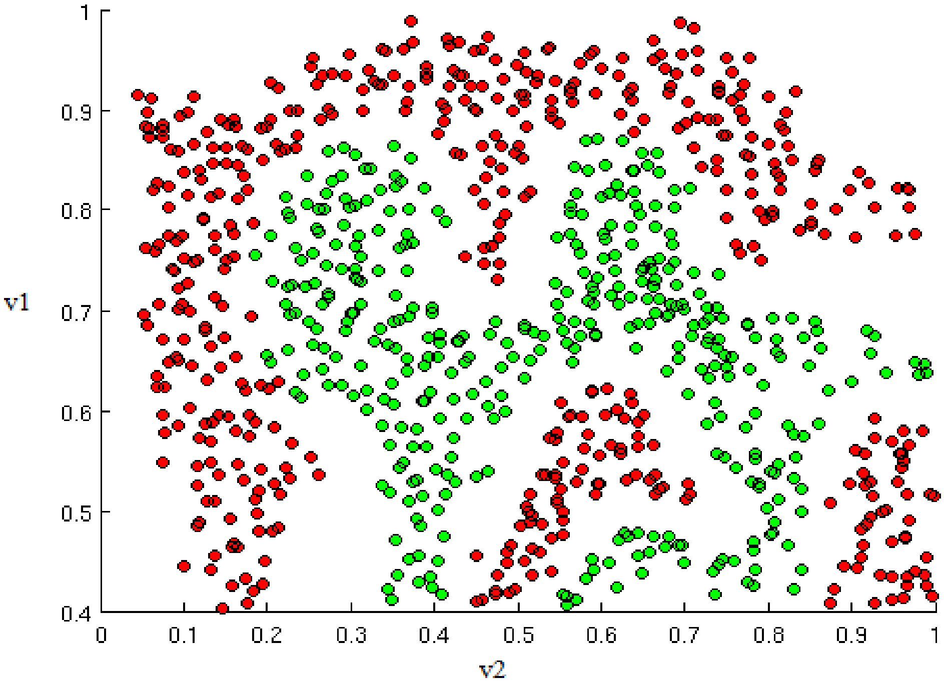


Figure 7 - A two-class dataset with non-linear separation (Ng, et al., 2010-2012)

A small γ gives a pointed bump, and a large γ gives a softer, broader bump. Therefore, a small γ tends to return low bias and high variance, while a large γ tends to return higher bias and low variance.

Flexible Margin

The constraints imposed in (4) don't allow the application of SVM in most of the existing classification problems in the real world, where classes cannot be separated completely. The solution was proposed by (Cortes & Vapnik, 1995), and it introduces a relaxing constraint variable $\xi = (\xi_{t=1},\dots,\xi_N),\xi > 0$, and then the margin constraints now are calculated as:

$$\begin{gathered} y(t)(\langle w, x(t)\rangle + b) \geq 1 - \xi_t \\ \min_{w,\xi} \min J(w,\xi) = \frac{1}{2}\|w\|^2 + c\sum_{t=1}^{N}\xi_t \end{gathered} \qquad (12)$$

Being c an additional constraint to the Lagrangean multipliers to penalize the classification errors.

User Parameter Optimization

The c variable used to penalize classification errors in the flexible margin, together with γ from the kernel functions demonstrated in (11), are parameters to be provided by the user. For large values of c , the margin adjusting (12) tends to choose a smaller-margin hyperplane. Conversely, small values of c will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of *c*, the algorithm tends to return more misclassified examples, even if the training data is linearly separable. The choice of *c* is a vital step and a good practice in the use of SVM, as structural risk minimization is partially implemented via the tuning of *c*.

The right choice of these parameters is considered the weakness of SVM, as a wrong set of parameters makes the SVM to perform poorly. The solution for this problem can be an optimization procedure to find the better set of values according to the dataset under study. (Hsu, et al., 2003) provides a practical guide to SVM, and proposes the use of a grid search to find the best values of c and γ, but first, use the linear kernel function, and compare the results with other functions. The grid search consists of using exponentially growing sequences of (c , γ), for example $c = 2^{-5}, 2^{-3}, ..., 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, ..., 2^{3}$. The pair with the best Accuracy after a cross-validation will be picked.

2.2.2 K Nearest Neighbors

The k Nearest Neighbor (KNN) is a supervised classifier algorithm, and despite its simplicity, it is considered one of the top 10 data mining algorithms (Wu, et al., 2007).

It creates a decision surface that adapts to the shape of the data distribution, making possible to obtain good Accuracy rates when the training set is large or representative. The KNN was introduced by (Fix & Hodges, 1951), and it was developed with the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine.

The KNN is a nonparametric lazy learning algorithm. It is nonparametric because it does not make any assumptions on the underlying data distribution. Most of the practical data in the real world does not obey the typical theoretical assumptions made (for example, Gaussian mixtures, linear separability, etc.). Nonparametric algorithms like KNN are more suitable on these cases (Dasarathy, 1991), (Duda, et al., 2001).

It is also considered a lazy algorithm. A lazy algorithm works with a nonexistent or minimal training phase, but with a costly testing phase. For KNN this means the training phase is fast, but all the training data is needed during the testing phase, or at the least, a subset with the most representative data must be present. This contrasts with other techniques like SVM, where one can discard all nonsupport vectors.

The classification algorithm is performed according to the following steps:

1. Calculate the distance (usually Euclidean) between an x_i instance and all instances of the training set T;
2. Select the k nearest neighbors;
3. The x_i instance is classified (labeled) with the most frequent class among the k nearest neighbors. It is also possible to use the neighbors' distance to weigh the classification decision.

The value of *k* is training-data dependent. A small value of *k* means that noise will have a higher influence on the result. A large value makes it computationally expensive and defeats the basic philosophy behind KNN: points that are close might have similar densities or classes. Typically, in the literature odd values are found for *k*, normally with *k = 5* or *k = 7*, and (Dasarathy, 1991) reports *k=3* allowing to obtain a performance very close to the Bayesian classifier in large datasets. An approach to determine *k* as a function (1) from the size of data *m* is proposed in (Duda, et al., 2001).

$$k = odd(\sqrt{m}) \tag{13}$$

The algorithm may use other distance metrics besides Euclidean (Sidorov, et al., 2014), (Argentini & Blanzieri, 2010), (Boriah, et al., 2007), (Wilson & Martinez, 1997).

2.3 Evaluation Measures

In supervised learning, it is necessary to use some measure to evaluate the results obtained with a classifier algorithm. The confusion matrix from Figure 8, also known as contingency table, is frequently applied for such purposes, providing not only the count of errors and hits, but also the necessary variables to calculate other measures.

The confusion matrix can represent either two class or multiclass problems. Nevertheless, the research and literature related to imbalanced datasets is concentrated in problem with two classes, also known as binary or binomial problems, where the less frequent class is named as positive, and the remaining classes are merged and named as negative. The confusion matrix must be a square matrix, and the main diagonal indicates the classifier hits, while the secondary diagonal indicates the errors.

	Positive prediction	Negative Prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

Figure 8 - Confusion Matrix

A systematic study about the evaluation measures applied to classification tasks can be found in (Sokolova & Lapalme, 2009). The classification measures presented in this section will be used to demonstrate the experiments results in Chapter 5. These results will be multiplied by 100 as this is a common approach in the state of the art.

2.3.1 Error Rate and Accuracy

Some of the most known measures derived from this matrix are the Error Rate (14) and the Accuracy (15). Both are complementary to 100%, e.g., if Accuracy is 67%, the Error Rate is 33%, and vice-versa.

$$Error = \frac{FP + FN}{TP + FN + FP + TN} \tag{14}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{15}$$

A way to define a good classification threshold in terms of Accuracy is comparing the Accuracy results with a random classifier, e.g., flipping a coin to take decisions in a binary problem. Classifiers with Accuracy lower or close to 50% are considered lower or close to a random classifier. This evaluation assumes both classes have 50% of distribution, which normally is not possible to obtain in real world problems.

By analyzing the equations (14) and (15), it is possible to notice that these measures do not consider the number of examples distributed between the positive and negative classes, and such measures are not appropriated to evaluate imbalanced datasets (Ling, et al., 2003), (Weis, 2004), (He & Garcia, 2009) , (He. & Ma, 2013), (Ali, et al., 2013). A complete discussion about Accuracy will be conducted along the Chapter 5.

The measures described in the next sections use the entries values in the confusion matrix to compensate the disproportion between classes. The measures Precision, Recall, and F-Measure are adequate when the positive class is the main concern. The measures G-Mean, ROC, and AUC are appropriated when the performance of both classes is important.

2.3.2 Precision

The Precision is a measure of exactitude, and it denotes the percent of hits related to all positive objects. When analyzing together the equation (16) and the confusion matrix (Figure 8), it is possible to see the ratio between the true positives and the sum of the column with positive predictions. It is also possible to notice that this measure is sensitive to class distribution, as the divisor is a sum of positive and negative instances.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

2.3.3 *Recall*

The Recall, also denominated as Sensitivity, is a completeness measure, and it denotes the percent of positive objects identified by the classifier. Analyzing the equation (17) and the confusion matrix (Figure 8) together, it is possible to notice a ratio between the true positives and the sum of the elements in the line “positive class”. Because Recall just computes positive instances in its formula, this measure is not sensitive to class distribution.

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{17}$$

2.3.4 F-Measure

The F-Measure (18), also known as F-Score, F_1-Score, or simply F_1, synthetizes the information from the last two measures, Precision, and Recall, obtaining in this way a harmonic mean between them, were β is a coefficient that adjusts the relative importance of Precision versus Recall, normally for $\boldsymbol{\beta = 1}$ (Van Rijsbergen, 1979).

A harmonic mean tends strongly towards the smallest elements of a population, having an inclination (if compared to the arithmetic mean) to mitigate the impact of large outliers and aggravate the impact of small ones. In terms of classification results, it can be observed that the F-Measure shows lower results, when compared with other measures, denoting that F-Measure tends to be a pessimistic measure.

$$F - Measure = (1 + \beta^2).\frac{Precision.Recall}{(\beta^2.Precision) + Recall} \tag{18}$$

As mentioned, the F-Measure, like the Precision and Recall, assumes one class as positive. By default, to compute these measures for both (positive and negative) or more classes, most of the machine learning tools use an average weighted by the number of instances for each class. This can be used for class imbalanced problems to compensate for the disproportion of instances, but it can result in an F-Measure that is not between Precision and Recall, and in fact, during the experiments conducted in this work, the weighted average approach presented too optimistic results that were not representing the actual classifier performance in terms of F-Measure for all classes.

Throughout the experiments in all this work, the pessimistic behavior of F-Measure showed to be useful to adjust the user parameters passed to the algorithms (also known as hyperparameters), along the modeling process. To properly represent the F-Measure for more than one class, the arithmetic mean was used instead of the weighted average.

2.3.5 G-Mean

The G-Mean (Barandela, et al., 2003) explores the performance in both classes, considering the distribution between them, by computing the geometric average between the Sensitivity (17) and Specificity (20), obtaining in this way a balance of true predictions in both classes, or zero, if one of the classes has no correct prediction (19).

$$G - Mean = \sqrt{Sensitivity\ .Specificity} \tag{19}$$

2.3.6 ROC Curve

The Receiver Operating Characteristics (ROC) chart, also denominated ROC Curve, is applied in detection and signal analysis since the Second World War, and recently in data mining and classification. It consists of a two-dimension chart, where the y-axis refers to Sensitivity or Recall (17), and the x-axis calculated as 1-Specificity (20), as depicted in Figure 9.

$$Specificity = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \tag{20}$$

According to (Fawcett, 2004), there are several points in this chart that deserve attention. In the Figure 9, the point (0,0) means none of the positive instances were classified; (1,1), means no negative instances were classified; and (0,1), also indicated by letter D, means the perfect classification. One point is better than another, if its position is more to northwest.

The closer a point is to the x-axis, the more conservative the classifiers behavior is, that is, it will only make predictions if there are strong evidences, which can lead to few true positives. On the other hand, points in the upper right side denote a classifier with a liberal and/or aggressive behavior, which can lead to a higher level of false negatives. For example, in Figure 9, the point A is more conservative than the point B.

The secondary diagonal from Figure 9, where *y=x* denotes the classifier has a random behavior. A point at (0.5, 0.5) shows that the classifier hits 50% of positive instances and 50% of negatives, the remaining 50% were classified in a random way. The point C in the Figure 9 indicates the classifier tried to guess the positive class 70% of the time.

At last, the point E, which is in the lower triangle, indicates a classifier with performance lower than aleatory.

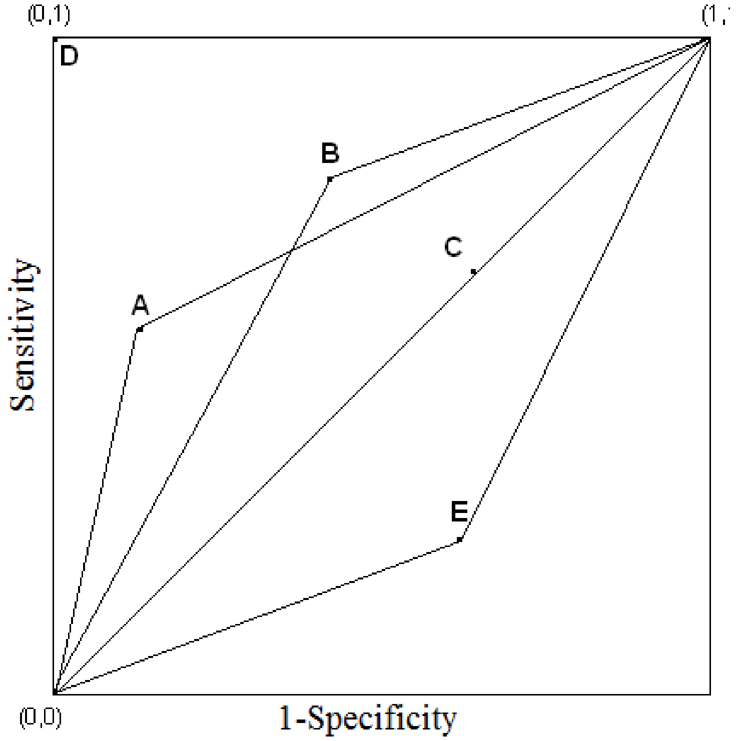


Figure 9 - The ROC chart and the interest points to be analyzed.

2.3.7 AUC

This work considers both classes, positive and negative, with equal importance, therefore, the Area Under Curve (AUC) measure is more appropriate for these cases, because it is insensitive to class imbalance problems (Fawcett, 2004). The AUC synthetizes as a simple scalar the information represented by a ROC chart (21).

$$AUC = \Phi\left(\frac{\delta}{\sqrt{\phi_{pos} + \phi_{neg}}}\right) \tag{21}$$

Where Φ is the normal cumulative distribution, δ is the Euclidean distance between the class centroids of two classes, and ϕ_{pos} , and ϕ_{neg} are the standard deviation from the positive and negative classes. An algorithm to calculate AUC is provided in (Fawcett, 2004).

2.4 – Text Mining

Text mining is considered a set of methodologies to extract useful information from text content. For this purpose, it is necessary to transform unstructured text content into a structured format readable by other algorithms. Text mining is derived from data mining research started during the ‘80s. It is considered a multidisciplinary field that involves information retrieval, natural language processing (NLP), data mining, statistics, and linguistics.

The main activities of text mining are: entity extraction, taxonomy extraction, sentiment analysis, document summarization, text categorization, text clustering, entity relationship, and visualization. Most part of these activities relies on data mining algorithms, but these algorithms are not able to deal directly with unstructured data, as they need a structured format, normally in a matrix shape (Weiss, et al., 2010). A text mining system normally has the architecture depicted in Figure 10, to be explained in the following sections.

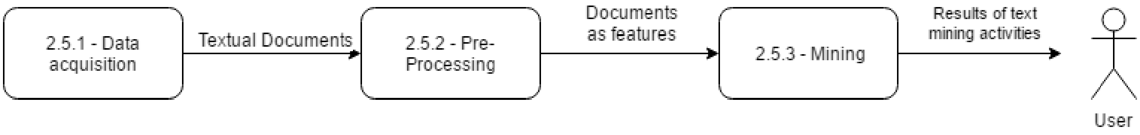


Figure 10 – Text mining system architecture.

2.4.1 - Data Acquisition

The set of textual documents, also known as corpus, can be collected from internet resources using a web crawler mechanism (Dhaka, et al., 2013), or another automated mechanism to collect unstructured data from email and messaging systems, databases, or textual files existing in a file system (e.g., log files, digitized books, speech to text, etc.). The selection of good and reliable sources of textual content is fundamental to obtain a successful text mining system.

2.4.2 - Pre-Processing

To transform unstructured data in features, the textual documents must be parsed into simple words, with the blank spaces and punctuation used to distinguish and separate the words. This process is also known as tokenization. A list with all existing words and the respective number of occurrences in the corpus can also be generated during this phase. After this, the words or terms are selected to form features. In this context, a feature can be understood as a value, and the feature name is the meaning of this value. Features can represent a word, a sequence of words or *n*-grams, which consists in a series of consecutive *n* words (Sidorova, et al., 2014), types of entities (e.g., company names, stock symbols), quantitative values (e.g., stock prices, date, time), syntactical structures like noun-phrases and part-of-speech, etc.

Not all the words carry information in the textual content. The stop words are terms with low importance for information retrieval (normally prepositions), and its removal is recommended. Terms with occurrence per document lower or above a specified threshold are also recommended for removal, because a few number of words have no representation, and do not carry significant information in the document. The same applies to repeated and abundant words. The min/max thresholds must be adjusted according to the problem under study, but normally values lower than ~5%, or greater than ~90% are reported in the literature.

The use of stemming reduces the number of words, by replacing a word to its base or stem (Lovins, 1968), (Porter, 1980), e.g., fruit = fructify, fruity, fruitful. The use of stemming requires caution and must be adjusted according to the problem under study, as it may remove important information existing in the original words.

The most common type of feature representation is the Bag of Words (BOW), first mentioned by (Harris, 1954) and still a predominant technique nowadays (Miner, et al., 2014), (Zhai & Massung, 2016). A BOW is basically a matrix, where each document is represented as a vector row, and the features (normally words) as the columns of this matrix. The columns of this matrix must contain not only the existing terms in the document, but also all the existing terms in the corpus. Not all the documents share the same terms, then the missing terms in a document are filled with zero or null, which can result in a sparse matrix, as demonstrated in Figure 32.

The feature values can be represented as categorical, binary (i.e., existence, nonexistence of a feature in a document), and numerical values. The numerical values can contain any integer or continuous value extracted from the textual content (e.g., prices, counting, etc.), or some measurement or weighting regarding that feature. For example, the Term Occurrence (TO), is the number of times a term occurs in a document, Term Frequency (TF) is the TO divided by the total number of terms in the document (22), since every document has a different length, it is possible that a term would appear many more times in long documents than shorter ones, then the division is a way of normalization.

$$TF(t,d) = \frac{number\ of\ occurrences\ of\ t\ in\ d}{number\ of\ terms\ in\ d} \tag{22}$$

Where:

- t is the term;
- d *is* the document.

When using TF, all terms have the same importance, however, to account for the fact that some words appear more frequently than others in all documents, the TF is inversely weighted by the frequency of the same word along the corpus (23), also known as the Term Frequency-Inverse Document Frequency (TF-IDF) (Robertson, 2004). Nowadays, TF-IDF continues being the most common approach for feature representation in text mining (Miner, et al., 2014), (Zhai & Massung, 2016).

$$TFIDF(t,d,D) \ = \frac{TF(t,d)}{log\,\frac{|D|}{n_t}} \tag{23}$$

Where:

- D is the corpus that contains the document *d;*
- $|D|$ is the number of documents existing in the corpus;
- n_t is the number of documents where the term *t* appears.

Meta-data information like source, author, document name, document type, date and time of creation/publication, time zone, and geographical origin, can also be added to the feature set.

As the result of the pre-processing phase, the corpus, and its documents are now represented by the extracted features.

2.4.3 - Mining

Once the data existing in the textual documents are readable in terms of features and values, they can be processed by a sort of algorithms. Documents can be grouped and associated using unsupervised learning (document clustering, association rules) to identify, visualize, and understand communities, concepts, taxonomies, and sentiments. Entities, groups of documents, taxonomies, sentiments, concepts, and meta-data can be used to assign a category (also known as label) to each document, to be used in supervised learning (document classification and regression) and recommendation systems (Weiss, et al., 2010), (Miner, et al., 2014), (Zhai & Massung, 2016).

The architecture and the most common techniques for text mining were presented in this section, and they will be referred frequently in this work. Nevertheless, text mining is an extensive and evolving area, and one section is not enough to describe all this branch of research. Other text mining techniques will be presented together with the bibliographic review in Chapter 3. The methodology of this work will be presented in Chapter 4, and new text mining techniques will be proposed.

Chapter 3 – Text Mining for Financial Market Prediction

This chapter presents the theoretical background in financial economics that supports the forecast of price movements in this branch of research, as well as the literature review about the efforts to use text mining to predict movements in the financial market.

3.1 Financial Economics Background

To predict changes in a market economy is a powerful ability, capable to create wealth and avoid losses. This kind of activity is based in some financial concepts that started to be developed centuries ago, but it had a strong development is the last six decades, with the advances in statistical techniques and computing applied to finance. Some of these concepts provide the theoretical background for this current work, while others are contradictory. These concepts can be categorized as asset valuation theories and financial behavior theories (Thomson, 2007), and a debate beyond the scope of this work is still unfinished. The correct understanding of all these concepts is essential to understand the research problem, and propose substantial solutions.

3.1.1 - Efficient Market Hypothesis

In the ‘50s, the use of probability theory and statistics to model asset prices started to be actively applied by financial economists. These developments led to the invention of Capital Asset Price Model (CAPM) (Treynor, 1961), (Sharpe, 1964), (Lintner, 1965), (Mossin, 1966). Initially as a rejection to CAPM and other statistical approaches at that time, the Efficient Market Hypothesis (EMH) (Fama, 1965), (Fama, 1965b) argues that the stock prices movements are a function of rational expectations based on publicly known information from companies, and these expectations are almost immediately reflected in the stock prices, and in the price history for instance. This implies that there is no justification for modeling stock prices changes using the price history, when these changes are already accommodated in the stock prices. The EMH claims these price changes cannot be explained only by the price history, and the external factors responsible by the price changes were identified as aleatory and not possible to predict, which assigns a random walk behavior to stock prices in EMH, theory also supported by (Malkiel, 1973), (Samuelson, 1972), and others.

In a review work (Fama, 1970), the author stated that there are three types of market efficiency: weak-form, semi-strong-form, and strong-form efficiency. The weak-form is considered a soft EMH, and it admites the price movements are determined entirely by information not contained in the price series, and it does not require that the prices remain in equilibrium all the time. The semi-strong form implies that the stock prices have a very quick and unbiased adjustment to public available new information. In the strong-form the share prices reflect all public and private information imediatelly, no one can earn excess returns, and it is considered a hypotetical scenario, because having access to private information means to ignore the current undisclosure laws. Despite all this time, the EMH continues to be an active theory under discussion, and it is supported by empirical and theoretical research (Read, et al., 2013).

3.1.2 - Behavioral Economics

In an answer to EMH, the behavioral economic (BE) theories (Camerer & Loewenstein, 2004) argue that the markets are not efficient, and the random walk element in fact can be explained by the human behaviour, as ultimately, they are responsible to take decisions within the economical agents, and as humans they commit irrational and systematic errors. These errors affect the prices and returns, and create market inefficiencies for instance. The behavioural economics theories are supported by studies in psychology, sociology, finance, and economy, and they analyse the psychological, social, cognitive, and emotional aspects of human behaviour when taking decisions, and their respective consequences on economy, financial markets, prices, and returns. It was observed that the same information can have different interpretations, as the market participants have cognitive biases, which are organized into four categories: biases that arise from too much information, not enough meaning, the need to act quickly, and limitations of memory (Haselton, et al., 2005). In order to reconcile EMH and BE, the Adaptive Market Hypothesis (AMH) (Lo, 2005) claims that the traditional models can coexist with behavioural models, and it implies that the degree of market efficiency is related to environmental factors such as the number of competitors in the market, the magnitude of profit opportunities available, and the adaptability of the market participants.

The recent findings in behavioural economic principles state that market conditions are products of human behaviour involved, (Tomer, 2007), (Jurevičienė, et al., 2013), (Hollingworth, et al., 2016). The recent speculative economic bubbles were used to refute EMH, and it was claimed that the bubbles and irrational exuberance are proofs of market inefficiency, and they can be explained by behavioural economics. Nevertheless, this discussion is still vibrant and ongoing, and it is beyond the scope of this work, but it seems the EMH and BE theories will continue to be opposite forces in the evolution of financial economics studies.

For those who believe the markets are predictable, the efforts in this area can be organized as: technical analysis, fundamental analysis, and technological approaches.

3.1.3 - Technical Analysis

The technical analysis (TA) relies on specific tools and visual patterns in a market graph and other indicators to mainly examine the supply and demand, to forecast the price movements and returns. TA is a widespread technique among the market brokers and other participants to support an investment decision. The methodology for TA varies greatly, but in general the past market data (normally price and volume) is used for study and backtesting, and the analysis of daily market data represented as visual chart elements, like head and shoulders, double top/reversal, is used to identify patterns like lines of support, resistance, and channels (Elder, 1993). The use of market indicators and moving average techniques are a common approach, but a range of tools, econometrics, and proprietary methods are also reported.

Despite the wide application in the industry, and part of financial practice for decades, TA ultimately relies on human interpretation, and due to its subjective nature, frequently technicians can make opposite predictions for the same data, which can be explained by BE theory. TA is commonly a target of controversies when submitted to scientific assertion, with some studies supporting it (Aronson, 2007), (Balsara, et al., 2007), (Irwin & Park, 2007), while others pointing problems such as low predictive power

(Griffioen, 2003), (Browning, 2007), (Yu, et al., 2013). TA is also object of discussion from EMH supporters, but according to (Lo, et al., 2000), TA can be an effective way to extract information from market prices.

3.1.4 - Fundamental Analysis

Fundamental Analysis (FA) is a technique to identify the underlying value of financial instruments, and it concentrates in examining the economic health and productive capacity of a financial entity as opposed to analyse only its price movements. FA started to be used as a trading mechanism in 1928, and the first book about it was published in 1934, now in its 6th edition (Graham, et al., 2008).

To perform this valuation, FA looks for financial economic indicators, also known as “the fundaments”. When applied to stocks, FA looks for company’s health by mainly examining business statements like assets, liabilities, earnings, as well as the company’s market, competitors, management, announcement of discoveries and new products or failures. In the case of Future Markets and ForEx, it looks for macro-economic announcements and the overall state of economy, in terms of interest rates, taxes, employment, GDP, housing, wholesale/retail sales, production, manufacturing, politics, weather, etc.

The predictive and profit capacity of FA relies on the events of mispriced financial instruments, for example, buy shares of stocks when a company is under valuated to its fundaments, and then sell the shares, when the market detects the inaccuracy and the prices are adjusted to a higher value, or when the company’s share prices become over valuated for its fundaments. FA tends to be related to long-term investment strategies, as companies and governments take time to change their fundaments. Another FA strategy is the “buy and hold”, where the fundaments allow to find good companies to invest, with lower risk, to keep the assets growing and earning dividends with the business development, rather than focus in immediate profit.

Fundamental analysts must understand quantitative (numeric terms) and qualitative information (non-measurable characteristics like quality, sentiments, opinions, etc.), and the public announcements are a crucial moment to operate. In an era of information, the use of automated tools for FA has become a mandatory practice for analysts, and these tools normally have features like stock scanners, alerts, data feed, strategy backtesting and order placement integration, but in conclusion, it continues to be a challenging and manual activity relying on human analysis, liable to errors and uncertainty (Kaltwasser, 2010), as described by BE theory. In this scenario, the automatic understanding of textual content seems to be an attractive alternative, but among the reviewed works in the section 3.2, only one is devoted to FA (Tetlock, et al., 2008).

3.1.5 - Technological Approaches

The advent of computing brought the financial markets to a next level in multiple aspects, and since the beginning the related literature is populated with examples of technological approaches applied to financial market prediction, and this wide range of possibilities is always bringing new advances, and so far, there is no clear taxonomy about these approaches. As an example, recently with the refinement of internet mechanisms, (Preis, et al., 2013) used trading strategies based on the search volume of 98 financial terms, provided by Google Trends[1], and demonstrated an accumulated return of 326% in eight years of backtesting simulation, and (Moat, et al., 2013) demonstrated the number of views of specific financial articles in Wikipedia are associated with stock market movements.

The advent of computing also made possible to use inferential statistics and artificial intelligence in large scale. These advances culminated with the creation of data mining as a subfield of computer science in the ‘90s. Despite the wide range of possibilities granted by technology, this section will concentrate on how data mining and automated trading systems are applied to financial market prediction.

A typical trader can only analyze, take decisions, and monitor a limited number of strategies simultaneously. In this scenario, the cognitive biases contribute for a human failure (Haselton, et al., 2005). The most pervasive problem with trading (which also includes TA and FA) is to overcome the emotions. As a branch of data mining, machine

[1] https://www.google.com/trends/

learning algorithms are capable to take automated decisions given a training dataset as input, and it could be an alternative to mitigate or solve that problem.

The use of data mining as a decision algorithm can be combined with any kind of automation in financial market, but in this industry, there is a vagueness about the meaning of terms such as automated trading system, algorithm trading, quantitative analysis, quantitative trading, and high frequency trading. These terms share some common characteristics, and to avoid confusion, a clear definition is required before advancing:

- Automated Trading System (ATS), also known as robot traders, is a generic term for computer programs that automatically take decisions, create negotiation orders, submit, and monitor the order execution in an exchange or other types of trading platforms. The terms quantitative trading, algorithm trading and high frequency trading are considered an ATS.

- Algorithm Trading has the intent to execute large orders and avoid costs, risks, and reduce market impacts, and it is extensively used by pension funds, hedge funds, and investment banks. For example, in the case of portfolio change in a pension fund, a huge number of shares from several stocks must be sold, and this capital must be reinvested in another stock. It normally uses time, price, and volume as input to calculate how to split the order and automatically submit the small orders over time (Kissell, 2013). Despite to be desirable, the main purpose of algorithm trading is not to make a profitable trade, but this term became commonly associated with any kind of automated trading strategy, especially the ones where the main purpose is to make profit. For these cases, according to (Johnson, 2010), the term quantitative trading sounds more appropriated.

- High Frequency Trading (HFT) is an automated trading system that can submit negotiation orders in a high velocity rate to exchanges or other types of trading platforms. HFT relies on Direct Market Access (DMA) or Sponsored Access with high speed connections and extremely low latency infrastructure (Johnson, 2010), (Aldridge, 2013) to deliver a negotiation order in milliseconds or microseconds. Recently a hardware vendor claimed that it took 85 nanoseconds for the entire messaging job to deliver an order to exchange (Sprothen, 2016). There is no clear definition of HFT in terms of order frequency, but it mainly depends on the trading strategy particularities, as trading opportunities can last from milliseconds to few hours. Nowadays, roughly 55% of trading volume in U.S. stock markets and 40% of European stock markets volumes are executed with HFT (Gerig, 2015), and about 80% of foreign exchange futures volumes are HFT (Miller & Shorter, 2016). HFT can be used by any type of ATS.

- Quantitative Analysis aims to understand the market and valuate financial instruments to predict behaviours and events using financial economics techniques, mathematical measurements, statistics, predictive modelling, and computing (Merton, 1973), (Hardie, et al., 2008).

- Quantitative Trading are automated trading strategies based on quantitative analysis. It is also known as black box trading, because some systems make use of proprietary and undisclosed algorithms.

The use of data mining with structured data for financial market prediction is a widespread technique, but still an evolving branch of research (Trippi & Turban, 1996), (Thawornwong & Enke, 2004), (Shadbolt & Taylor, 2013), (Halls-Moore, 2015). The use of unstructured data as input for data mining, also known as text mining, has an immense potential to contribute with BE and financial market prediction, in terms of automatic extraction of concepts, entities, patterns, trends, and sentiments from textual content. The first initiative with TMFP appeared in (Wuthrich, et al., 1998), and as the object of research in this current work, the efforts and findings in this branch of research will be detailed in the next section.

Once the financial economics theories and concepts are reviewed, it is possible to say that the research problem of this branch of research is to predict the effect of textual information on the economy and respective asset prices and returns. It is also possible to say that TMFP can be considered a quantitative trading approach, and in the case of intraday prediction, a HFT quantitative trading approach. The next section presents a survey about the efforts to solve the research problem.

3.2 Related Works

In this section, a bibliographic review about the developments and state of the art of Text Mining Applied to Financial Market Prediction (TMFP) will be conducted. The criteria used to select a research regarding this subject are: it must have some text mining or NLP methodology; it must predict economical events or changes in some financial instrument; and publications with number of items (i.e., news articles) lower than 200 were not included, as they do not carry conclusive results. An extensive survey about this branch of research was conducted by (Nassirtoussi, et al., 2014), bringing expressive contributions and insights about TMFP. The bibliographic review in this current work aims to use important aspects from that survey. It also includes some missing publications and bar charts for quick understanding and identification of trends. The aspect of sentiment analysis received more attention, and new works that came up after 2014 were added.

Table 17 from Appendix A contains 36 reviewed works, and depicts the evolution of TMFP methodology since the first reported effort in this branch of knowledge, until the main aspects of this current work at the bottom. Table 17 is chronologically ordered by year of publication, and cells marked with “-” correspond to information not mentioned in the reviewed work. The respective results from these researches will be compared and discussed along the section 5.4. The meaning of acronyms and financial terms can be found in Table 19, Appendix A.

One of the first researches published about TMFP is (Wuthrich, et al., 1998). The authors developed a prototype to predict the trend of one day of five major stock indexes (DJIA, Nikkei, FTSE, HSE, STI). The forecast was based on daily news published overnight in portals, like for example, the Financial Times, Reuters, and the Wall Street Journal. The documents were labeled according to a model of three categories: up, steady, and down. A dictionary with 423 features was defined manually by experts. The Bayesian, Nearest Neighbor, and a Neural Network classifiers were trained, and categorized overnight all newly published articles.

These predictions were used for investment simulation, with 7.5% of cumulative return after three months, what can be considered a good result, if compared with the return of 5.1% from DJIA index in the same period. Nevertheless, as mentioned before, the nature of markets is complex, and they are extremely difficult to forecast using any methodology. Unfortunately, the authors also reported an average Accuracy of 43.6% over the five indexes, which does not guarantee these investment results can be reproducible in different contexts. These low results announced the challenges in the coming years, and even text mining and other ways of prediction in financial market continue to be an open problem, but since then, the design of TMFP systems follow a structure like Figure 11.

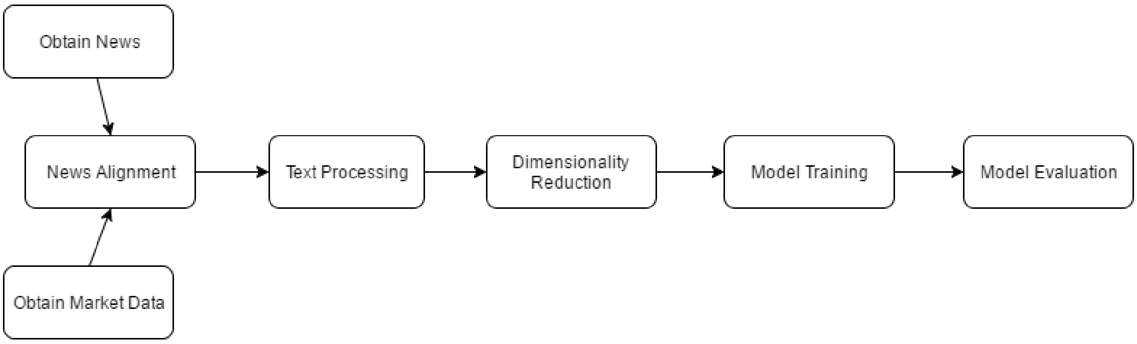


Figure 11 - General design of a TMFP process.

In the next sections, each methodological aspect represented as a column from Table 17, and its respective research efforts will be discussed and compared. The approaches in this current work will be also cited and compared when applicable, but the complete methodology will be presented in Chapter 4.

3.2.1 – Year of Publication

Figure 12 represents all the reviewed works, grouped by year of publication. Despite all this time, there is no expressive number of publications about TMFP, if compared with other branches of research. The number of publications has risen before the 2008 crisis, then another increase in 2012, with a peak of six articles, most of them motivated by a sudden interest in sentiment analysis applied to behavioral finance. Since then it is observed a decreasing number of publications.

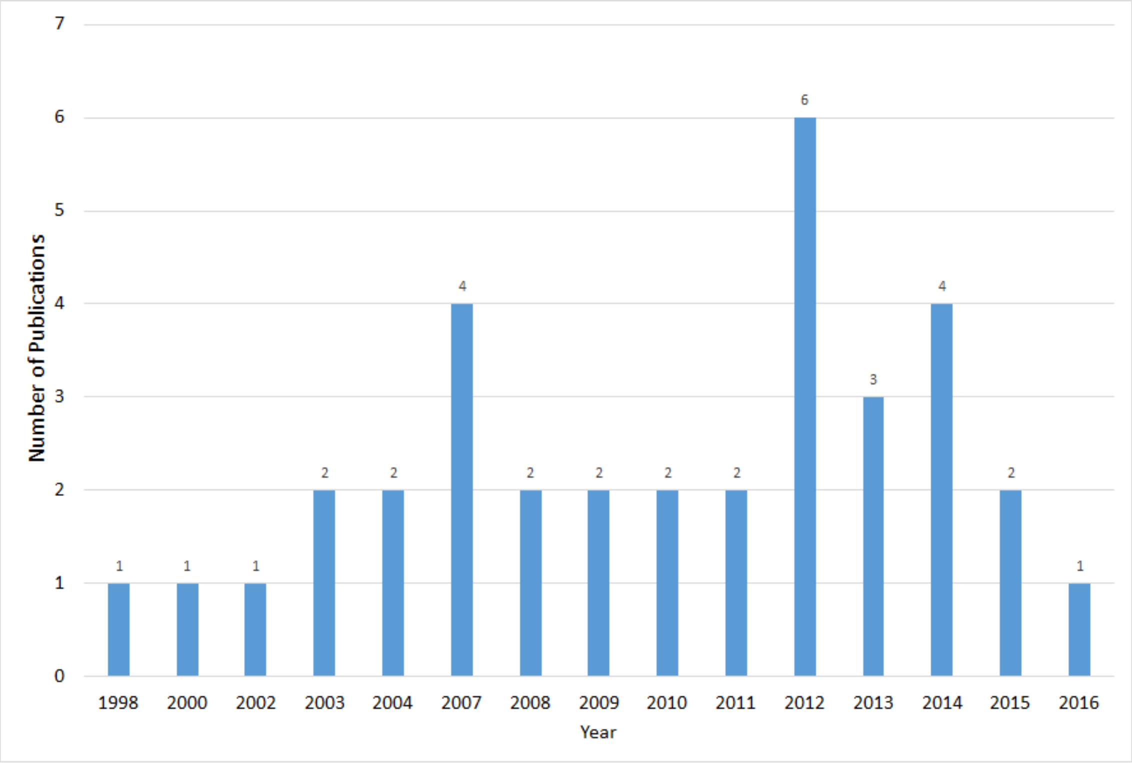


Figure 12 - Number of publications related to TMFP grouped by year.

The remaining sections will explain the methodological aspects in this branch of research, and point problems that could explain this decreasing number of publications in recent years.

3.2.2 - Source of News

The first thing to do in a TMFP process is to gather news articles. To achieve this, several types of web crawler mechanisms were used to obtain news content, and then some source of news is necessary to feed up these mechanisms. Since the beginning, the digital version of the main communication vehicles for financial markets were used as a source of news: Bloomberg, Down Jones, German Society for Ad Hoc Publicity (DGAP), Financial Times (FT), Forbes, Reuters, Wall Street Journal (WSJ), etc. Most of these sources provide news feeding services embedded with sentiment attributes, and (Crone & Koeppel, 2014) used 14 built-in sentiment indicators from Reuters MarketPysch aiming to anticipate ForEx movements. The specialized news aggregators like Yahoo! Finance and Google Finance were also applied as source of specialized news, and they were also used in this current work. When the exchange is outside of American and European markets, using local news also demonstrated to be more appropriated in several works. Recently, social media contents like blogs, forums, and Twitter started to be used (Yu, et al., 2013), while others focused solely on Twitter (Bollen & Huina, 2011), (Vu, et al., 2012), (Makrehchi, et al., 2013).

3.2.3 - Number of Items

Among the reviewed works, the most common numbers of items, i.e., news articles, to be processed, are between 10k and 1M (Figure 1). According to Table 17, most of these numbers are associated with the period of time and the source of news, with volumes ranging from 216 (Zhai, et al., 2007) from Australian Financial Review to 30M of items (Makrehchi, et al., 2013) using Twitter as data source. The current work uses 128k news articles.

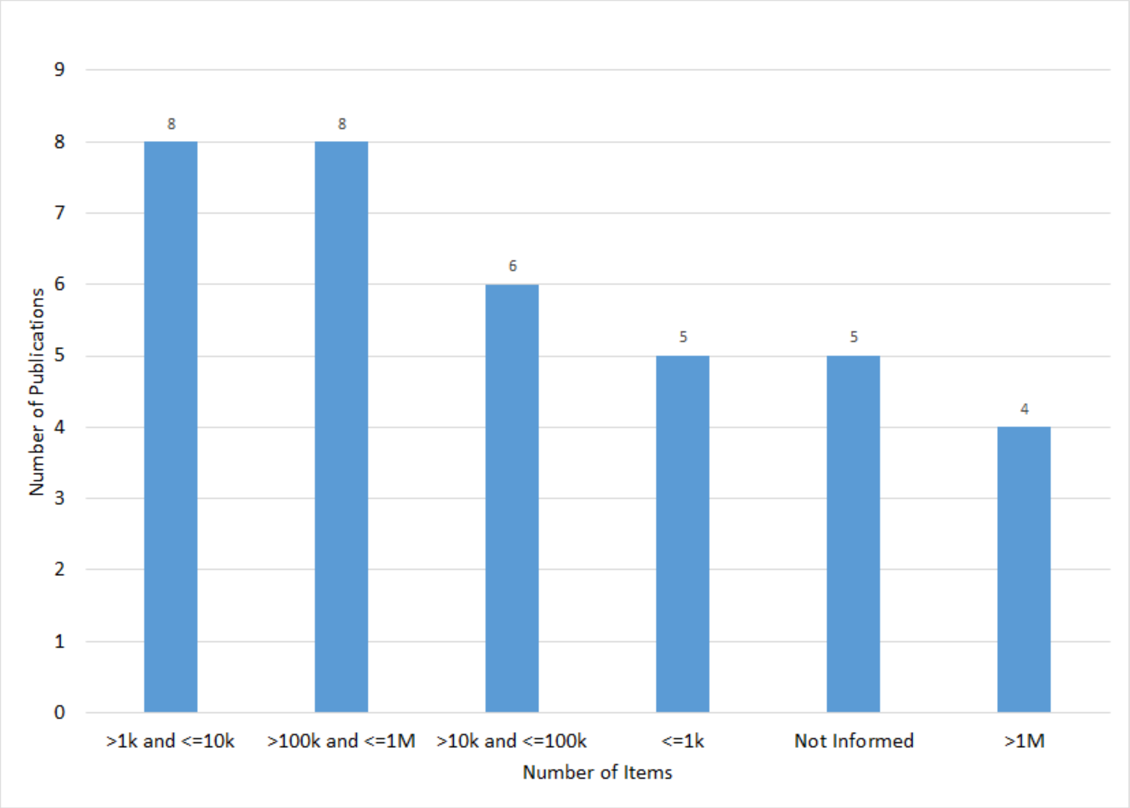


Figure 13 – Number of publications grouped by the number of items (news articles) collected.

Due to scalability and timing constraints, in some cases the number of items could justify the use of big data frameworks like the Hadoop Environment (White, 2009), but among the reviewed works there was no mention about this methodology.

3.2.4 - Market / Index / Exchange

In terms of market, the great majority of the reviewed works are devoted to predict the movements of stocks and foreign exchange (ForEx), and (Groth & Muntermann, 2011) used news articles to predict the risk when investing in stock markets.

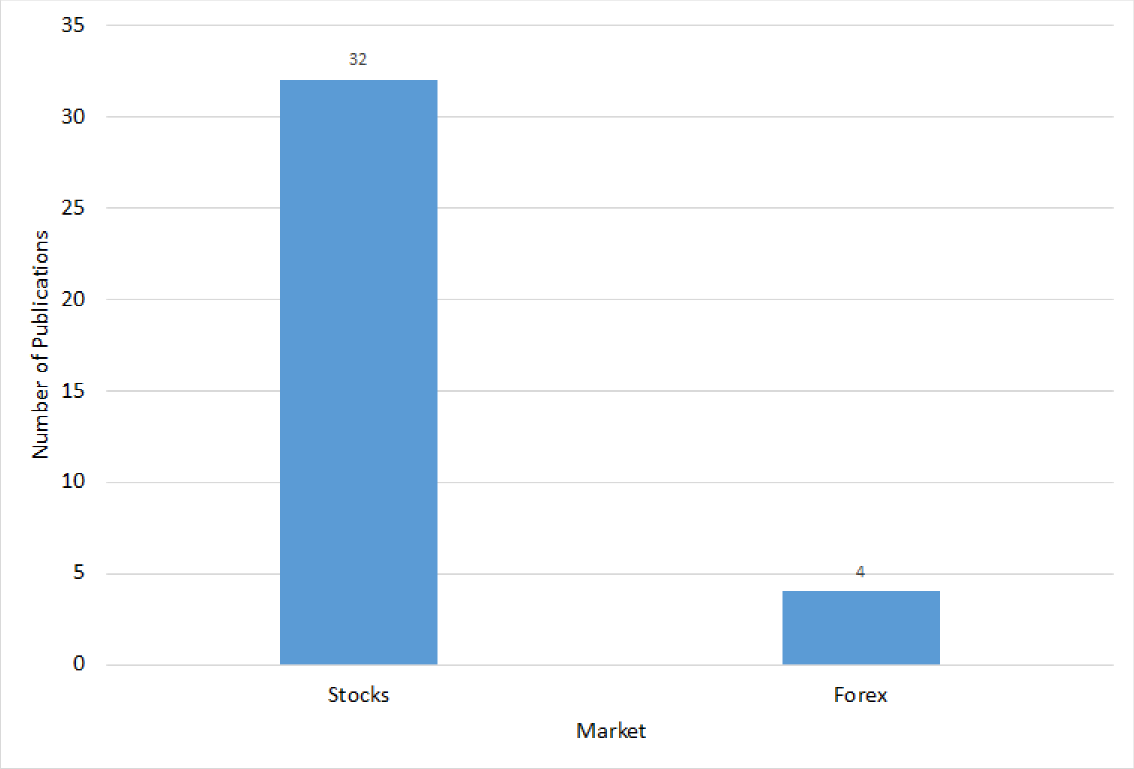


Figure 14 - Number of publications grouped by market.

The most studied indexes are DJIA and S&P 500 (10 papers), followed by the local indexes according to authors' country. These studies focus on predicting the price movements from stocks that compose the index, and (Makrehchi, et al., 2013) focused to forecast the whole S&P 500 index movements. The same happens for exchanges, with more studies focused on NYSE and NASDAQ, and other exchanges according to the author's country.

3.2.5 - Time-Frame / Alignment offset

Time-Frame means the periodicity of the predictions. Most of the reviewed works aims at predicting the market movements on a daily basis (Figure 15). (Butler & Kešelj, 2009) and (Li, 2010) made it on a yearly basis; and (Vakeel & Shubhamoy, 2014) conducted a study to predict the effect of news on the stocks before and after the elections in India.

The studies with intraday time-frame aim to predict the market movements within the trading hours, and the alignment offset represents the period between the news article is published and the asset price is affected. The most common values are between 15 and 20 minutes, but predictions with larger periods of one and three hours were also studied. This current work has the lowest alignment offset, with periods of 1, 2, 3, and 5 minutes, and it relies on the technological capacity of trading in a very short period with HFT (Johnson, 2010). Recently, an analysis about the effects of macroeconomic news on the intraday ForEx prices, with an offset of 5 minutes, was conducted by (Chatrath, et al., 2014).

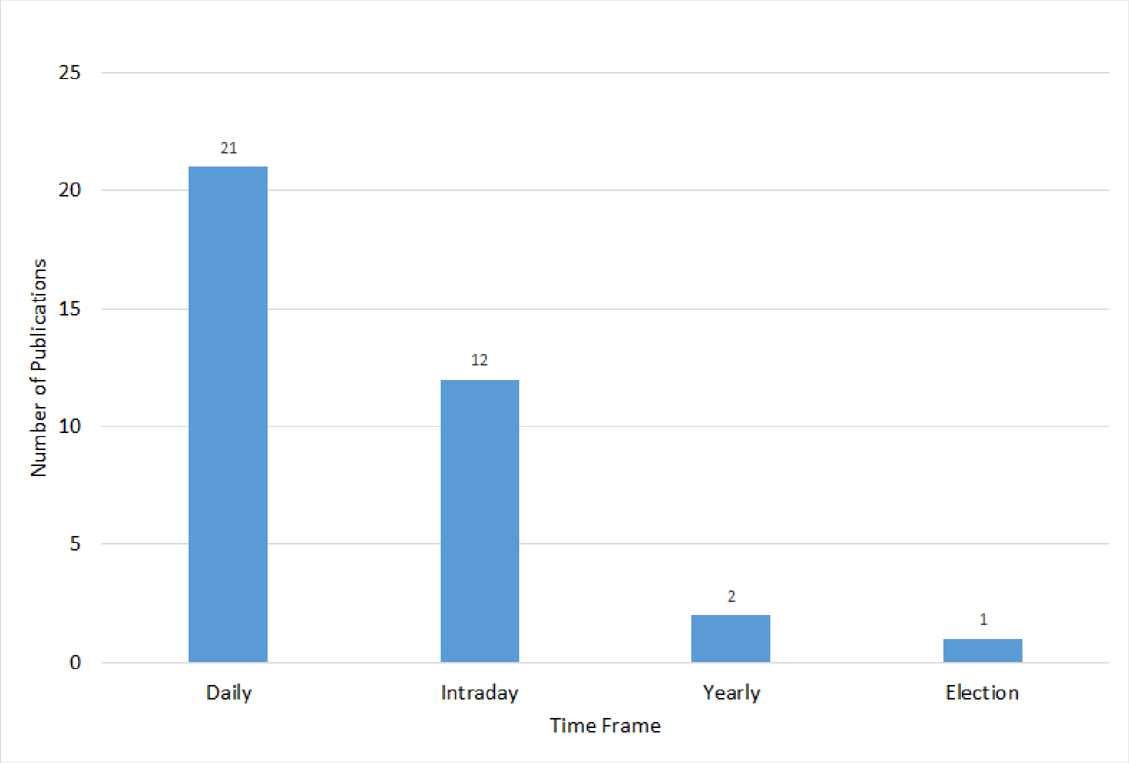


Figure 15 - Number of publications grouped by the time frame.

3.2.6 – Period of News Collection / Number of Months

In terms of the number of months collecting news articles, there is a discrepancy among the reviewed works, as can be seen in Figure 16, with most of the researches with less than six months or more than 24 months of news articles gathering.

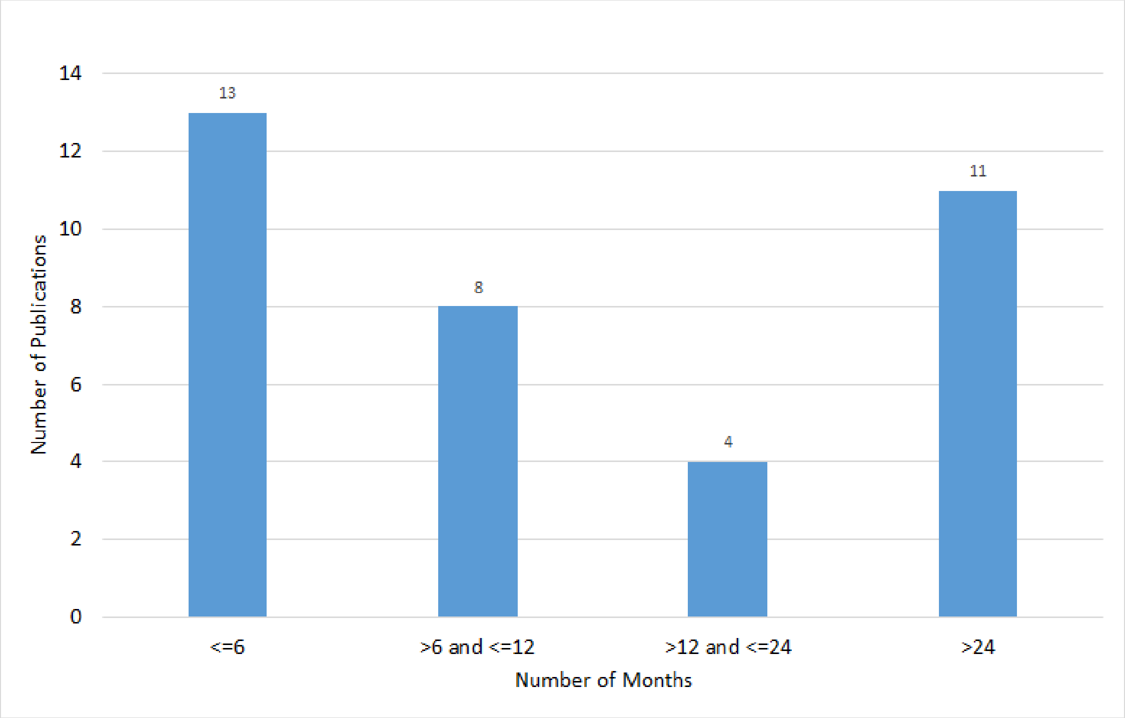


Figure 16 - Number of publications grouped by the number of months collecting news articles.

Another observed aspect is the gap in years, after the news articles are collected and the results are published. According to Figure 17, in most part of the cases this gap is above three years. This approach is very common in finance, and it is known as backtesting, as some problems need more time and focus to be studied, due to the nature of financial markets.

(Vu, et al., 2012) was the only research to report an online test (i.e., the predictions were made with live news). This test was conducted between 8/Sep/2012 and 26/Sep/2012.

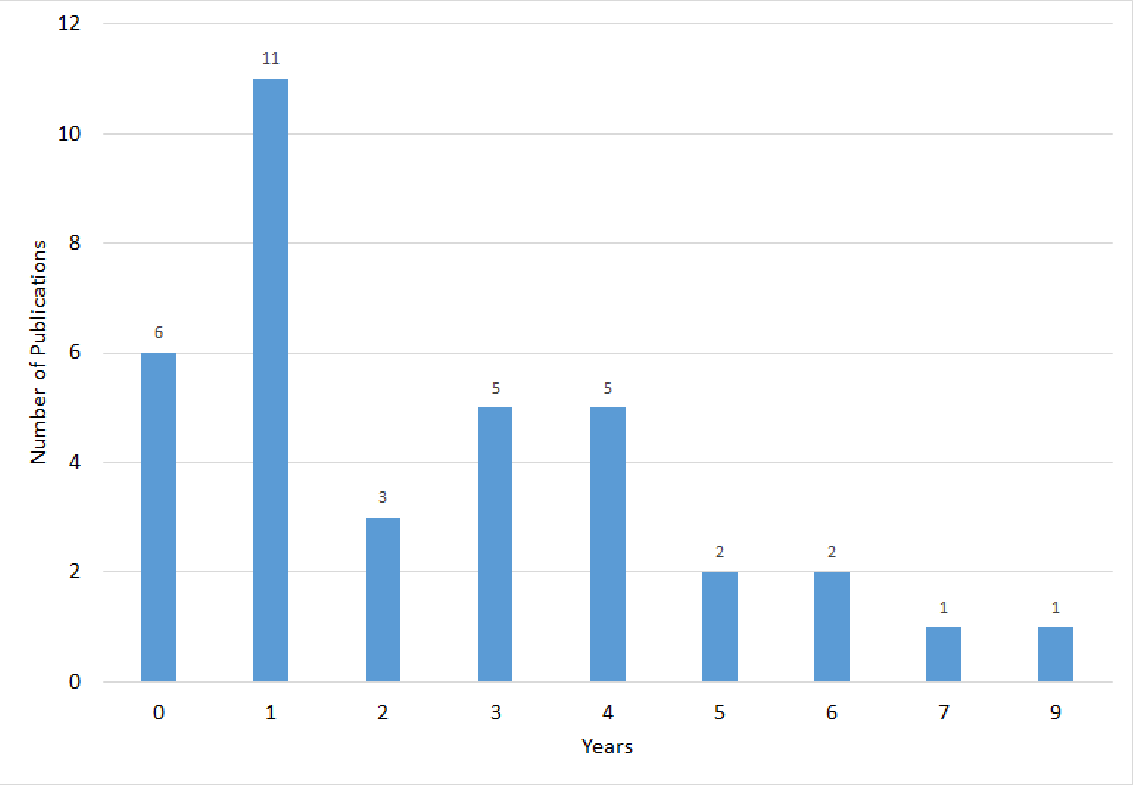


Figure 17 – Number of publications grouped by the number of years between the data collected and publication.

3.2.7 - Number of Classes / Target prediction

Most of the reviewed works focus on classification, and according to Figure 18, the majority of the publications use two classes to be predicted, denoting the prices will rise or fall. The studies with three classes aim to predict if the prices will rise, fall or will be stable, and studies with four or five classes represent a finer gradient of three classes. In a different approach, this current work focuses in the best moment to buy, and uses two classes that identify the rise or stability/fall of prices.

Few reviewed works are devoted to regression (Bollen & Huina, 2011), (Schumaker, et al., 2012), (Jin, et al., 2013), and in an effort to conciliate fundamental analysis with text mining, (Tetlock, et al., 2008) used linear regression to predict companies' earnings, and they found out that negative words have more predictive power; and more recently, (Yang, et al., 2015) used regression with abnormal sentiment scores.

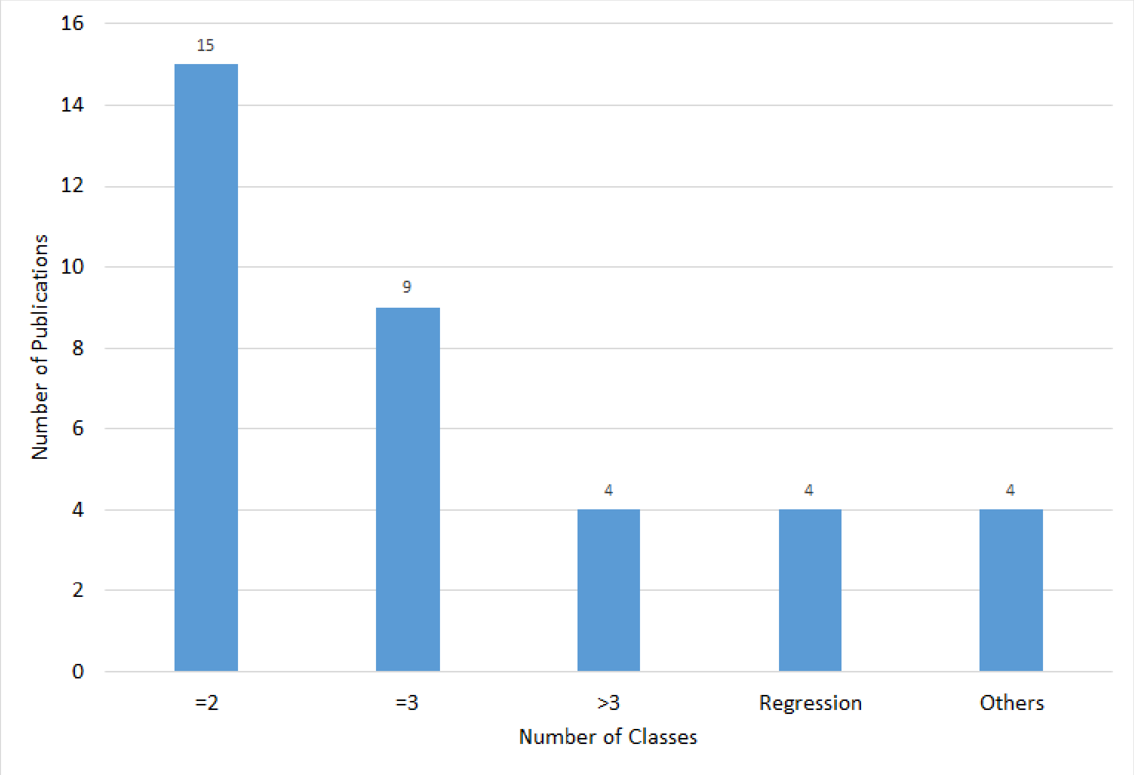


Figure 18 - Number of publications grouped by number of classes / target prediction.

As other approaches, (Das & Chen, 2007) was one of the first works that used sentiment indicators, and applied an aggregate sentiment index aiming at predicting the stock price changes. (Schumaker & Chen, 2009) used a derivation of SVM to make discrete stock price prediction, and in order to assist the investors, (Huang, et al., 2010) used association rules to obtain a significant degree assignment of each newly arrived news.

3.2.8 - Feature Selection / Representation

In text mining, the way the text content will be represented is crucial. An incorrect text representation in terms of features can lead to information loss and a meaningless outcome.

Nowadays, most research with text mining relies on bag of word (BOW) (Harris, 1954), and this is also reflected among the reviewed works, as about 2/3 of them use BOW (Figure 19). In terms of feature representation, 1/4 of the reviewed works use TF-

IDF (23); other 1/4 use term frequency (TF), TF-CDF, or binary representation; and most of these feature representations occur together with BOW.

TF-CDF consists in TF divided by CDF. However, the CDF calculation requires to know the class of the new article in advance. For this reason, it is not possible to use such technique in real predictive scenarios, because the class is completely unknown. In spite of that, two reviewed works used TF-CDF as part of their studies (Peramunetilleke & Wong, 2002), (De Faria, et al., 2012).

In order to reduce the dimensionality and maintain the word ordering (i.e., syntax) at a certain level, the combination of BOW and n-grams (Sidorova, et al., 2014) was applied by (Das & Chen, 2007), (Butler & Kešelj, 2009), (Hagenau, et al., 2012), (Vakeel & Shubhamoy, 2014), and this current work.

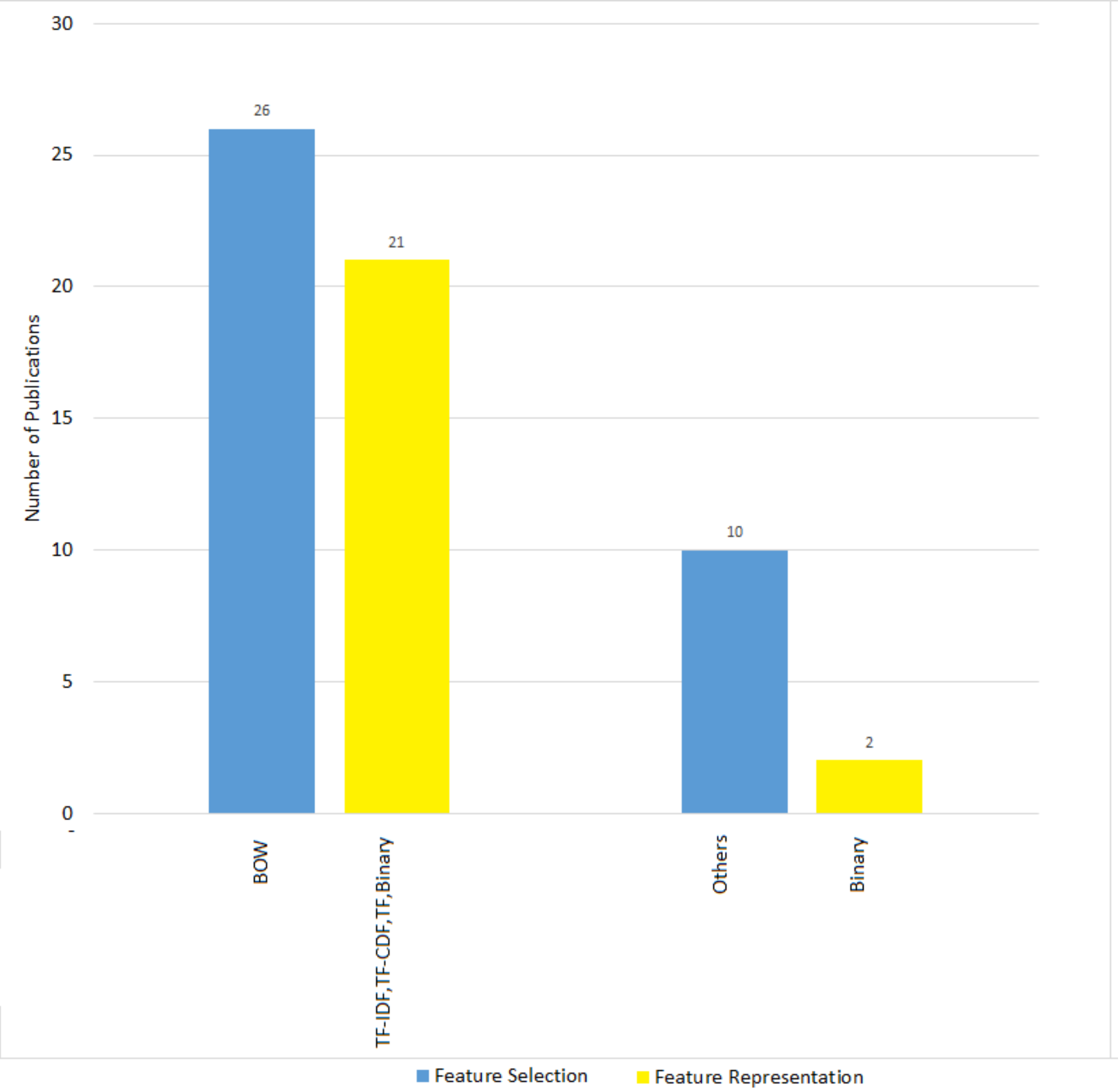


Figure 19 – Number of publications grouped by the feature selection, and the application of TF-IDF, TF-CDF, TF or Binary representation.

In terms of representing text content as sentiment and opinion features, (Bollen & Huina, 2011), (Schumaker, et al., 2012) used Opinion Finder (Wilson & Hoffmann, 2005); while (Tetlock, et al., 2008; Nassirtoussi, et al., 2015) combined BOW with positive/negative words representation; and (Li, 2010), (Lugmayr & Gossen, 2012), (Nassirtoussi, et al., 2015), (Yang, et al., 2015) combined BOW with some type of sentiment measurement.

Among the reviewed works outside the "bag of words realm", the feature selection relies on Latent Dirichlet Allocation (LDA) (Mahajan, et al., 2008), (Jin, et al., 2013); visual coordinates (Soni, et al., 2007); simultaneous terms and ordered pairs (Huang, et al., 2010). (Wong, et al., 2014) applied a Sparse Matrix Factorization + ADMM methodology that encapsulates the feature selection, dimensionality reduction and machine learning phases in a single algorithm. (Yu, et al., 2013) used a daily number of positive/negative emotions, and bullish/bearish anchor words, and (Crone & Koeppel, 2014) used 14 built-in sentiment indicators from Reuters. Recently (Fehrer & Feuerriegel, 2016) used a deep learning method (Bengio, 2009) called recursive autoencoders (Liou, et al., 2014), which combines numerical vectors, n-grams, dimensionality reduction (the auto encoder optimization) and logistic regression classifier in one algorithm.

3.2.9 - Dimensionality Reduction

The text mining processing generates a large number of features, but for classification purposes, most of them do not carry any meaning or association with the underlying label (Donoho, 2000), and must be removed.

According to Figure 20, the majority of reviewed works use some type of statistical measurement such as Language Models, Information Gain, Chi-Square, Minimum Occurrence Per Document to define the most valuable features given a threshold (Forman, 2003), and normally this is combined with BOW, stemming, and stop words removal.

Another common approach is the use of pre-defined dictionaries, where the non-existing words will be removed. Most of the dictionaries were created by specialists and have some association with the related companies/market/exchange. However, (Makrehchi, et al., 2013) used a mood list, (Tetlock, et al., 2008) used the Harvard-IV-4 to define positive and negative sentiments, and (Kim, et al., 2014) created an automated sentiment dictionary algorithm.

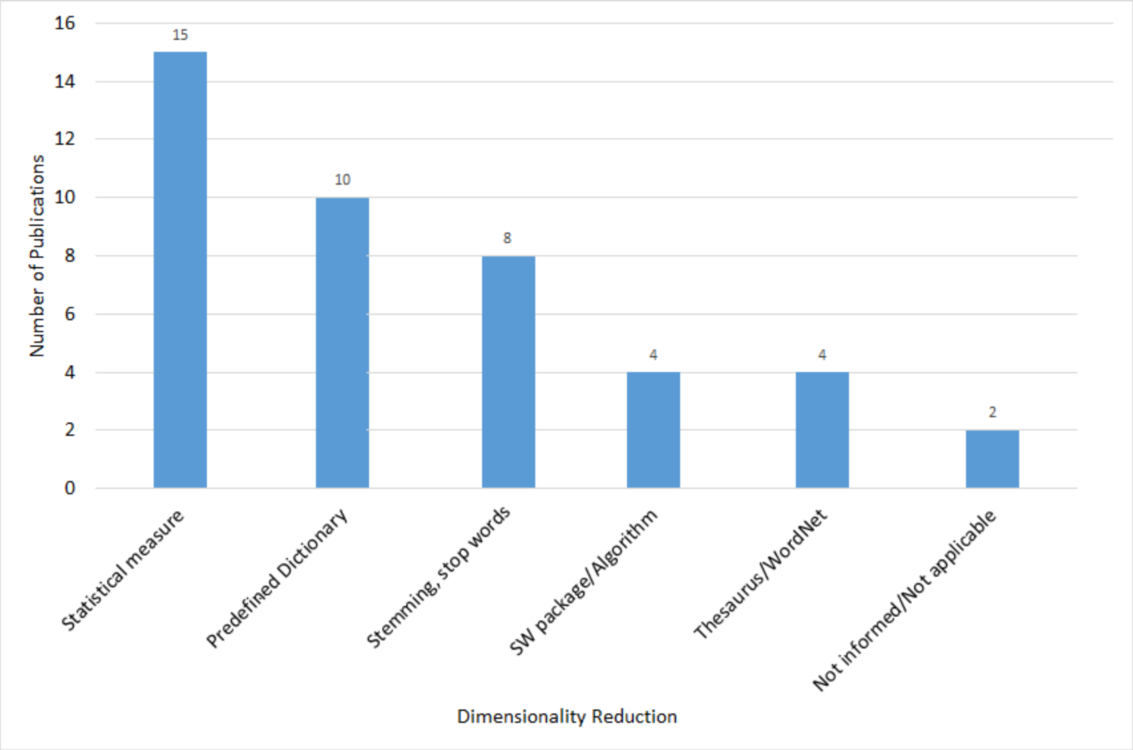


Figure 20 - Number of publications grouped by the dimensionality reduction method.

The synonym and hypernym replacement using some type of Thesaurus or WordNet (Miller, 1995) is also a promising approach for dimensionality reduction, but it was only explored by (Soni, et al., 2007), (Zhai, et al., 2007), (Huang, et al., 2010), and (Nassirtoussi, et al., 2015).

3.2.10 - Learning Algorithm

Once the text content is transformed into features, the choice of a learning algorithm is also important. According to Figure 21, the most common machine learning algorithm applied to TMFP is the Support Vector Machine (SVM) (Cortes & Vapnik, 1995), followed by the Naïve Bayes (Rish, 2001), or a combination of these two or more algorithms like k-NN (Fix & Hodges, 1951), Decision Trees (Kohavi & Quinlan, 2002), and others (Wu, et al., 2007) in the same study. Among the reviewed works using Artificial Neural Networks, most of them applied the classical implementation of feed forward algorithm, but (Bollen & Huina, 2011) applied Self-Organizing Fuzzy Neural Network (SOFNN) (Leng, et al., 2005), as a combinatorial optimization approach, and (Fehrer & Feuerriegel, 2016) used a new deep learning approach called Recursive

Autoencoders (Liou, et al., 2014), (Bengio, 2009). Among the other learning methodologies, (Wong, et al., 2014) applied a Sparse Matrix Factorization + ADMM, but it was reported an Accuracy close to a random classifier.

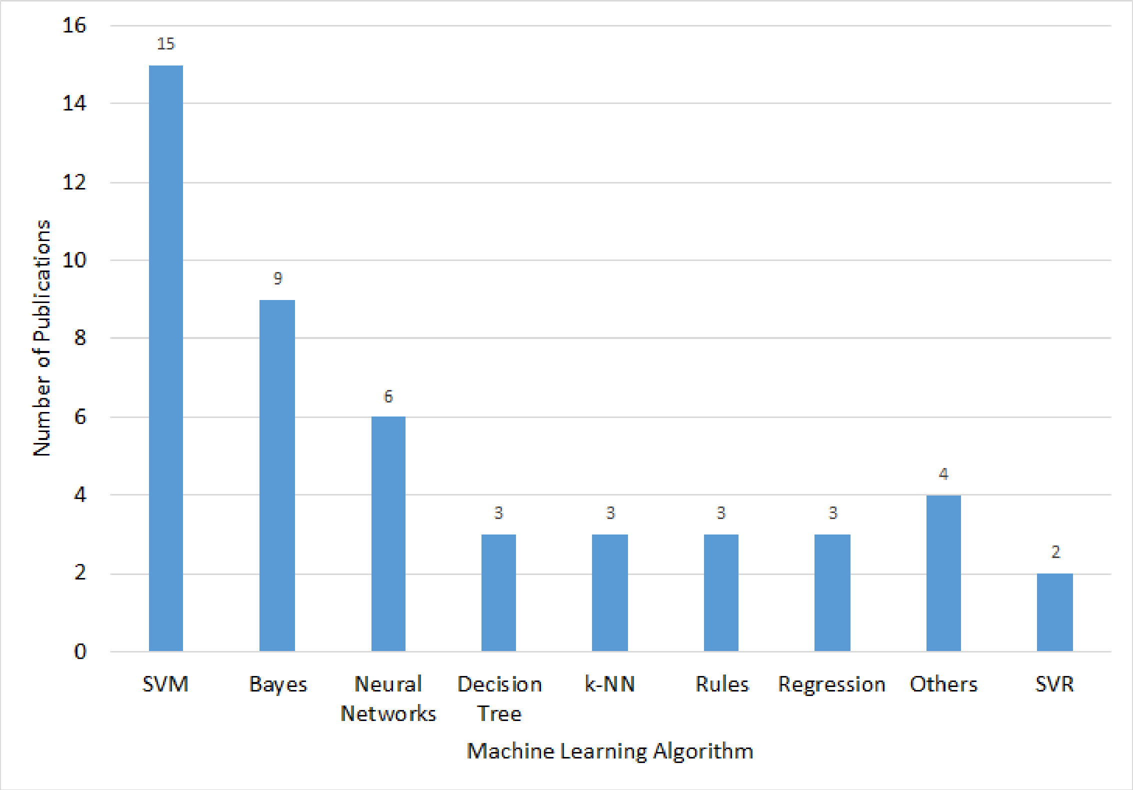


Figure 21 - Number of publications grouped by the machine learning algorithm.

In terms of regression, most of the reviewed works used the Linear Regression (Tetlock, et al., 2008), (Jin, et al., 2013), (Chatrath, et al., 2014), while (Schumaker, et al., 2012) applied Support Vector Regression (SVR) for discrete numeric prediction instead of classification, and (Hagenau, et al., 2012) used SVM together with SVR as a second algorithm to predict the discrete value of stock returns.

3.2.11 - Training vs. Testing

Normally in predictive machine learning, for model evaluation purposes the data is split in one set for training, and another set for testing, and for time series problems like TMFP, the order of data must be kept (Hastie, et al., 2003). It is a common sense that, in most of the cases, the lower the ratio between the size of test set and training set, the better the quality of predictions in the test set, and among the reviewed works, (Nassirtoussi, et al., 2015) conducted an experiment asserting this. Figure 22 depicts the number of publications, grouped by the ratios between test and training (calculated as *test size / training size . 100*), and also the number of publications that used cross validation or did not provide any information about this subject.

The way the dataset is split for model evaluation is crucial for predictive analytics, but according to Figure 22, unfortunately almost 1/3 of the reviewed works did not provide this information, and specifically for (Yang, et al., 2015), the results of their works are not applicable for prediction, because the model evaluation was applied in the training set. Another concern is the use of cross validation in 14% of the reviewed works, as this procedure disrupts the natural order of a time series, giving to classifier model an unfair advantage, as information from the future is used to predict events in the past, making the resulting models unreliable for prediction. A complete discussion about the use of cross validation for TMFP will be conducted in section 5.4.

The data splitting between 20% and 50% is the second most frequent group, and ratios between this range are a common practice in machine learning. Recently (Wong, et al., 2014) used an additional validation set to evaluate the model, before it is applied to the test set, which is becoming a common practice with the advent of data mining competitions at Kaggle [2].

[2] http://www.kaggle.com

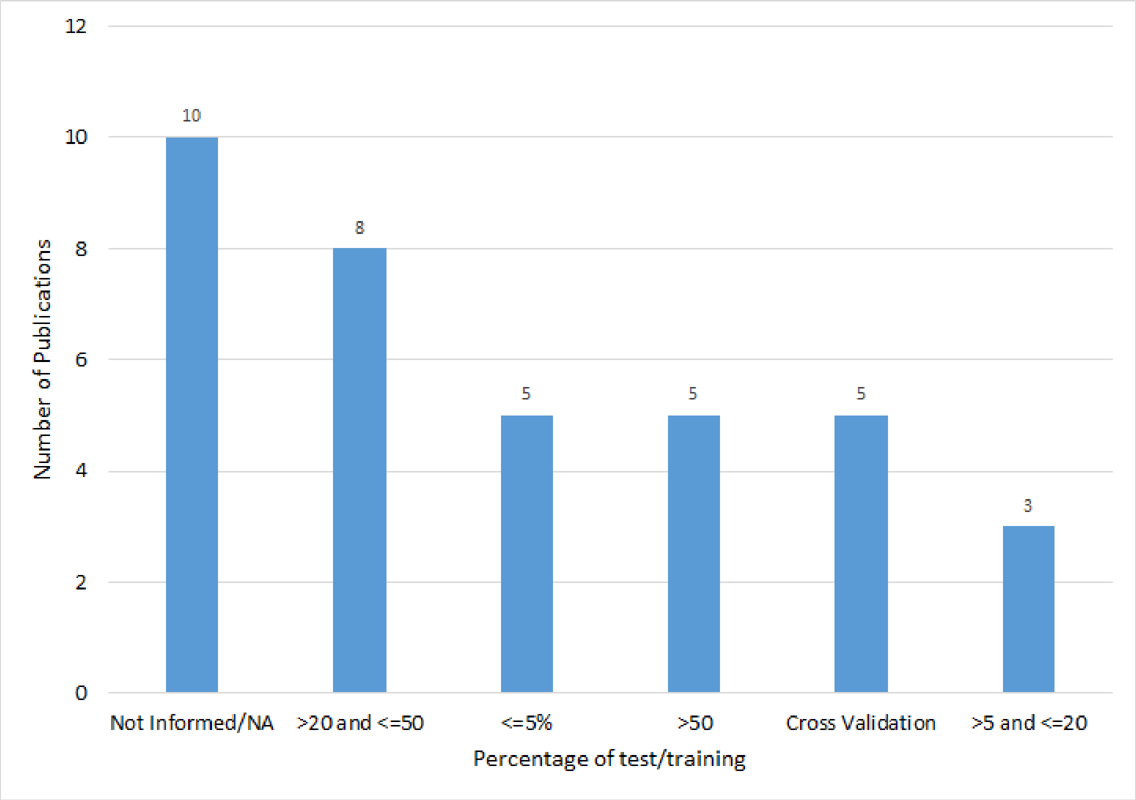


Figure 22 - Number of publications grouped by the percentage of test size/training size, cross validation and not informed.

In order to maximize the performance of predictive models, there is a group of reviewed works (this current work included) with splitting rations lower than 5%. On the other hand, there is another group with proportions of training and test above 50% and values of 71%, 80%, 100% and 1650% (Mittermayer, 2004).

3.2.12 - Sliding Window

As mentioned in the previous section, in most cases, the lower the ratio between the size of test set and training set, the better the quality of predictions in the test set. In the case of time series, one of the techniques to maximize the size and quality of the training set is known in the literature as sliding window (Dietterich, 2002).

The concept of sliding window can be explained by, a dataset is split in 10 blocks, where seven blocks are designated for training, and three blocks for testing. But instead of performing the test along the three blocks together, the evaluation is performed in the first test block, then after the first block from the training set is discarded and the first test block is added to the training test, the classifier is rebuilt, then the evaluation is performed to the second test block, and then the process repeats until all the test blocks are evaluated, as exemplified in Figure 31.

The experiments demonstrated in Chapter 5 show that the sliding window is a suitable approach for time series, as it adjusts the classifier model to the new reality occurring at the edge of the problem (the new market conditions), and it discards old concepts and events occurred at the beginning of the training set. Despite these advantages, only 22% of the reviewed works applied this technique: (Wuthrich, et al., 1998), (Peramunetilleke & Wong, 2002), (Tetlock, et al., 2008), (Butler & Kešelj, 2009), (Vu, et al., 2012), (Jin, et al., 2013), (Vakeel & Shubhamoy, 2014), (Nassirtoussi, et al., 2015), and this current work.

3.2.13 – Sentiment Analysis

The purpose of the Sentiment and Emotional analysis is to extract and measure the sentiments preserved in the text content, and it is widely used in product evaluation and consumer feedback (Pang, et al., 2002), (Cambria, et al., 2013).

According to Figure 23, among the reviewed works, the use of sentiment analysis for TMFP started in 2007, by (Das & Chen, 2007) in a work initiated before 2004, which applied an assembly of different classifiers to extract the investor sentiment from message boards, in order to predict a similar behaviour of stock price changes, presented as a chart visualization and a correlation table of sentiments and stock returns. Unfortunately, it was not possible to extract any classification measurements because the published confusion matrix was not square. In an attempt to predict companies' earnings in a long-term period, (Tetlock, et al., 2008) used the Harvard-IV-4 psychological dictionary to define positive and negative sentiments, and they have found that negative words have more predictive power for fundamental analysis. After a pause of three years (after the 2008 crisis), this technique just started to be applied again in 2011, achieving a peak in number of publications in 2012, motivated by a sudden interest in sentiment analysis applied to behavioral finance, and being a predominant subject in the reviewed works of 2013.

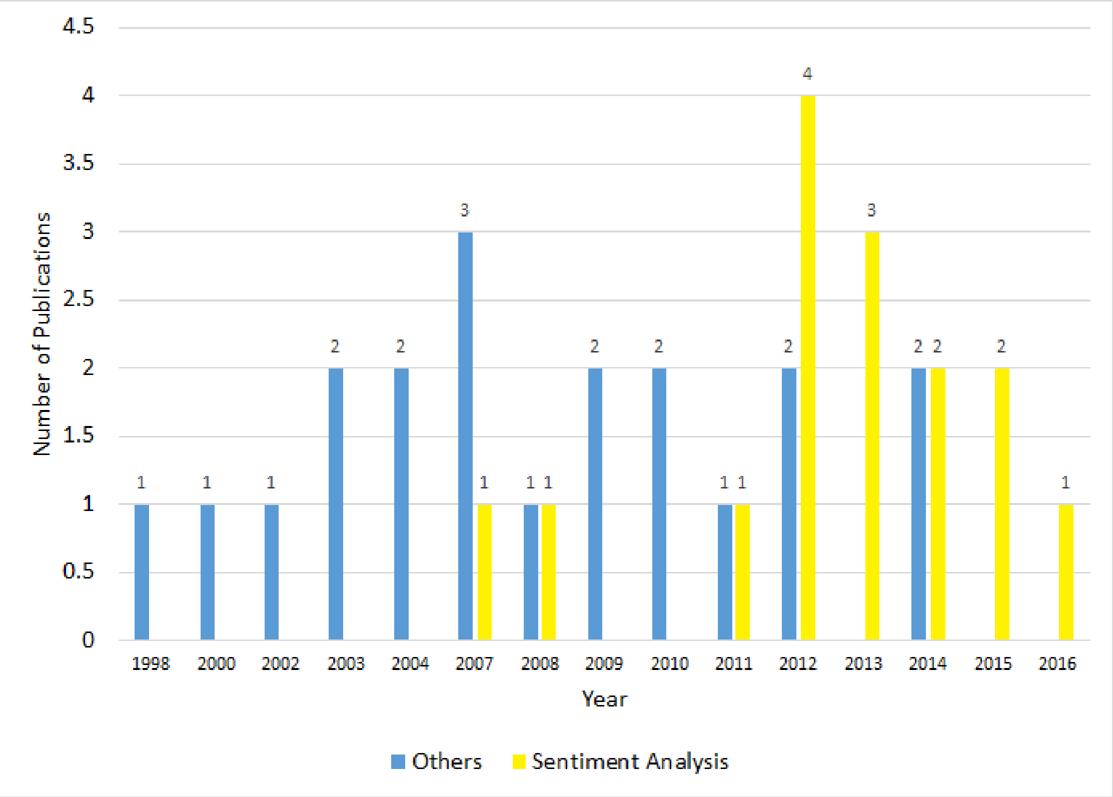


Figure 23 - Number of publications grouped by application of sentiment analysis.

Restarting from 2011, (Bollen & Huina, 2011), and (Schumaker, et al., 2012) used Opinion Finder (Wilson & Hoffmann, 2005) to measure the emotional polarity (positive and negative) of the sentences, in an effort to predict the price movements in the stock market. (Vu, et al., 2012) used Twitter Sentiment Tool (Go, et al., 2009) to determine the polarity of sentiments from Twitter posts, and define the level of confidence the consumers have in a company, to predict the daily outcome of NASDAQ stocks. For this task, a Part-of-speech (POS) tagger proposed by (Gimpel, et al., 2011) was applied to extract adjective, noun, adverb and verb words and associate them as anchor words to “bullish” and “bearish” sentiments.

.

In 2013, (Jin, et al., 2013) applied topic clustering methods and used customized sentiment dictionaries to uncover sentiment trends by analysing relevant sentences, and (Makrehchi, et al., 2013) used a pre-defined mood word list, obtaining a gain of 20% above the S&P 500 in three months of investment simulation. To analyse the effect of social and conventional media, their relative importance, and their interrelatedness on short term firm stock market performances, (Yu, et al., 2013) used a Naïve Bayes algorithm and the Cornell Movie Review dataset[3] to identify positive and negative words, to explore the document-level polarity and generate 20 sentiment scores for 862 companies. These sentiment scores were applied with an econometric coefficient, and compared with their respective market performances (e.g., return and risk). They found out that blog sentiment has a positive impact while forum sentiment has a negative impact on return, and both blog and Twitter sentiment have a positive effect on risk. Further, they found out the interaction effect between Twitter and news sentiment has a significant negative effect on returns, but not a significant effect on risk.

In 2014, (Kim, et al., 2014) developed an algorithm for automatic discovery of sentiments to form a dictionary. Nowadays, several vendors are specialized in providing news feeding services embedded with sentiment attributes, and (Crone & Koeppel, 2014) used 14 built-in sentiment indicators from Reuters MarketPysch to predict ForEx movements.

Recently, (Nassirtoussi, et al., 2015) developed a muti layer architecture, with one of the layers devoted to sentiment integration using SumScore Features and SentiWordNet dictionary, and (Fehrer & Feuerriegel, 2016) used a variant of recursive autoencoders (Liou, et al., 2014), which includes an additional layer in each autoencoder, to extract and predict sentiment values.

3.2.14 - Semantics

Semantics deals with the meaning of the words, and according to Table 17, more than half of the reviewed works used some semantics approach, but in general it was only applied to discover word relationships like synonyms and hypernyms, aiming at the word weighting and dimensionality reduction, by weighing or replacing related words using a thesaurus or WordNet (Miller, 1995).

[3] Polarity dataset v2.0 URL: http://www.cs.cornell.edu/people/pabo/movie-reviewdata/.

3.2.15 - Syntax

Syntax deals with the sequence and grouping of words, but according to Table 17, less than 1/3 of the reviewed works used some syntax methodology, and in general, the use of n-grams was a common practice. (Das & Chen, 2007) applied triplets, i.e., a sequence of an adjective or adverb, followed or preceded by two words; (Schumaker & Chen, 2009), (Hagenau, et al., 2012) applied noun-phrases, i.e., a phrase composed of a noun (substantive) and the modifiers (articles, possessive nouns, possessive pronouns, adjectives, and participles); and (Vu, et al., 2012) applied part-of-speech (POS) tagger to extract adjective, noun, adverb, and verb words and fixed them to “bullish” and “bearish” as anchor words.

3.2.16 – Data Balancing

Normally in financial market prediction, the opportunities to obtain profit are rare events, and this is reflected in the underlying data, with a majority number of examples representing no changes, and a minority of examples representing the best moment to buy or sell shares and other securities. This condition denotes an imbalanced dataset (Weiss & Provost, 2001), and it brings some problems for supervised learning. A detailed study about data balancing will be conducted in section 4.3, but one of its undesirable effects is the inappropriate use of classification measures, and especially in the case of Accuracy, it does not consider the number of examples distributed between the majority and minority classes.

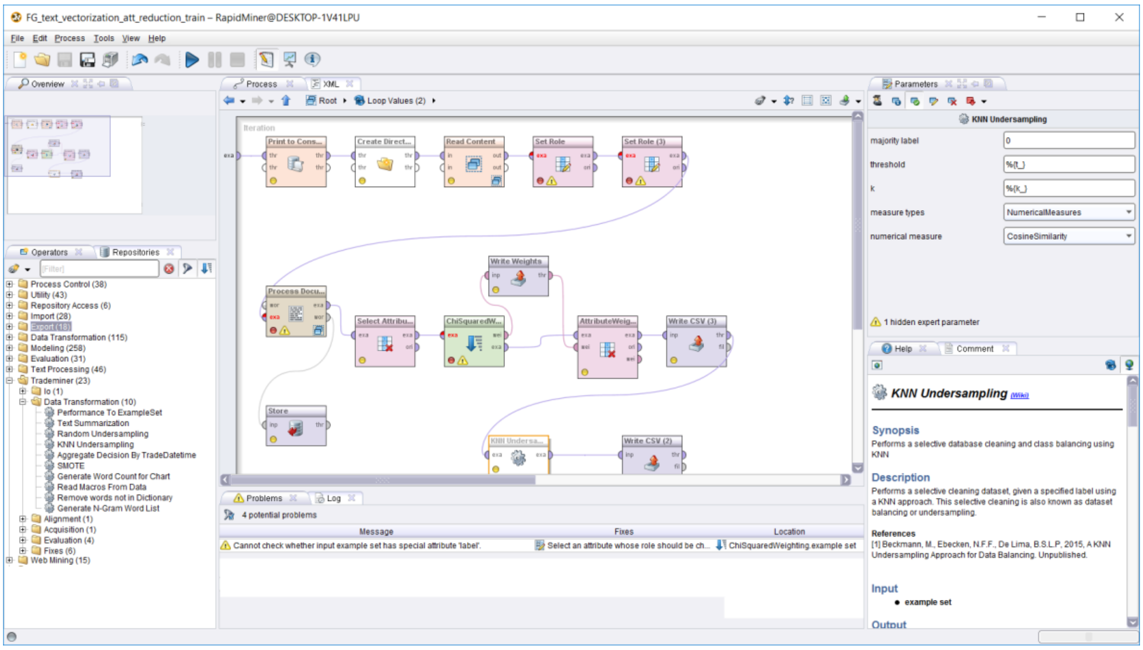
Despite the importance of data balancing for TMFP, among the reviewed works, only six studies paid attention to this subject (Peramunetilleke & Wong, 2002), (Mittermayer, 2004), (Soni, et al., 2007), (De Faria, et al., 2012), (Makrehchi, et al., 2013), and this current work. The remaining works barely cited the existence of this problem in their studies, but as depicted in Table 16, about 50% of the results were published only as Accuracy, which made most of these results questionable, and raises serious concerns in this branch of research, as it undermines the investor’s confidence.

A hypothesis to explain why there are few publications in this branch of research, and why this number continues decreasing in the last years, is the lack of confidence from investors on the application of TMFP in a real investment scenario, mostly caused by the problems identified above and in section 3.2.11, which for instance, is reflected in the academic interests and activities.

In order to contribute to revert this lack of confidence scenario, the next chapter will present the methodology proposed in this work.

Chapter 4 – Methodology

With the purpose of build models to predict the price changes with text mining, a long and automated process is necessary, from the collection of news and stock prices, the treatment of this textual data into a bag of words vector, training, test, and simulation. All the TMFP process was developed with the RapidMiner platform and its respective extensions (Mierswa, et al., 2006), and the innovation proposed in this work was developed in a new extension called TradeMiner (Figure 24), following the instructions in (Land, 2010).



KNN Undersampling

majority label: 0
threshold: %{t_}
k: %{k_}
measure types: NumericalMeasures
numerical measure: CosineSimilarity

1 hidden expert parameter

4 potential problems

Message	Fixes	Location
Cannot check whether input example set has special attribute 'label'.	Select an attribute whose role should be ch...	ChiSquaredWeighting.example set

KNN Undersampling (Wiki)

Synopsis

Performs a selective database cleaning and class balancing using KNN

Description

Performs a selective cleaning dataset, given a specified label using a KNN approach. This selective cleaning is also known as dataset balancing or undersampling.

References

[1] Beckmann, M., Ebecken, N.F.F., De Lima, B.S.L.P, 2015, A KNN Undersampling Approach for Data Balancing. Unpublished.

Input

- example set

Output

Figure 24 - Screenshot of RapidMiner desktop with a workflow using TradeMiner operators.

The main processing flow can be seen in Figure 25, and it was repeated for each company listed in the Dow Jones Index (DJIA) during the time the data was gathered, with each company owning a predictive model. As mentioned, only data mining and text mining techniques will be used, no econometric techniques will be applied during this process. Each sub-process from Figure 25 will be explained in the next sections.

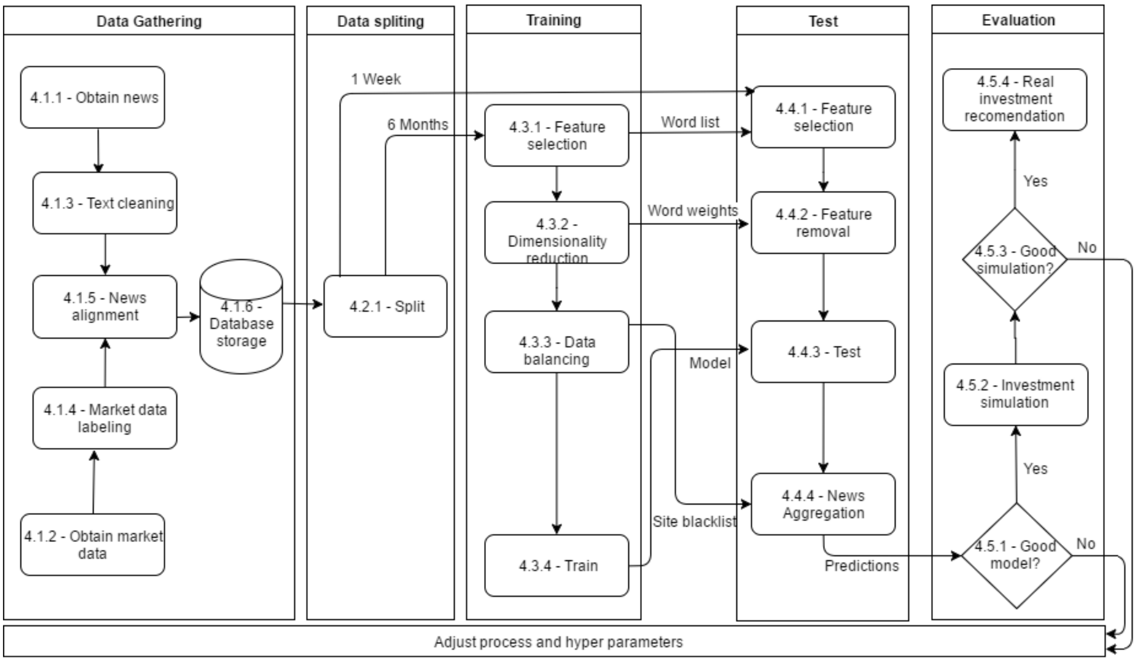


Figure 25 – The text mining modelling process applied to price change prediction in financial market.

4.1 – Data Gathering

The first step in the TMFP process is to obtain data. For this purpose, the news articles and stock prices will be collected from the internet to form the experiment's dataset, commonly known in finance as backtesting.

4.1.1 – Obtain news

The process starts with the gathering of news articles, also known as documents, from the internet. To make this possible, a web crawler (Dhaka, et al., 2013) was developed using the RapidMiner´s Web Mining extension. This web crawler acted as a client for Rich Site Summary (RSS) feeds, and it started under test from April/2012, and stayed in full operation from January/2013 until September/2013, collecting 128,195 news articles related to the 30 companies listed in DJIA, with the total size of three gigabytes of data.

The source of news came from Yahoo Finance[4], and Google Finance[5]. Both engines use an entity identifier algorithm similar to (Carpenter, 2007), which makes possible to retrieve news articles associated with the company´s stock symbol.

Each news article record is composed of the news content in English, the stock symbol, and the published date and time in the time zone where the stock is traded, in the case of DJIA, the Eastern Standard Time (EST). Later, the published date and time are converted from EST to UTC. For the news articles released when the markets are closed, the publication date and time to be considered are the exchange opening time in the next available trade date. For example: the news published Tuesday night will only be considered on Wednesday 14:30 UTC, news published during the weekend will only be considered on Monday 14:30 UTC. The news articles records are strictly stored and processed in their chronological order.

4.1.2 – Obtain market data

To obtain the stock prices (also known as market data) associated with the companies under study, a web service client was developed in Java language. The source of market data was supplied by a free web service[6], which provides minute by minute stock prices and other quantitative values from the companies negotiated at NYSE and NASDAQ exchanges. Some complementary market data was also retrieved from EODData[7]

This web service client also started under test from April/2012, and stayed in full operation from January/2013 until September/2013, collecting 1,929,522 market data records related to the 30 companies listed in Dow Jones Index (DJIA), with the total size of 700 megabytes of data.

Each market data record consists of the stock symbol, the close price from the day before, the open price, the last price traded at that minute, and the respective date and

[4] http://finance.yahoo.com
[5] http://finance.google.com
[6] http://www.restfulwebservices.net
[7] http://eoddata.com

time, in the Eastern Standard Time (EST), and the same date and time in UTC. The market data records are strictly stored and processed in their chronological order.

4.1.3 - Text Cleaning

This step removes the existing HTML tags in the textual content, and deletes news articles records where the remote text content is inexistent or decommissioned, i.e., the text content has messages related to page not found or broken links.

4.1.4 - Stock Price Labeling

Once the market data is available, it's necessary to identify the higher and the lower prices, and assign a label, also known as class value, to each record. In this work, the market data records were labeled as SURGE, and PLUNGE, and the rest of records were labeled as NOT RECOMMENDED. These labels are respectively identified as 2, -2, and 0 in the database. Table 1 summarizes these labels and their respective creation rules and usage.

Table 1 - Summary of labels used in this work.

Label	**Description**	**Label in Database**	**Investment recommendation**
SURGE	Prices with rise >= 75% of the maximum ascent observed during the day.	2	Buy
PLUNGE	Prices with fall <= 75% of the minimum descent observed during the day.	-2	Sell
NOT RECOMENDED	All other cases	0	Do not buy or sell (Hold position)

The labeling method uses slopes to measure the price changes (24), being v_{t+1} the current price, v_t the previous price, and $\max(\Delta)$ is the maximum ascent (if $\Delta \geq 0$) or minimum descent (if $\Delta < 0$) observed during the day.

$$\begin{aligned} \Delta &= v_{t+1} - v_t \\ slope &= \frac{\Delta}{|\max(\Delta)|} \end{aligned} \tag{24}$$

The labeling method consists in: for positive slopes, the prices with rise greater than or equal to 75% of the maximum ascent observed during the day are labeled as SURGE. Similarly, for negative slopes, prices with fall less than or equal to 75% of the minimum descent observed during the day are labeled as PLUNGE. For all other cases, the records are labeled as NOT RECOMENDED. For the first price traded in the day, also known as open price, the maximum and minimum slopes come from the last trading day. Figure 26 shows the results of the labeling process, compared with the actual prices of Microsoft (MSFT).

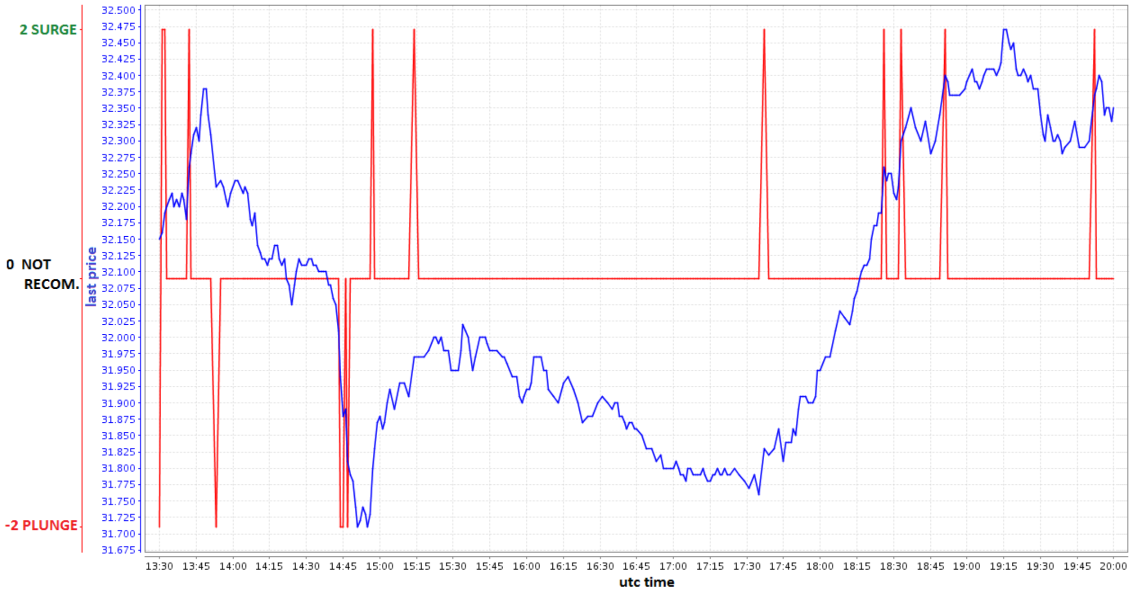


Figure 26 - Comparison of Microsoft (MSFT) real stock prices time series obtained in 14/Aug/2013, with one minute of interval (in blue), and the labelling method (in red).

In the real financial markets, the SURGES and PLUNGES are rare events, and this behavior was also observed in this work during the labeling process of stock prices in all companies, which leads to an imbalanced distribution of labels in the market data records, as can be seen in Figure 27.

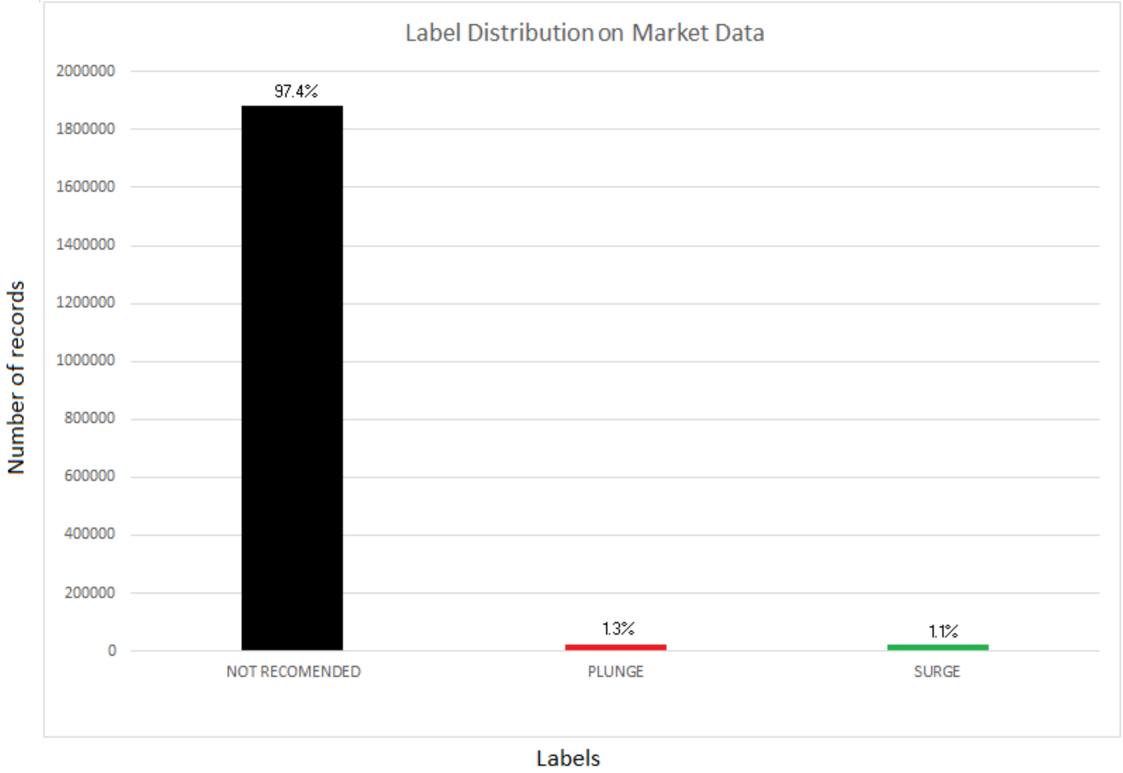


Figure 27 - Label distribution on market data records for all stocks, after alignment.

4.1.5 - News articles and prices alignment

Since the beginning of financial markets, the information exchange and news publication are responsible for directly affecting the stock prices. To correctly classify the news articles, they need to be labeled according to the changes in the stock prices, also known in the literature as price alignment.

The alignment between news articles and stock prices aims to label news articles, considering the labels SURGE, PLUNGE, NOT RECOMENDED which are already assigned to a set of prices from a specific company's stock, in a period close to the date and time the news article was published. In the literature, this period is called time offset, henceforward identified by τ.

Figure 28 shows an example of news articles and labeled stock prices in the same time series. Basically, the alignment process needs to assign a label to the news article, based in some criteria that considers the SURGE, PLUNGE, and NOT RECOMENDED labels existing in the stock prices, in a specified time offset. The decision criteria developed for this work relies on associating a label *r(C)* to a new article published at time *t*, given a set of stock prices labels *C={c(t-1), c(t), c(t+1), ..., c(t+τ)}*. The labeling function *r(C)* is explained by equation (25).

$$r(C) = \begin{cases} qs > qp \text{ } and \text{ } \Delta C > 0, SURGE \\ qs < qp \text{ } and \text{ } \Delta C < 0, PLUNGE \\ NOT\ RECOMENDED \end{cases} \tag{25}$$

Being *qs* the number of occurrences of SURGE and *qp*, the occurrences of PLUNGE, and the price delta before and after *C*, is represented as $\Delta C = c(t+w+1) - c(t-1)$. The rationale for this alignment proposal is that only a strong turnaround in the stock prices, and the continuous change of prices before and after the time offsets, will make it possible to identify the proper characteristics in the news articles for a profitable trading recommendation. The remaining news articles are labeled as NOT RECOMENDED.

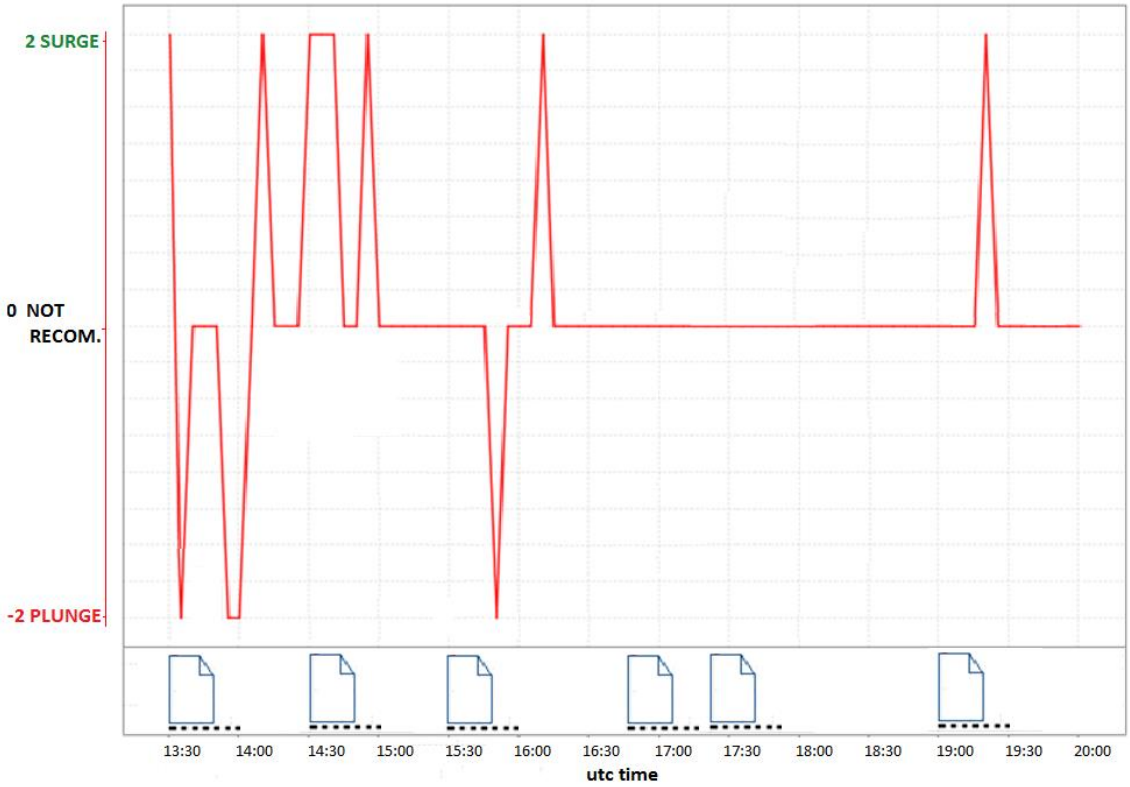


Figure 28 - A hypothetical example of news articles and labelled stock prices in the same time series. The time offset of 30 minutes is identified by the dotted line.

In order to observe how the market reacts to news articles, this research analyzed and compared the effects of news articles in the stock prices given a period after its publication. During the preliminary experiments conducted in this work, several values for time offsets, in minutes, were tested. The value of τ= *45* minutes was initially attempted, but this led to the generation of noise, because the accumulation of news articles and the variation of stock prices that occurred during this period, caused a set of similar documents to point to different labels, generating low performance results by the machine learning classifier. The same occurred for τ= *30, 15, 7,* and 5 minutes. Negative values for τ= *-15, -7, -5, -3, -2,* and *-1* minute, which denote an information leak, e.g., some price sensitive information was disclosed before to be officially released were also attempted, with no performance improvement and poor performance in some cases.

In this work, since the beginning the lower values in minutes for τ (τ=*1,* and *2)* led to better results in terms of performance and market simulation, and the rationale for this is after a long period, the market inevitably will be able to consume the released information, and this will be reflected in the stock prices. It was also observed that the low quantity of news articles accumulated in one or two minutes avoids the noise generation and class overlapping, because the news articles won´t accumulate along a wide time offset, facilitating this way the classifier decision. The experiments conducted in Chapter 5 focuses on analyzing and comparing the classification results and investment simulations, given a time offset of τ=*1, 2, 3, and 5* minutes.

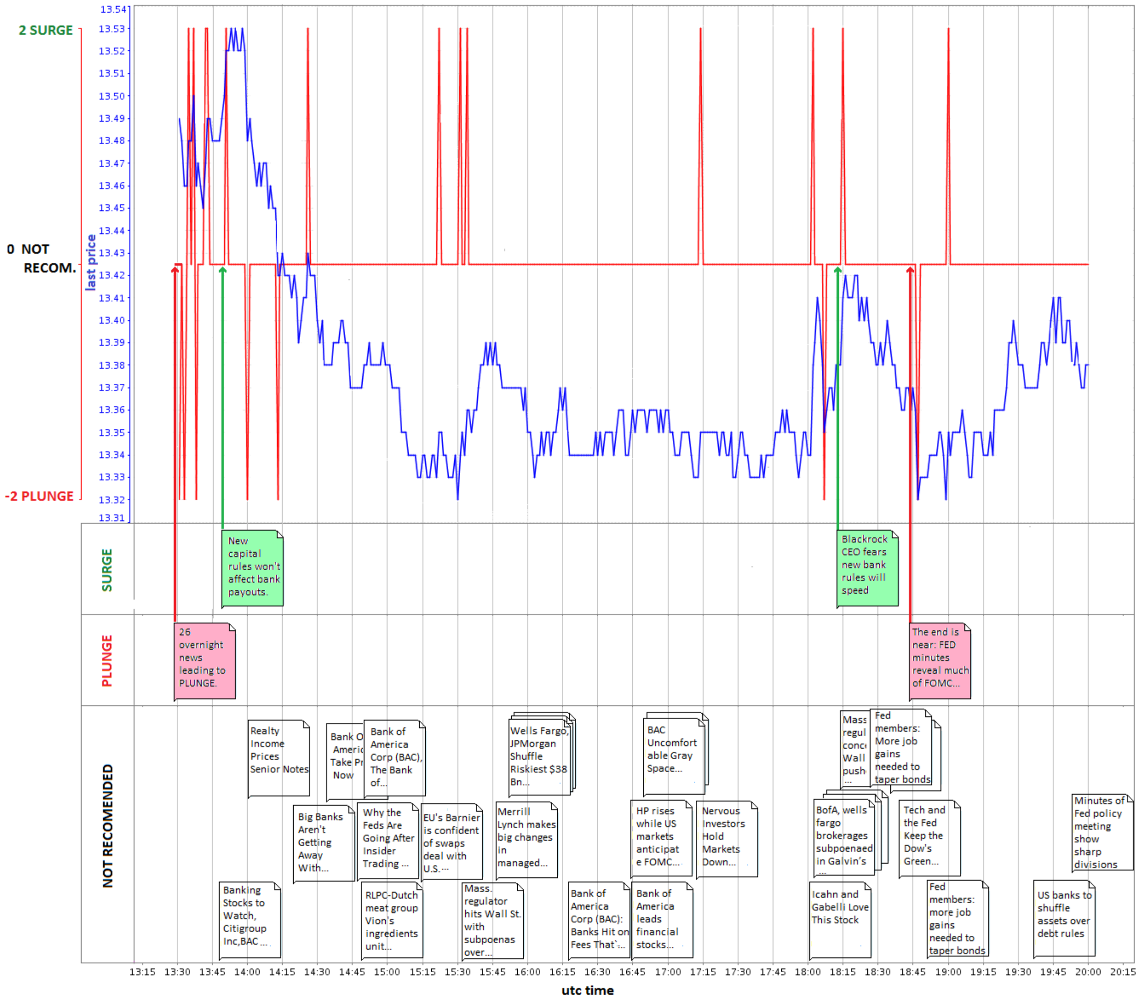


Figure 29 - Comparison of real stock prices from Bank of America (BAC) obtained in 10/Jul/2013 (in blue), and the stock labelling (in red). The bottom lines represent all the news articles gathered by the web crawler during that day.

Figure 29 depicts a time series with real data for Bank of America (BAC), and how these news articles can be labeled given the labeled stock prices and the proposed criteria in equation (25), for $\tau=1$. The news articles colored in green represent the SURGES, the pink ones are PLUNGES, and the white ones are NOT RECOMMENDED, given a time offset of one minute. It is noticed the high number of news articles labeled as NOT RECOMENDED, denoting an imbalanced dataset with class overlapping problems, as the opportunity to properly buy or sell shares of stocks or other securities are rare events. This same problem is present in the data for all the 30 companies studied, as demonstrated in Figure 30, and it makes part of the challenges faced in this work. The solutions proposed to solve these and other problems will be explained in the next sections.

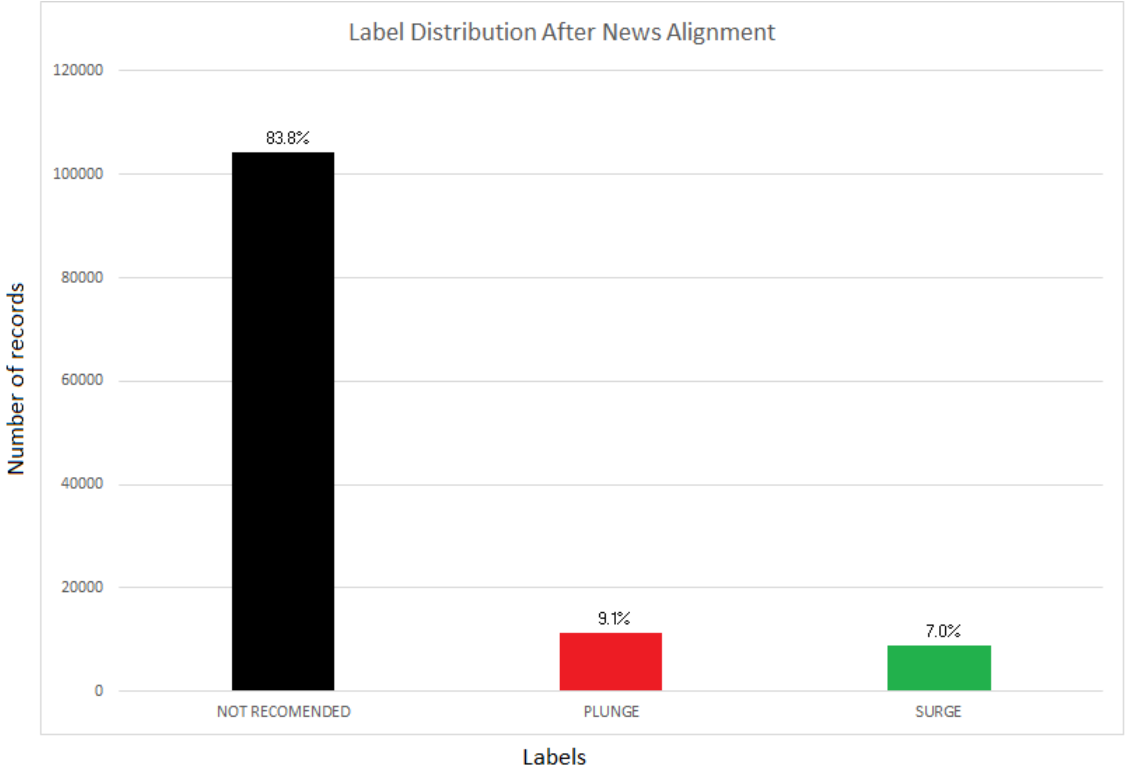


Figure 30 - Label distribution on the news articles for all stocks, after alignment, for τ=1.

4.1.6 – Database Storage

All the gathered news articles and market data records are stored in a MySQL database engine (Widenius, et al., 2002), forming this way the backing test data used in the experiments along this work. The entire TradeMiner database with its respective table indexes and data structures occupies 30 gigabytes of disk space.

Table 2 summarizes the number of records grouped by company stock.

Table 2 - Summary of gathered data grouped by company stock.

Stock Symbol	Company Name	Quantity of news records	Quantity of market data records
AA	Alcoa Inc	2,476	67,514
AXP	American Express Co	2,150	65,683
BA	Boeing Co	7,592	66,240
BAC	Bank of America Corp	9,189	73,214

CAT	Caterpillar Inc	2,877	66,544
CSCO	Cisco System Inc	541	48,802
CVX	Chevron Corp	3,704	65,124
DD	E. I. du Pont de Nemours and Co	1,374	64,983
DIS	Disney	4,914	66,589
GE	General Electric Co	5,659	67,890
HD	Home Depot Inc	2,619	64,658
HPQ	Hewlett-Packard Co	5,396	66,657
IBM	IBM	4,634	67,536
INTC	Intel	5,006	51,939
JNJ	Johnson & Johnson	3,739	64,816
JPM	JPMorgan Chase and Co	10,650	67,545
KO	The Coca-Cola Company	3,756	65,033
MCD	McDonalds Corp	3,448	64,896
MMM	3M Co	1,289	65,897
MRK	Merck & Co Inc	2,468	65,129
MSFT	Microsoft Corp	12,177	47,502
PFE	Pfizer Inc	3,382	66,494
PG	Procter & Gamble Co	3,245	65,063
T	AT&T Inc	5,996	66,223
TRV	Travelers Companies Inc	1,001	64,337
UNH	UnitedHealth Group Incorporated	369	64,181
UTX	United Technologies Corp	2,321	64,126
VZ	Verizon Communications Inc	5,768	64,926
WMT	Wal-Mart Stores Inc	5,525	64,399
XOM	Exxon Mobil Corp	4,930	65,582
Total of records		128,915	1,929,522

4.2 – Data splitting

4.2.1 – Split

With the purpose of obtaining a recommendation model properly adjusted for a time series, it is necessary not to contaminate the training data with information from the future. To achieve this, the dataset was kept in its chronological order, and split in training and test. The experiments conducted in this work are supported by nine months of data, six months for training and three months for testing.

This work observed a high level of noisy examples along the data, as will be explained in step 4.3.3. One of the measures to mitigate this problem is to perform a training with the new data every week, to adjust the model to the new reality and maximize the classifier efficacy. This technique is known in the literature as sliding window (Dietterich, 2002).

The training dataset incorporates six months of records, and the test dataset contains one week of new records to evaluate the model. As the processing advances to a new week, the training dataset incorporates the previous week, and discards the first week from six months ago, as shown in Figure 31.

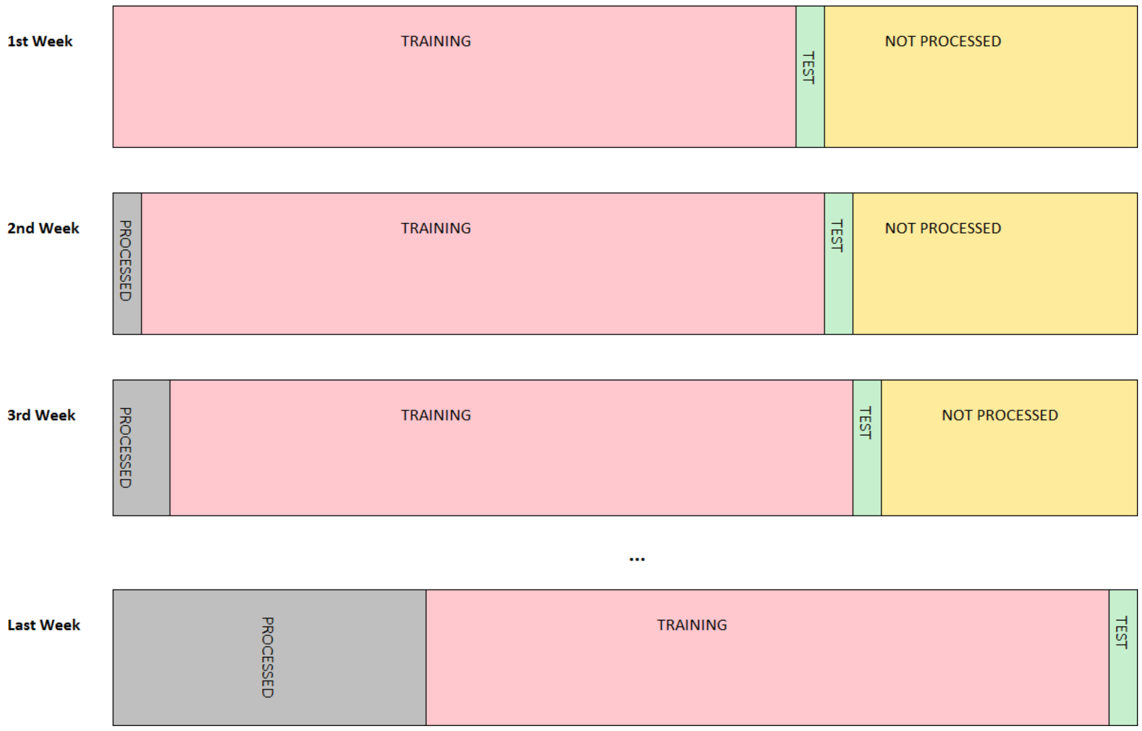


Figure 31 – Data splitting process.

Once the data is split, a new model must be rebuilt and tested. This process is repeated every week, for every stock symbol, until the end of the test data.

4.3 – Training

In the training phase, the raw text existing in the news articles will be transformed in vectors, then a predictive model and word lists will be built as input for the test phase.

4.3.1 – Feature Selection / Representation

With the labelled news articles stored, it's necessary to transform these documents into a structured format to be processed by the statistical and machine learning methods ahead.

The gathered documents were processed with the text mining extension of RapidMiner (Mierswa, et al., 2006) , which provides several operators for text processing such as tokenization, stemming, stop words, n-grams and integrated dictionaries, and the capability to use languages other than English.

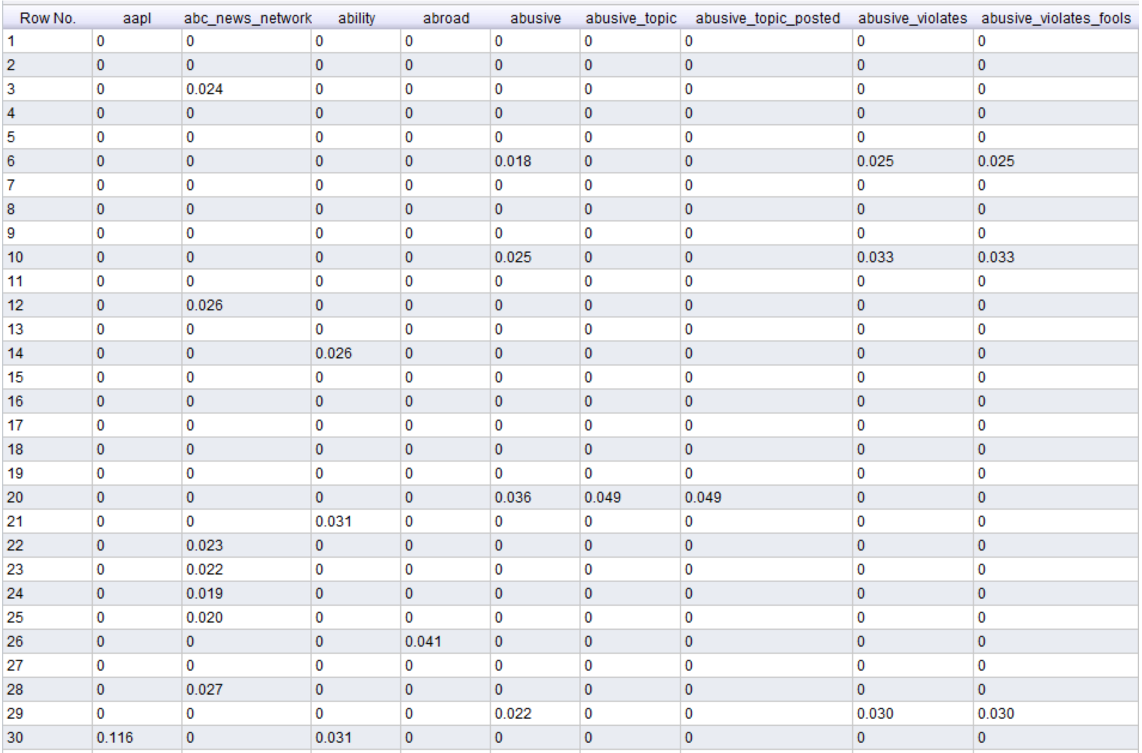
The entire set of documents, also known as corpus, is converted into a matrix, also known as bag of words (BOW), where each document is represented as a vector row, and the words and terms as the columns of this matrix. The first step to transform a corpus into a BOW is to parse the text content in words and terms for each document (tokenization), and generate a word list regarding the entire corpus. All the documents and respective word list are converted to lowercase.

Not all the words in the word list carries information, then both stop words and words with size less than two characters are removed. The words with frequency lower than 2% and greater than 95% are removed as well.

The use of stemming (Lovins, 1968), (Porter, 1980) was not applied in this work, as it has failed to produce satisfactory results during the initial experiments.

The automatic discovery of n-grams (Sidorova, et al., 2014), which consists in a series of consecutive words of size n, with the maximum n=3 used in this work, and its respective insertion in the word list, helped to reduce the dimensionality and carries the existing semantics from the original text, improving the classifier performance.

As a last step, the words are represented as a TF-IDF measurement (23). The generated BOW contains columns that represent the selected words and n-grams. The resulting word list represents the BOW column names, and the respective word frequency along the corpus. This word list is stored to transform the textual documents in the test dataset during the step 4.4.1.



Row No.	aapl	abc_news_network	ability	abroad	abusive	abusive_topic	abusive_topic_posted	abusive_violates	abusive_violates_fools
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0.024	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0.018	0	0	0.025	0.025
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0.025	0	0	0.033	0.033
11	0	0	0	0	0	0	0	0	0
12	0	0.026	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0.026	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0.036	0.049	0.049	0	0
21	0	0	0.031	0	0	0	0	0	0
22	0	0.023	0	0	0	0	0	0	0
23	0	0.022	0	0	0	0	0	0	0
24	0	0.019	0	0	0	0	0	0	0
25	0	0.020	0	0	0	0	0	0	0
26	0	0	0	0.041	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0
28	0	0.027	0	0	0	0	0	0	0
29	0	0	0	0	0.022	0	0	0.030	0.030
30	0.116	0	0.031	0	0	0	0	0	0

Figure 32 - Excerpt of a bag of words matrix with TF-IDF representation, from a corpus of news articles related to Exxon Mobil Corp (XOM).

Figure 32 shows an excerpt from the BOW with TF-IDF representation, generated from news articles related to Exxon Mobil Corp (XOM). It can be noticed a matrix with several 0 (zero) values, denoting a sparse matrix structure, which is common in a BOW document representation.

4.3.2 – Dimensionality Reduction

In the previous step, the use of n-grams and the removal of words with low and high frequency helped to reduce the number of words, represented as attributes in the BOW matrix. However, even after this processing there is a high number of attributes, also known as dimensions, with an average 8,500 attributes for each stock dataset. This high number of attributes could become an intractable problem for learning algorithms, also known as dimensionality curse, and three main difficulties were reported by (Donoho, 2000):

1- Difficulties of optimization by exhaustive enumeration on product spaces;

2- Approximating a general high-dimensional function;

3- Integrating a high dimensional function.

In this work, the high number of attributes increased the consumption of memory and computer processing, and even worse they generated noise. To have a more representative set of attributes and remove noise, the use of Pearson's Chi-Square statistic was applied with satisfactory results.

This process calculates the relevance of an attribute by computing the value of the Chi-Squared statistic (Pearson, 1900), (Forman, 2003) for each attribute with respect to the class attribute. At the end, the attributes with relevance lower than 0.10 were removed.

At the end of this step, the overall number of attributes (i.e. words), decreased from 8,500 to 2,600 on average, for each stock dataset.

4.3.3 – Data Balancing

When dealing with supervised learning, one of the major problems in classification activities lies in the treatment of datasets where one or more classes have a minority quantity of instances. This condition denotes an imbalanced dataset, which makes the algorithm to incorrectly classify one instance from the minority class as belonging to the majority class, and in highly skewed datasets, this is also denoted as a "needle in the haystack" problem (Weiss & Provost, 2001), due to the high number of instances from a class overcoming one or more minority classes. Nevertheless, in most cases the minority class represents an abnormal event in a dataset, and usually this is the most interesting and valuable information to be discovered.

To establish some notations that will be used in this step, the definition of an imbalanced dataset is: given a training set *T*, there is a subset with positive instances $\boldsymbol{P} \subset \boldsymbol{T}$, and a subset of negative instances $N \subset T$, where $|P| < |N|$.

Learning from imbalanced datasets is still considered an open problem in data mining and knowledge discovery (Duman, et al., 2012), (Thammasiria, et al., 2014), and it needs real attention from the scientific community (He. & Ma, 2013). The experiments performed in (Japkowicz, 2003) demonstrated that the class overlapping is commonly associated with the class imbalance problem, and this scenario was also confirmed by (Prati, et al., 2004) using synthetic datasets. The class overlapping can be defined as two or more instances sharing a similar set of attributes, but with different classes. The class overlapping can also be designated as a type of noise.

An empirical study was performed by (Weiss & Provost, 2001), to understand why classifiers perform badly in the presence of class imbalance. In (Qiong, et al., 2008) the authors conducted a taxonomy of methods applied to correct or mitigate this problem, and their study has found three main approaches: data adjusting, cost sensitive learning, and algorithm adjusting. In the data adjusting, there are two main sub-approaches: creation of instances from minority class (oversampling), and removal of instances from majority class (undersampling).

In section 4.1, the data gathering process and initial exploratory data analysis denounced a highly skewed distribution among the classes for this TMFP study, as demonstrated in Figure 27 and Figure 30. One of the actions taken to mitigate this problem was to merge the class PLUNGE into the NOT RECOMMENDED class, simplifying the decision boundaries along all the dataset. This merge comes with a drawback: now the TradeMiner recommendation engine will be able to predict only SURGES and NOT RECOMMENDED. Nevertheless, it was observed that the news articles pointing to a SURGE are not directly associated with new articles pointing to a PLUNGE in a short period. As the investment strategy applied in this work uses a very short hold period, it showed to be worthless to predict PLUNGES to define when to sell the stocks. In spite of this, the investment simulation conducted in Chapter 5 demonstrated that the recommendation with two classes are still profitable, if there are good quality predictions. It is also applicable for a future work to train a new set of predictive models using the class PLUNGE, and the class NOT RECOMMENDED merged with SURGE, in order to predict when to sell the assets. The Figure 33 demonstrates the distribution of label values after this merge.

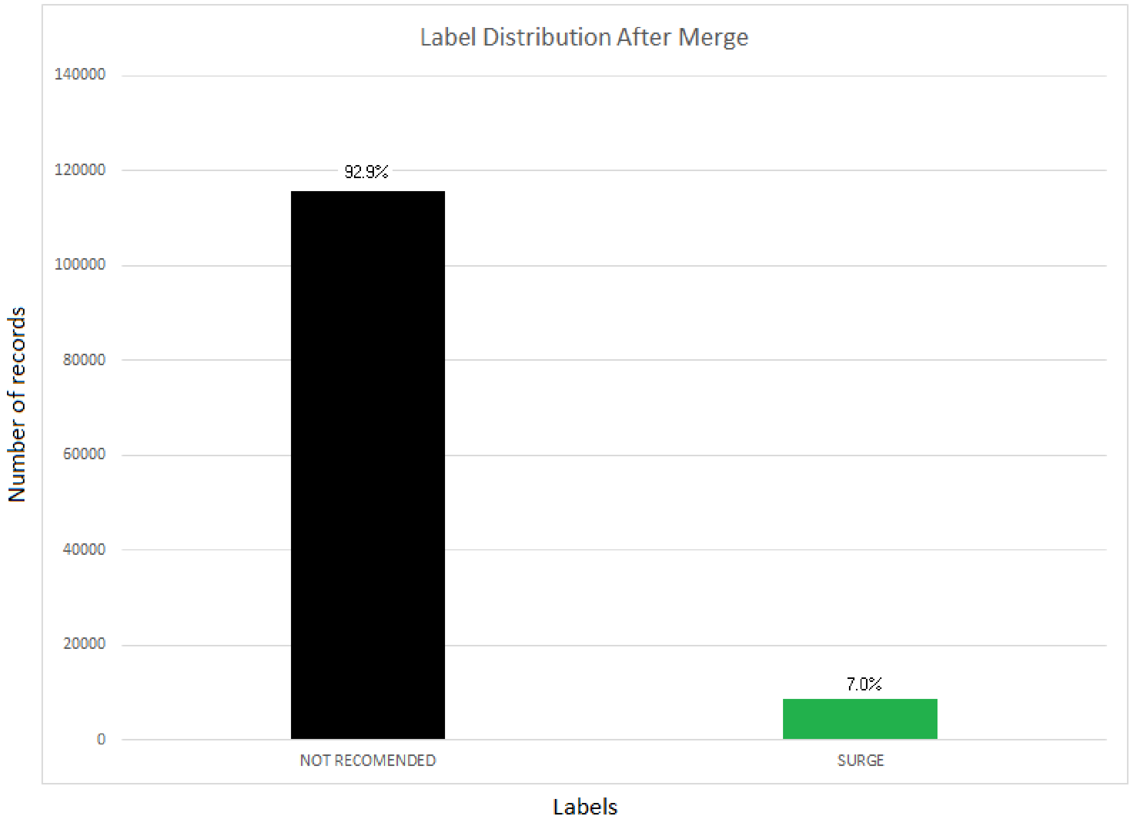


Figure 33 - Label distribution on the news articles for all stocks, after the merge of PLUNGE to NOT RECOMENDED, for *τ=1*.

Despite all the benefits cited above, as can be seen in Figure 33, the skewed distribution became more evident. To solve this problem an undersampling data adjusting algorithm called KNN Undersampling (KNN-Und) (Beckmann, et al., 2015) was applied to the training dataset.

The KNN-Und method works removing majority classes from *N* subset, based on its *k* nearest neighbors, and it works according to the steps below:

1- Obtain the *k* nearest neighbors for $x_i \in N$;
2- x_i will be removed if the count of its positive neighbors $\in P$ is greater or equal to *t*;
3- The process is repeated for every majority instance of the subset *N*.

The parameter *t* defines the minimum count of neighbors around x_i belonging to the *P* (minority) subset. If this count is equal or greater than *t*, the instance x_i will be removed from the training set *T*. The valid values of *t* are $1 \leq t \leq k$ and as lower *t* is, as aggressive is the undersampling. This algorithm can also be used in multiclass problems, as in the negative subset *N* may contain instances from several majority classes. In this work the Cosine Distance (Sidorov, et al., 2014) was used to calculate the neighbor's distances.

KNN-Und only acts in the class overlapping areas, because an instance from majority class will only be removed if a number *t* of instances from other classes are present in its neighborhood. In the cases that an instance of the majority class is not surrounded by *t* instances of other classes, that instance will not be removed. This situation only occurs in non-overlapping areas. Despite this behavior, in our experiments *t=1* was kept in most of the cases, because the KNN-Und only acts in overlapping areas. The non-overlapping areas, which are far from the decision surface are kept untouchable. This explains why the KNN-Und can also be used to solve the class-overlapping problem, which is commonly associated with imbalanced datasets (Japkowicz, 2003).

The KNN-Und can be considered a very simple algorithm, and it has the advantage of being a deterministic method, as there is no random component. Nevertheless, like in all the data balancing methods, the KNN-Und requires a priori information about the underlying class distribution, in this case, the number of neighbours from majority and minority class.

For predictive analytics, the test set must be completely isolated, for this reason, the KNN-Und can be applied only in the training dataset to construct a more robust model. Figure 34 demonstrates the effect before and after applying KNN-Und on a training dataset. It can be noticed the KNN-Und makes a slight (but guided) removal of 3.6% of records from the majority class.

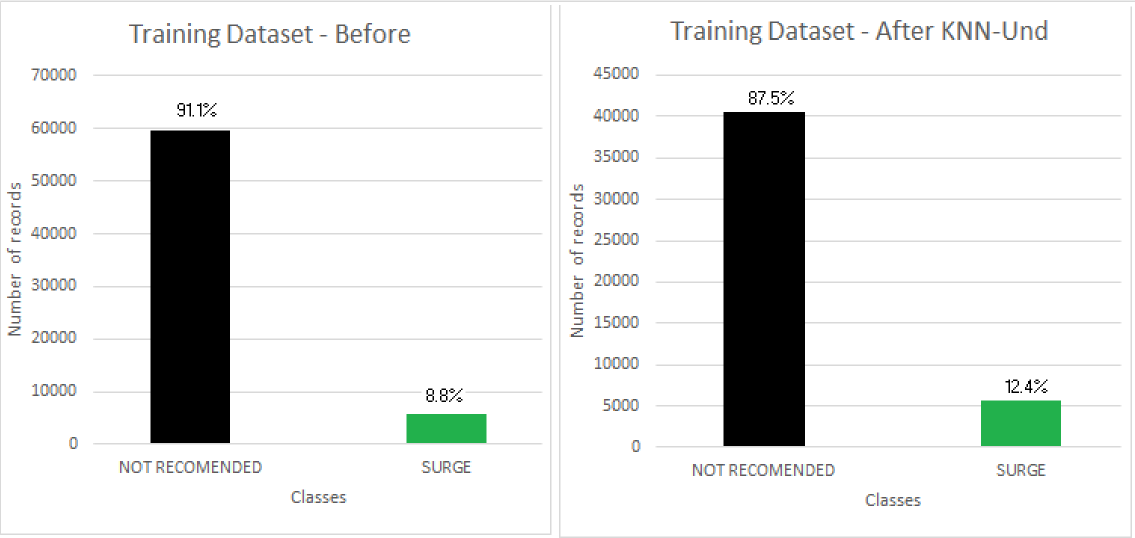


Figure 34 - Training dataset for all stock symbols, from Jan/2013 to May/2013, before and after to applying the KNN Undersampling algorithm.

The t-SNE (Van der Maaten, 2014) is an algorithm for dimensionality reduction well suited for the visualization of high dimensional datasets. Figure 35, Figure 36, Figure 37, and Figure 38 depict a 2D scatter plotter obtained with t-SNE. The 0 and + represent the NOT RECOMMENDED and the SURGE instances in the training set. The bold 0 and X represent the NOT RECOMMENDED and the SURGE instances in the test set.

The Figure 35 and Figure 37 depict an excerpt of data from two companies, before the KNN-Und. The groups where the positive examples in the test set occurs (represented by X) are marked with a labelled circle. The Figure 36 and Figure 38 represent the same excerpt of data after the application of KNN-Und. There is a displacement of points between the figures before and after, and this is due to the stochastic behaviour of t-SNE algorithm, as the initial conditions in the dataset changed after to apply the KNN-Und.

Comparing the figures before and after, it can be noticed that some negative examples were removed from the training set (represented by 0). This removal occurs only around the positive examples from training set (represented by +), and now the positive examples from test set (represented by X) are surrounded by less negative examples than before. The guided removal of negative examples from the training set with KNN-Und helps the classifier to adequate a model in the presence of class imbalance and class overlapping.

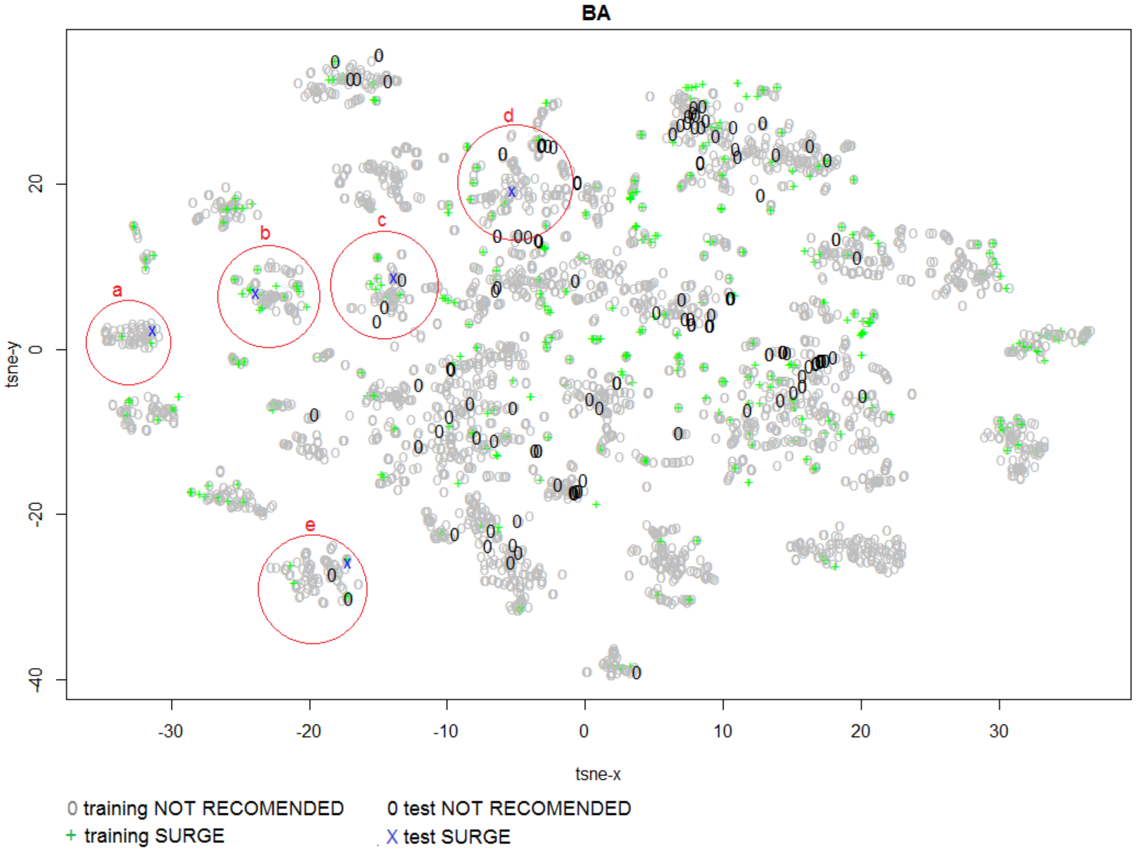


Figure 35 - t-SNE scatter plot from Bank of America (BA), before the KNN-Und removal.

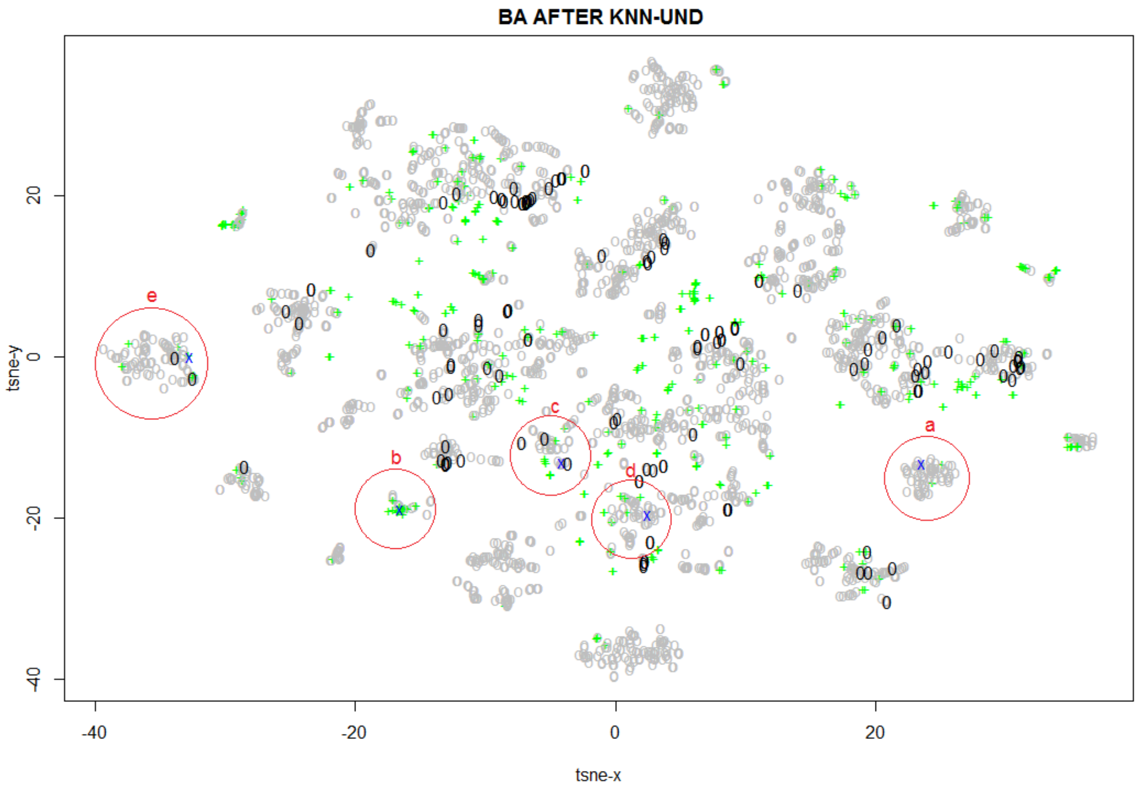


Figure 36 - t-SNE scatter plot from Bank of America (BA), after the KNN-Und removal.

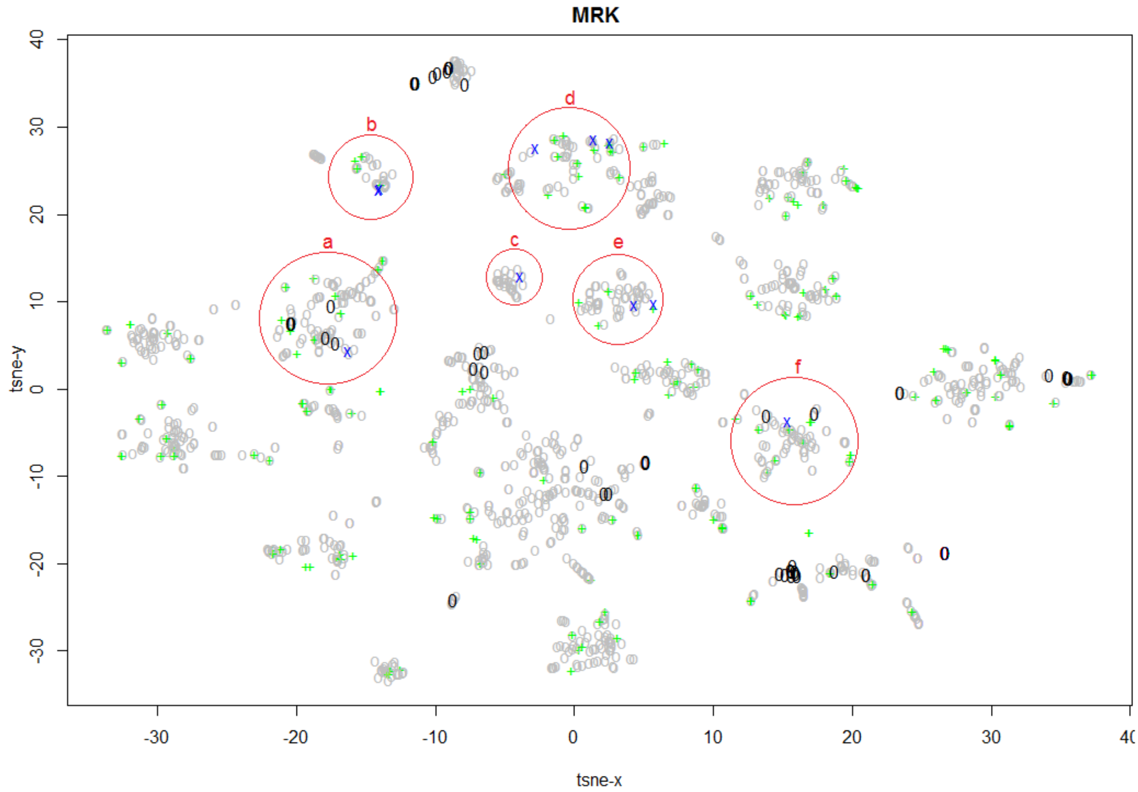


Figure 37 - t-SNE scatter plot from Merk (MRK), before the KNN-Und removal.

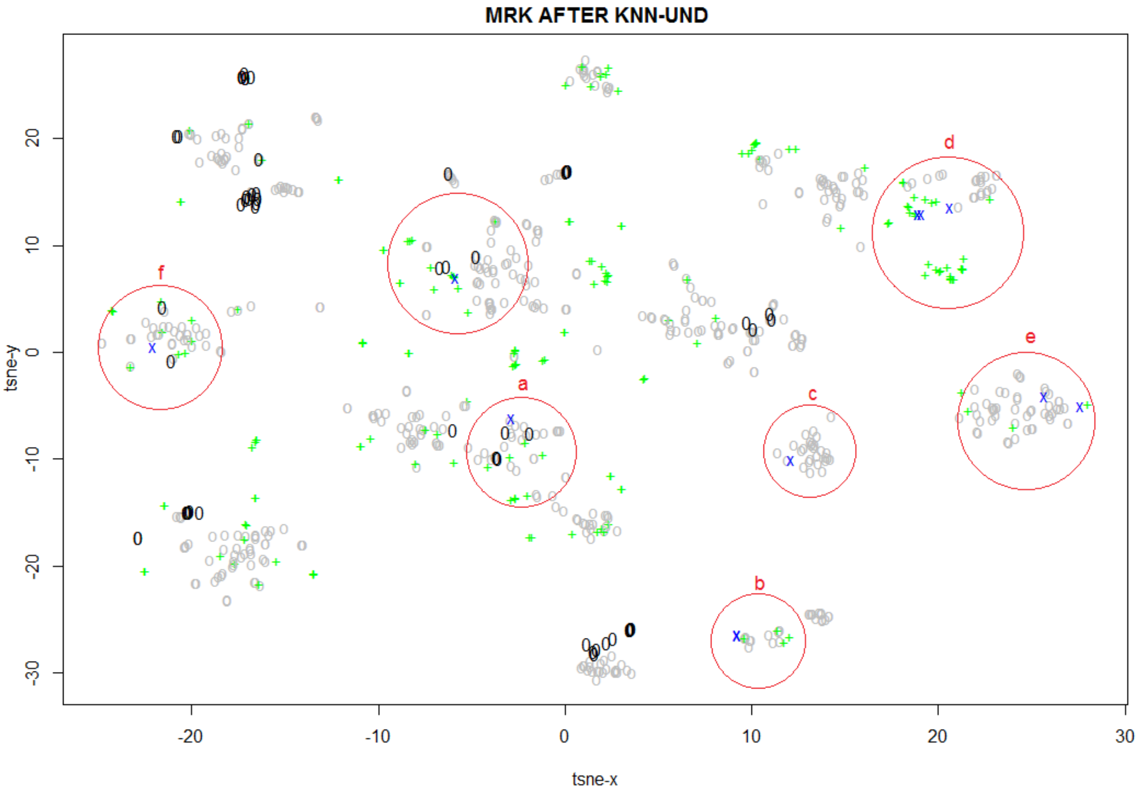


Figure 38 - t-SNE scatter plot from Merk (MRK), after the KNN-Und removal.

As a sub-product of the instances removal, the KNN-Und helped to create a black list to be used with the test dataset. This black list contains noisy sources of information, for example, a website that always posts alarming news that could lead to a SURGE, but that does not cause expressive changes in the stock prices, being in fact a NOT RECOMENDED. To avoid this problem, if all news articles from a specific website were removed by KNN-Und, this website will be included in the site blacklist, to be used during the test phase to avoid noise and incorrect predictions, as the news articles in the site blacklist can be marked as NOT RECOMMENDED or simply removed.

According the initial experiments, the site blacklist helped to avoid false positives during the test phase, and it was observed that the results of G-Mean, AUC, and F-Measure improved up to 2.00 after applying this approach.

The experiments demonstrate that the use of KNN-Und only in the training dataset helped the classifier to adjust its model to the imbalanced and noisy conditions found in this work, and improvements up to 8.00 in terms of G-Mean, AUC, and F-Measure were observed since this approach started to be applied.

4.3.4 – Training

This work applied the Support Vector Machine (SVM) (Cortes & Vapnik, 1995) as the machine learning algorithm, with the LIBSVM implementation (Chang & Lin, 2011), and the Radial Basis Function (RBF) as kernel.

The parameters C and Gamma required by SVM are adjusted through a grid search, using the training dataset with a 10-fold cross validation to discover the best value of F-Measure obtained with the SVM classifier, given a pair for C and Gamma parameters, as described in (Hsu, et al., 2003).

4.4 – Test

Once the training phase is complete, it generates the word list used to construct the bag of words (BOW), the word weights generated by feature selection, the site blacklist generated by KNN-Und and the predictive model generated by classifier.

The test phase aims at predicting the label in the test dataset, using the predictive model and other information generated by the training phase, and at the end of the test phase, a new column “prediction” containing the classifier prediction will be added to the test dataset. The next steps explain how this is done.

4.4.1 – Feature Selection

Similarly to the training phase, the textual documents existing in the test dataset must be transformed into a BOW, for this, the same feature selection process executed in step 4.3.1 will be repeated.

In supervised learning, the training and test datasets must have the same column names and data structure, otherwise the predictive model generated by the classifier won't work.

The word list generated during the step 4.3.1 is used to generate a BOW with the same column names from the training dataset. The word frequency that comes with the word list are used to calculate the TF-IDF. At the end, training and test datasets will have the same data structure.

4.4.2 – Feature Removal

The test dataset still has a high number of columns, then the word weights generated previously by step 4.3.2 will be used to prune the words with relevance *<0.10*. The test dataset now contains the final data structure necessary to be recognized by the classifier model generated in the step 4.3.4.

4.4.3 – Test

Given the existing features on each example in the test dataset, the classifier model generated during the step 4.3.4 will predict its actual label value (SURGE, NOT RECOMMENDED). The predicted value is stored in a new column called "prediction".

4.4.4 – News Aggregation

As mentioned in the data gathering section, the news articles are organized in time series, and each news article owns a publication date and time (converted from EST to UTC). For the news occurring out of trading hours, the publication date and time to be considered is the exchange opening time in the next available trade date, e.g., the next trade date and time for a news article published Monday night will be Tuesday 14:30 UTC, and for a news article published Wednesday 15:50 UTC will be Wednesday 15:50 UTC. All the news articles are strictly stored and processed in their chronological order.

During the experiments, the occurrence of several news articles in the same time offset was observed, especially if they accumulate overnight or during the weekend, and it is also common to have several news articles being published at the same minute when the market is already open.

As a recommendation engine, it is only necessary to have one recommendation of SURGE or NOT RECOMMENDED for each time offset, i.e., for $\tau=1$, in the same minute, because there is only one actual label value to be predicted in the time offset. For this reason, the news articles and the respective prediction generated by the classifier must be aggregated in the same time offset, and only one decision must be taken and passed ahead as a recommendation.

To provide a unique decision given a set of documents in the same time offset, and to improve the level of true positives along the investment recommendation, this work proposes a novel ensemble approach named Cascading Aggregation for Time Series (CATS). This algorithm aggregates the news articles in the same time offset, their predictions are counted, and a final prediction is taken based on the counting of new articles predicted as SURGE or NOT RECOMMENDED.

A challenge faced with this approach for TMFP problems was how to decide the adequate thresholds to define when a set of documents predicted as SURGE and another set of documents predicted as NOT RECOMMENDED, in the same period, will lead to a correct and unique prediction. To solve this problem, a genetic algorithm (GA) was built to create decision rules.

GA is a heuristic applied in search problems and optimization. GA is a method inspired by nature, having probabilistic and non-deterministic characteristics (Holland, 1975), (Goldberg, 1989), (Whitley & Sutton, 2012). The search problem must be defined as a chromosome codification $x= [g_1, g_2, ..., g_n]$, being g_j a gene, and a fitness function $f(x_i)$, also known as objective function. The GA applies operators like cross over and mutation, that imitate the evolutionary process existing in nature, to find the fittest individual s (also known as fittest solution) after a specified number of generations or a stop criteria, being $s = [z_i, x_i]$, and $z_i = f(x_i)$. The fittest solution given a stop criteria can be considered a local optima, and not necessarily is the best possible solution, also known as global optima.

The developed GA creates decision rules and defines thresholds given the counting of news articles predicted as SURGE or NOT RECOMMENDED. The fitness function is the F-Measure resulting of the SVM classifier, applied to the training dataset with 10-fold cross validation. Table 3 shows the chromosome codification, with values for each gene between brackets, and Table 4 shows the respective parameters used for GA.

Table 3 – The chromosome codification for GA in the CATS algorithm.

Threshold values for SURGE (-1 means no comparison for SURGE)	[-1 to 15]
Boolean operators for SURGE	[<, <=, =, >=, >, < >]
Boolean conjunction	[and, or]
Threshold values for NOT RECOMENDED (-1 means no comparison for NOT RECOMENDED)	[-1 to 20]
Boolean operators for NOT RECOMENDED	[<, <=, =, >=, >, < >]

Table 4 – The GA parameters for threshold optimization used in the CATS algorithm.

Structure/ Parameters	**Value**
Fitness function	F-Measure
Maximum generations	80
Early stop	10 generations without improvement
Population Size	50
Keep best	Yes
Selection	Tournament, 0.25
Crossover probability	0.9
Mutation type	Gaussian

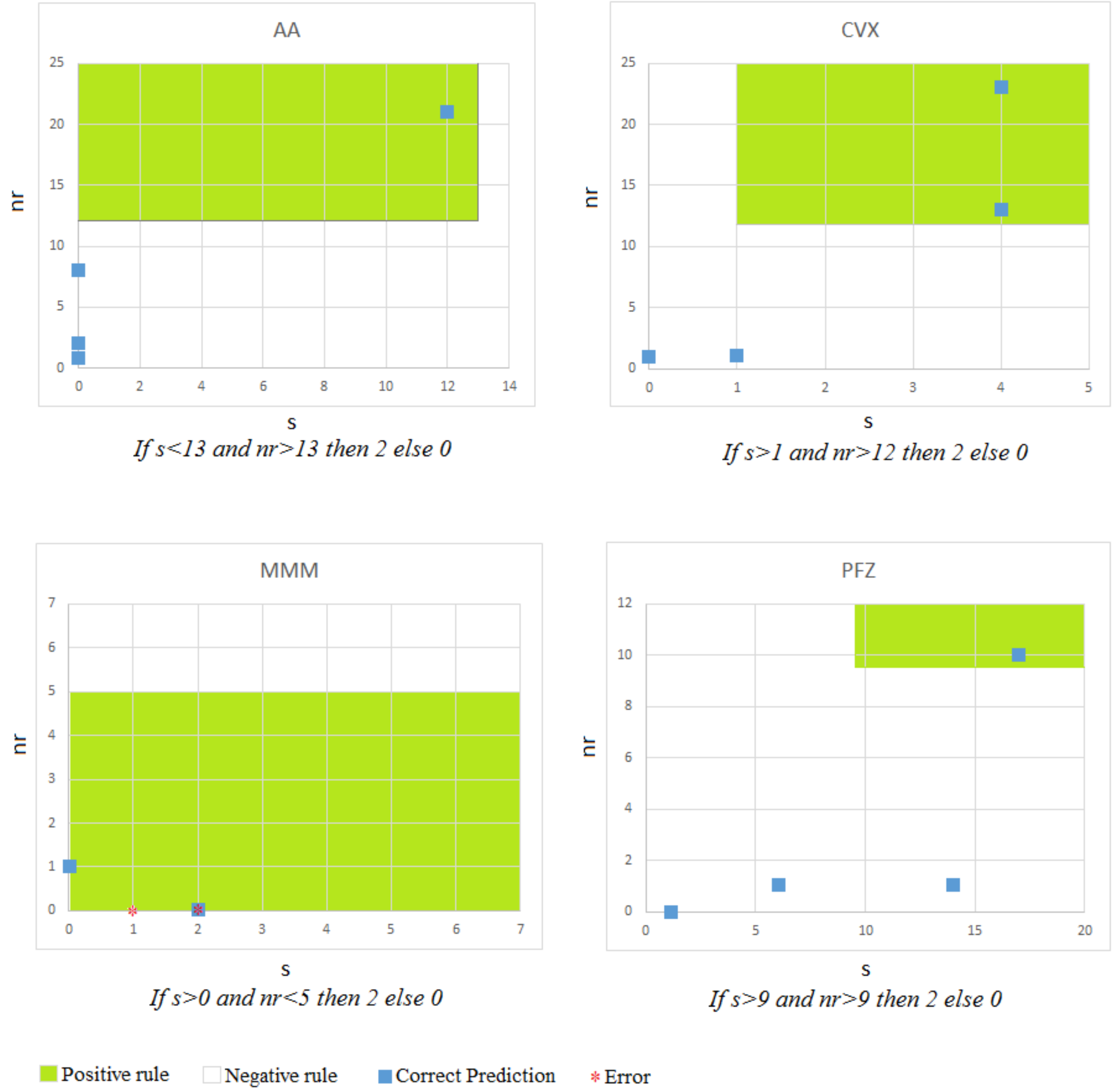
To avoid that suspicious news content affects the prediction count, and for processing convenience, the news articles from websites existing in the KNN-Und blacklist are removed before the counting.

Table 5 shows four examples of decision rules created by the GA, applied to an excerpt from the test dataset, with the counting of SURGE (the positive class) and NOT RECOMMENDED (the negative class) predictions obtained from step 4.4.3. Each line represents a time offset in one minute ($\tau=1$) and the respective ensemble decision. To facilitate the visualization, the label values in the columns “Actual Label” and “Ensemble Decision” were kept with its numerical representation, being 0=NOT RECOMMENDED and 2=SURGE.

The same data from Table 5 is represented in the plot charts from Figure 39, with the counting of SURGE (the positive class) and NOT RECOMMENDED (the negative class) predictions from section 4.4.3, and the respective decision areas given a rule evolved by the GA at the bottom. The decisions for SURGE are represented in green (positive rule), and the decisions for NOT RECOMMENDED are the white area (negative rules). The y axis represents the number of NOT RECOMMENDED (nr), the x axis represents the number of SURGES (s). The blue squares are the correct predictions, and the red asterisks are the incorrect predictions. It can be observed that the decision rules created a linear boundary among the prediction counting.

Table 5 - Excerpt of test dataset from four stocks, with the prediction counting in the same time offset (*τ=1*), and the respective ensemble decision, given a decision rule created by GA.

Stock Symbol/ Company Name	Decision Rule	Date Time UTC (yyyy/mm/dd hh:mi)	NOT RECOMENDED Count (nr)	SURGE Count (s)	Actual Label	Ensemble Decision	Error
AA/ Alcoa Inc.	*If s<13 and nr>13 then 2 else 0*	2013/06/12 18:25	2	0	0	0	
		2013/06/12 20:20	2	0	0	0	
		2013/07/02 14:30	8	0	0	0	
		2013/07/10 14:30	21	12	2	2	
		2013/07/10 14:31	1	0	0	0	
		2013/07/16 19:19	1	0	0	0	
		2013/07/30 14:30	2	0	0	0	
CVX/ Chevron Corp	*If s>1 and nr>12 then 2 else 0*	2013/06/05 18:47	1	0	0	0	
		2013/06/17 14:30	23	4	2	2	
		2013/06/26 17:24	1	0	0	0	
		2013/07/09 15:15	1	0	0	0	
		2013/07/22 14:30	13	4	2	2	
		2013/08/02 14:49	1	0	0	0	
		2013/08/02 14:54	1	1	0	0	
MMM/ 3M Co	*If s>0 and nr<5 then 2 else 0*	2013/06/07 14:30	0	1	0	2	*
		2013/06/14 18:46	0	2	2	0	*
		2013/06/17 16:08	1	0	0	0	
		2013/06/27 16:03	1	0	0	0	
		2013/07/01 19:29	1	0	0	0	
		2013/07/12 14:30	0	2	2	2	
		2013/08/23 17:28	0	1	0	2	*
PFZ/ Pfizer Inc.	*If s>9 and nr>9 then 2 else 0*	2013/06/10 14:30	14	1	0	0	
		2013/06/13 16:10	1	0	0	0	
		2013/07/25 19:28	1	0	0	0	
		2013/07/30 14:45	1	0	0	0	
		2013/07/30 14:46	1	0	0	0	
		2013/07/31 14:30	17	10	2	2	
		2013/08/07 14:30	6	1	0	0	



If s<13 and nr>13 then 2 else 0

If s>1 and nr>12 then 2 else 0

If s>0 and nr<5 then 2 else 0

If s>9 and nr>9 then 2 else 0

Figure 39 - Representation of decision rules and respective prediction adjustments made by CATS, given the data from Table 5. The y axis represents the number of NOT RECOMMENDED (nr), the x axis represents the number of SURGES (s).

It can be observed in Table 5, for CVX at 2013/08/02 14:54, there is one new article predicted as NOT RECOMMENDED, one new article predicted as SURGE, and according to the decision rule (*If s>1 and nr>12 then 2 else 0*), both examples must be labelled as NOT RECOMMENDED, which leads to a correct prediction as a true negative. A similar situation occurred for PFZ at 2013/06/10 14:30 and PFZ at 2013/08/07 14:30.

The SURGE decision taken for AA at 2013/07/10 14:30, given the decision rule (*If s<13 and nr>13 then 2 else 0*), with 21 news articles predicted as NOT RECOMMENDED, whilst 12 news articles were predicted as SURGE, led to a correct prediction of true positive, and the incorrect prediction for 21 news articles was properly adjusted. The same occurred for CVX at 2013/06/17 14:30, CVX at 2013/07/22 14:30, and MMM at 2013/07/12 14:30.

As described in section 4.1, due to the highly skewed distribution among the classes in the datasets, there is a high quantity of time offsets without a SURGE prediction, for example AA at 2013/07/02 14:30. In these cases the CATS algorithm did not interfere on the prediction results, and they were correctly predicted as true negatives.

Due to the noisy nature of TMFP, the CATS algorithm cannot create a general rule for all situations, and maybe a decision rule is liable of failure in some cases, and make the correct prediction in other cases. In Figure 39, this situation is represented by a blue square overlapped by an asterisk in the chart for MMM stock. The failed adjustments are marked in the "Error" column of Table 5. Despite of this, according to the experiments demonstrated in chapter 6, the decision rules evolved by the GA in the CATS algorithm helped to mitigate this problem, and improved the predictive performance and simulation results.

The concept of combining different classifiers to get a weighted vote of their predictions or a model adjustment is known as Ensemble Learning, and a reasoning about the use of ensemble classifiers is presented in (Dietterich, 2001). The most known ensemble algorithms are Bagging (Breiman, 1996), and Boosting (Schapire, 2003), and in fact, these methods were tried during the experiments, but the results were not satisfactory. Another ensemble approach is called Cascading, which uses the output of one classifier as input to another classifier (Gama & Brazdil, 2000).

CATS can be considered a cascading ensemble approach, because the output of one algorithm (i.e., the prediction counting from SVM in the same time offset), is used as input by the GA[8] to evolve the decision rules. The CATS algorithm also shares some similarities with the consensus estimation technique used in fundamental analysis, which is applied when several analyses from one company are available, then each analysis is considered, but only one decision must be taken (the consensus).

[8] According to (Domingos, 2012), in the current context a GA can be considered a learning algorithm, because there is a model representation, an objective function and an optimization process.

The CATS algorithm developed in this work uses a linear decision rule to adjust a set of predictions made in the same period, providing a single predictive result at the end of the process. The experiments for *τ=1* showed a maximum improvement of 8.30 in terms of F-Measure, demonstrating that the CATS algorithm mitigated the effects of class overlapping and reduced the variance of results, helping this way to increase the number of true positives and true negatives, and improving the classification performance. These results also indicate that the CATS algorithm is a promising approach that deserves further investigation in a future work, for example, the use of normalized counting values, new measurements, and other attributes to be included in the decision process.

4.5 – Evaluation

The evaluation is the last step of the process, and it checks the quality and applicability of the predictive models constructed during the previous phases.

4.5.1 – Model Evaluation (Good Model?)

During this step, the output of the test phase containing examples with the actual label and the classifier prediction is submitted to the evaluation measure G-Mean (Barandela, et al., 2003), as described in section 2.3. As in the evaluation phase of the CRISP-DM process, as depicted in Figure 3, this step decides if the model will be deployed, or if the entire process must be revisited and adjusted.

In this work, the TradeMiner recommendation engine trained 30 predictive models generated from 30 datasets, each one belonging to one company listed on DJIA index. Not necessarily all the 30 predictive models must perform well in obtaining profits in the stock market. In fact, if it was possible to predict all the movements of a single financial instrument, that would be more than enough to obtain high return rates.

As a quality threshold of the predictive models, if at least 10 predictive models have a minimal value of *G-Mean >= 55.00*, this will be enough to go ahead to the next step, the investment simulation, otherwise, the parameters and algorithms along the data mining process must be revisited and adjusted.

4.5.2 – Investment Simulation

The investment simulation step checks if the predictions generated by the recommendation engine are profitable, then an investment simulator was developed for this purpose. This investment simulator receives the predictions from the test phase (which also contains the considered published date and time), and the market data related to the stock symbol under simulation as input. In this work, the investment simulation will use the predictions regarding the news articles published between 03/Jun/2013 and 03/Sep/2013 (3 months of test dataset).

This simulation relies on two assumptions: the stocks shares will be available to buy and sell at the moment they are requested, and the transactions have zero cost, which is common in similar evaluations (the transaction costs are easily absorbed by increasing the volume of each transaction, as long the trades are profitable).

A simple investment strategy was developed, which is like the one described by (Lavrenko, et al., 2000): If a SURGE prediction occurs, purchase $10,000 of shares from the related stock at τ*-1* minutes after the news article being published (i.e., for τ*=1*, 0 minute, or as soon as possible; for τ*=2*, one minute after the news article being published, etc.). Hold the stock position for *n=3* minutes, if during that *n=3* minutes the stock can be sold to make a profit of *>=2%*, then sell it immediately. At the end of *n* minutes, the stock is sold at the current market price, and take a loss if necessary. In real investment scenarios, this short-term strategy can be executed using HFT, that can accomplish in microseconds the execution of investment orders (Johnson, 2010). For further details about HFT, please visit section 3.1.5.

4.5.3 – Good Simulation?

One of the main criteria to analyse an investment result, which is also present in the related literature, is the rate of return. The rate of return is the profit (or loss) of an investment over a period. The rate of return for a single period, also known as rate of return by roundtrip (being roundtrip the complete operation of buy and sell a stock share), is defined by (26).

$$r = \frac{V_f - V_i}{V_i} \tag{26}$$

Where V_f is the final value, and V_i is the initial value.

The rate of return over *n* periods, also known as cumulative return (CR), is defined by (27).

$$\sum_{i=1}^{n} r_i = r_1 + r_2 \dots + r_n \tag{27}$$

Just as a comparison of the results of investment simulation with a risk-free asset, the rate of return from United States Treasury Bond (US T-Bond)[9], starting from June/2013, with three months of maturity, was 0.05%.

Another comparison approach is simply to check if the TradeMiner predictions are more profitable than to use a random trader as a predictive input for simulation. Nevertheless, for the sake of Precision, the statistical significance of predictive models given their rate of return was verified through a null hypothesis test (Neyman & Pearson, 1933). The null hypothesis has an analogy to a criminal trial, in which the defendant is assumed to be innocent (null is not rejected) until he is proven guilty (null is rejected) given a statistically significant degree. The null hypothesis H_0 in this work is to use a random trader as a recommendation engine for investment, and the alternative hypothesis H_a is to use TradeMiner as a recommendation engine for investment. The price change prediction in stock market is assumed to be aleatory (null is not rejected) until it is proven

[9] https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield

the results of a TMFP recommendation engine (in this case, the TradeMiner) has no relationship with the results of a random trader (null is rejected), given a *p-value* test.

The p-value can be calculated by any statistical significance test. In this work, the one sample t-test (Gosset, 1908) will be used to define the p-value, as explained by equation (28).

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \tag{28}$$

Where:

- μ_0 is the population mean, in this case, the mean of rate of return given by the simulation with random trader;
- $\bar{x}$ is the sample mean, in this case, the mean of rate of return given by the simulation with TradeMiner;
- *s* is the sample standard deviation;
- n is the sample size, being $n=10$ runs of simulation with TradeMiner.

The value of *t* is the p-value itself, and the bigger is the *t*, the higher the confidence to reject the null hypothesis. To define a confidence threshold, a cut point of α=2.821 is established, corresponding to the value from t-distribution table at 99% of confidence, nine (10-1) degrees of freedom, and one tail verification (Federighi, 1959).

If p-value> α, the null hypothesis H_0 will be rejected (the random trader), and the alternative hypothesis H_a will be accepted (the TradeMiner).

The t-test was chosen because the t-student distribution can handle small samples more appropriately. In this work, to obtain a sample means to execute the training, test, and simulation phases, and this must be repeated for all 30 stock symbols, which is computationally expensive. For this reason, a sample size of 10 is defined as the 10 runs the TradeMiner algorithm and the investment simulation was executed.

The simulation using a random trader as input is computationally cheaper, then the entire process of investment simulation will be repeated 100 times (i.e., the population size is 100). The random trader will iterate over the news articles grouped in the same minute, and it will make a recommendation of SURGE or NOT RECOMMENDED, using a pseudo random number generator with Gaussian distribution, following the same class distribution in the test dataset. The percentage of SURGE examples in the entire dataset is about 7% (Figure 33), and this percentage is adjusted according to the underlying test dataset, and it is expected (but not guaranteed, as this is an aleatory selection) that the random trader selects around 7% of test examples as SURGE.

Even with a good simulation in terms of rate of return, there are other approaches that take into account the balance of risk and reward, e.g., Sharpe Ratio (Sharpe, 1994), Risk adjusted return on capital (Herring, et al., 2010), Calmar Ratio (Young, 1991) , etc. However, the use of these financial tools and the decision to apply a recommendation model in a real investment scenario are beyond the objectives of this work.

4.5.4 – Real Investment Recommendation

In case of a good simulation led to the decision to apply the recommendation model in a real investment scenario, the Figure 40 depicts how the predictive model could be used to process new data.

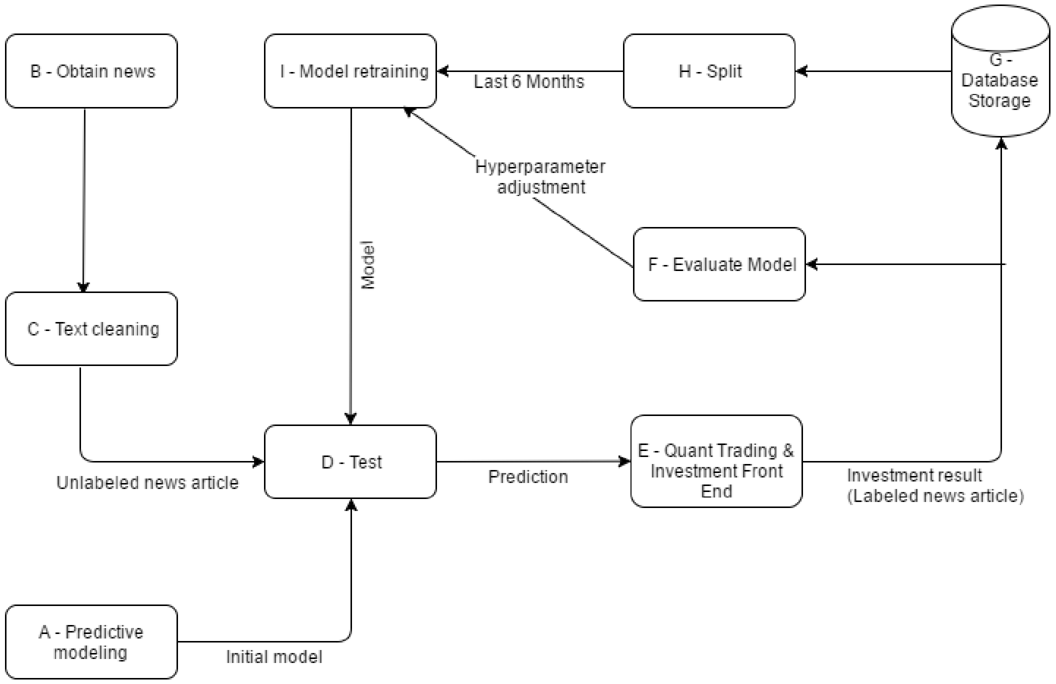


Figure 40 – The TMFP recommendation engine, deployed into a real investment scenario.

If compared with the process from Figure 25, Figure 40 shows the entire modelling process now enclosed in step A. The resulting model from A is used to predict the next price changes from the news articles arriving in step D. The role of step D is similar to the test container from Figure 25 (Feature Selection, Dimensionality Reduction, Predictive Test, and News Aggregation), the output of step D is the recommendation of SURGE or NOT RECOMMENDED, this will be consumed by step E. The output of step E is the news articles with the actual label value, after a period τ. The labelled news articles are then stored in a database, and the predictive model is evaluated in step F, and eventually the step F will trigger a new model retraining in I, if the hyperparameters changed. Periodically a new model is built in I, with the last six months of news articles (the oldest records removed, the newest records are included in the training set), then the process continues in D with a new model adjusted to the changes in the world.

Another approach is to apply the system above to online test, i.e., the predictive model could be constructed with recent historical data, and the classification and simulation could be performed on fresh new data. This approach can be the last and safe step to apply TMFP in a real investment scenario.

Chapter 5 – Experiments

This chapter will demonstrate and discuss the experiments with TMFP developed in this work. The experiment setup is explained, followed by the presentation of results from classification and investment simulation. Finally, a comparison with previous works existing in the literature, discussion, and proposal of good practices will be presented.

5.1 – Experiment Setup

A careful selection of algorithms, transformations, new techniques, and respective user parameters (also known as hyperparameters) was made to obtain a good predictive model. Also, 725 catalogued experiments were conducted to validate this process, resulting in the total of 10,947 predictive models created for all the 30 stocks listed in the DJIA index. In total, all the experiments required 18 months of CPU processing in a single computer with 2 cores, 2.2 GHz, and 8 GB of memory, and later using an Amazon EC2[10] Linux Ubuntu instance with 8 cores, 2.4 GHz, and 32 GB of memory. The time elapsed to conduct all the experiments, adjust the models and perform predictions could be reduced using the Hadoop environment and respective tools (White, 2009), but nowadays not all the pre-processing and machine learning algorithms were migrated to this platform. Once all the algorithms necessary to TMFP are well defined, they can be migrated using the big data extension developed for RapidMiner in (Beckmann, et al., 2014) in a future work.

The following machine learning algorithms were tested: Bagging (Breiman, 1996), Boost (Schapire, 2003), Classical Neural Networks, Convolutional Neural Networks (Collobert & Weston, 2008), Decision Stump (Iba & Langley, 1992), Decision Trees (Kohavi & Quinlan, 2002), Random Forests (Breiman, 2001), KNN (Fix & Hodges, 1951), Logistic Classifier (le Cessie, 1992), Naïve Bayes, Weighted Bayes (Rish, 2001), SVM (Cortes & Vapnik, 1995), SMO (Keerthi, et al., 2001), Vote (Kittler, et al., 1998),

[10] https://aws.amazon.com

just to cite the ones catalogued along the experiments. The next sections will demonstrate that all this effort led to satisfactory and robust predictive results.

As mentioned in Chapter 4, so far with the experiments conducted in this work, the best classification algorithm is the SVM, with the LIBSVM implementation (Chang & Lin, 2011). However, only the SVM classification algorithm was unable to provide good results without a good data preparation and new techniques proposed in this work.

To observe how the stock prices are affected by news articles, the experiments will demonstrate how the classifier can predict a SURGE or NOT RECOMMENDED, after a period τ=*1, 2, 3,* and *5* minutes after the news article has been released. Four experiments were executed, one for each time offset τ=*1, 2, 3,* and *5* minutes, and the classification and simulation results were compared and analysed. Experiments with values for $\tau<1$ and $\tau >5$ returned low performance and will not be presented.

The G-Mean and F-Measure were used to evaluate the classifier predictions, but in this work, the most important measure is the F-Measure, because it was used as fitness function in the optimization algorithms along the process. As explained in section 2.3.4, the F-Measure, like the Precision and Recall, assumes one class as positive. The arithmetic mean was used to calculate the F-Measure for both (positive and negative) classes to avoid value discrepancies, as the weighted average sometimes brings values higher than Precision and Recall, and the F-Measure is supposed to be a balanced measure between them. The G-Mean showed to be the a more reliable measure to filter the stocks predictions for investment simulation (*G-Mean >=55.00*).

The experiments will follow exactly the procedures explained in the methodology from Chapter 4, but the modelling process will be applied to 29 stocks listed on DJIA index instead of 30, because of the low quantity of the news articles gathered for CSCO (Cisco System Inc.), this was causing errors in the split step, and this stock was removed from the experiments, but it will be studied in a future work.

Despite the amount of experiments and models generated, this is far from being a complete and exhaustive search, and the proper combination of algorithms, transformations, new techniques, and hyperparameters could lead to more precise and robust results, especially with the perspective to apply this hyperparameter adjustment in a highly scalable computing environment that reduces the duration of experiments. However, the experiments executed so far demonstrated that the current solution is a robust and valuable alternative for investment recommendation, if compared with results published previously in the literature. These results will be demonstrated in the next sections.

5.2 – Classifier Performance

This section demonstrates the classification performance when predicting a SURGE or NOT RECOMMENDED movement in the stock prices, given a period of *τ=1, 2, 3,* and *5* minutes after the news article is released. These predictions refer to the news articles published between 03/Jun/2013 and 03/Sep/2013 (3 months of test dataset). To evaluate the classifier results, the G-Mean and F-Measure will be used to compare these experiments. The results are presented in Table 6, with the best value for each stock symbol (one symbol per row) marked in bold face. The values from Table 6 are an arithmetic mean after 10 runs, and the observed standard deviation represented between parenthesis. The Table 7, Figure 41, and Figure 42 show the descriptive statistics of each classification measure, extracted from the entire population of experiments after 10 runs.

The results marked in bold face in Table 6 showed a more frequent number of high values of G-Mean and F-Measure for *τ=1* and *τ=2*, despite some few high values from *τ=3* and *τ=5*. The descriptive statistics in Table 6 demonstrates high values of arithmetic mean, maximum values, and median for *τ=1*, while Figure 41 and Figure 42 demonstrate a rise in the statistics, when the values of *τ* approaches 1.

Table 6 - Classifier results in terms of G-Mean and F-Measure, when predicting a SURGE or NOT RECOMMENDED, for a time offset of τ=1, 2, 3, and 5 minutes after the news article be released.

	τ=5		*τ =3*		*τ=2*		*τ=1*	
Stock Symbol	**G-Mean**	**F-Measure**	**G-Mean**	**F-Measure**	**G-M ean**	**F-Measure**	**G-Mean**	**F-Measure**
AA	26.84 (0.4)	55.61 (0.2)	6.10 (0)	2.38 (0)	38.32 (1.2)	59.82 (0.5)	**72.12 (3.8)**	**75.31 (1.3)**
AXP	31.74 (0.3)	58.30 (0.2)	22.94 (0)	54.13 (0)	57.94 (1.3)	51.95 (0.4)	**62.87 (13.4)**	**59.29 (0.6)**
BA	25.16 (2.6)	54.59 (0.8)	33.28 (0)	58.26 (0)	54.95 (0.5)	63.03 (0.1)	**60.57 (1.0)**	**68.92 (0.3)**
BAC	22.12 (1.4)	53.27 (0.3)	15.07 (0)	51.51 (0)	**31.69 (0.2)**	**57.69 (0.1)**	26.85 (0.3)	55.64 (0)
CAT	11.92 (9.5)	11.35 (19.5)	25.82 (0)	55.58 (0)	31.90 (0.7)	57.30 (0.3)	**51.51 (4.4)**	**65.94 (0.8)**
CVX	0.00 (0)	8.66 (16.7)	**47.89 (0)**	**63.81 (0)**	31.96 (0.7)	59.00 (0.4)	46.29 (4.5)	59.62 (1.7)
DD	42.77 (0.7)	61.94 (0.7)	**69.61 (0)**	**68.09 (0)**	57.63 (0.6)	55.34 (2.7)	59.45 (7.7)	60.90 (5.5)
DIS	28.92 (0.2)	56.48 (0.1)	**41.81 (0)**	59.74 (0)	22.40 (8.6)	45.00 (21.8)	37.93 (6.3)	**61.52 (1.5)**
GE	30.72 (2.9)	18.93 (14.4)	23.22 (0)	50.86 (0)	29.04 (0.5)	54.99 (0.3)	**87.85 (4.6)**	**63.79 (0.2)**
HD	10.53 (16.2)	10.92 (19.3)	**44.65 (0)**	**64.44 (0)**	39.40 (0.7)	59.74 (0.8)	42.98 (4.0)	62.14 (1.7)
HPQ	21.94 (11.1)	53.14 (2.7)	**31.54 (0)**	**56.83 (0)**	25.02 (4.2)	55.46 (1.8)	27.88 (0.4)	56.47 (0.4)
IBM	33.03 (2.0)	57.10 (0.5)	30.83 (0)	57.47 (0)	29.80 (13.4)	48.40 (23.5)	**45.07 (0.8)**	**66.68 (0.5)**
INTC	22.22 (9.1)	51.71 (0.9)	22.32 (0)	53.50 (0)	10.82 (8.2)	11.44 (19.8)	**42.86 (17.5)**	**67.07 (7.0)**
JNJ	22.60 (5.4)	53.49 (1.2)	22.91 (0)	53.83 (0)	**46.00 (7.9)**	**66.32 (4.2)**	27.89 (0.4)	56.55 (0.3)
JPM	6.01 (4.2)	9.09 (17.5)	17.96 (0)	52.66 (0)	18.34 (17.4)	13.78 (23.5)	**42.36 (3.9)**	**60.32 (1.5)**
KO	19.10 (5.4)	46.41 (17.9)	8.05 (0)	2.60 (0)	31.96 (0.7)	59.01 (0.4)	**33.62 (0.7)**	**59.91 (0.4)**
MCD	7.71 (10.4)	10.26 (19.5)	43.07 (0)	60.42 (0)	**60.70 (3.7)**	65.86 (1.9)	59.47 (4.5)	**71.25 (0.3)**
MMM	4.76 (11.7)	10.86 (19.8)	0.00 (0)	3.81 (0)	37.26 (1.8)	48.27 (12.5)	**92.66 (7.8)**	**60.93 (6.3)**
MRK	35.57 (3.1)	60.24 (1.1)	68.98 (8.9)	67.26 (4.6)	77.04 (6.6)	**70.27 (0.1)**	**85.52 (0.2)**	65.65 (2.4)
MSFT	17.75 (1.1)	49.80 (0)	14.05 (0.7)	30.85 (23.1)	15.50 (7.8)	50.19 (0.2)	**34.99 (14.3)**	**62.19 (5.0)**
PFE	44.84 (1.4)	62.64 (0.6)	**60.50 (0.1)**	63.25 (0.5)	41.15 (0.7)	62.91 (0)	41.07 (3.9)	**64.12 (2.2)**
PG	24.24 (0.3)	53.26 (0.4)	35.24 (10.1)	51.91 (2.3)	29.10 (0.5)	56.79 (0.2)	**38.21 (1.0)**	**61.74 (1.2)**
T	32.44 (4.9)	56.07 (1.2)	22.92 (0)	54.04 (0.1)	30.42 (0.6)	57.01 (0.9)	**38.21 (1.1)**	**61.41 (1.1)**
TRV	38.08 (15.5)	58.83 (3.8)	28.45 (14.3)	38.04 (29.7)	35.78 (17.9)	62.91 (6.7)	**59.59 (4.5)**	**76.00 (3.0)**
UNH	34.99 (14.3)	61.11 (5.0)	42.73 (13.5)	35.27 (20.3)	**51.55 (3.1)**	70.36 (2.1)	49.49 (20.2)	**70.96 (8.7)**
UTX	35.27 (0.2)	**57.21 (1.2)**	0.00 (0)	49.17 (0)	11.36 (8.8)	13.54 (21.8)	**54.14 (2.8)**	52.35 (8.3)
VZ	30.41 (9.9)	48.82 (19.0)	32.34 (0.1)	56.77 (1.8)	**61.54 (16.9)**	**63.04 (3.7)**	8.08 (9.6)	8.82 (19.4)
WMT	18.20 (2.6)	52.32 (0.7)	21.47 (3.7)	53.43 (1.0)	39.38 (2.6)	**59.24 (0.5)**	**41.05 (14.7)**	55.03 (22.1)
XOM	26.85 (9.6)	54.21 (1.4)	34.87 (2.4)	58.15 (0.3)	51.47 (4.2)	**59.90 (0.6)**	**69.75 (11.1)**	59.80 (2.2)

Table 7 - Descriptive statistics for all stock symbols and time offsets.

Statistical Measure	*τ=5*		*τ=3*		*τ=2*		*τ=1*	
	G-Mean	**F-Measure**	**G-Mean**	**F-Measure**	**G-Mean**	**F-Measure**	**G-Mean**	**F-Measure**
Arithmetic Mean	24.37	44.71	29.95	49.24	38.07	53.71	**49.67**	**61.06**
Std. Deviation	**13.50**	21.81	18.15	19.54	17.39	18.04	20.70	**13.08**
Max Value	44.83	62.64	69.61	68.09	77.04	70.27	**92.66**	**76.00**
Median	25.92	53.72	25.82	54.13	36.48	58.81	**47.07**	**62.85**
Mode	34.00	52.00	22.00	54.00	31.00	57.00	**57.00**	**58.00**

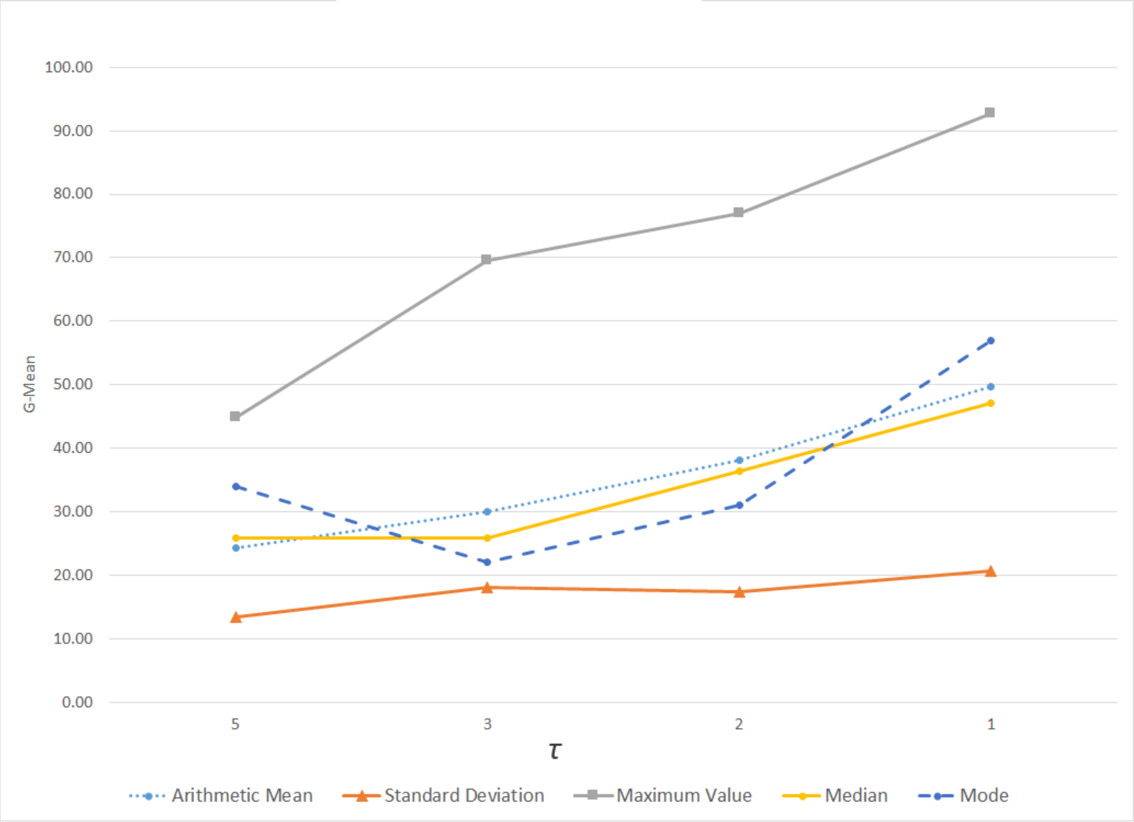


Figure 41 – G-Mean statistics for all stock symbols and time offsets of *τ=1, 2, 3,* and *5*.

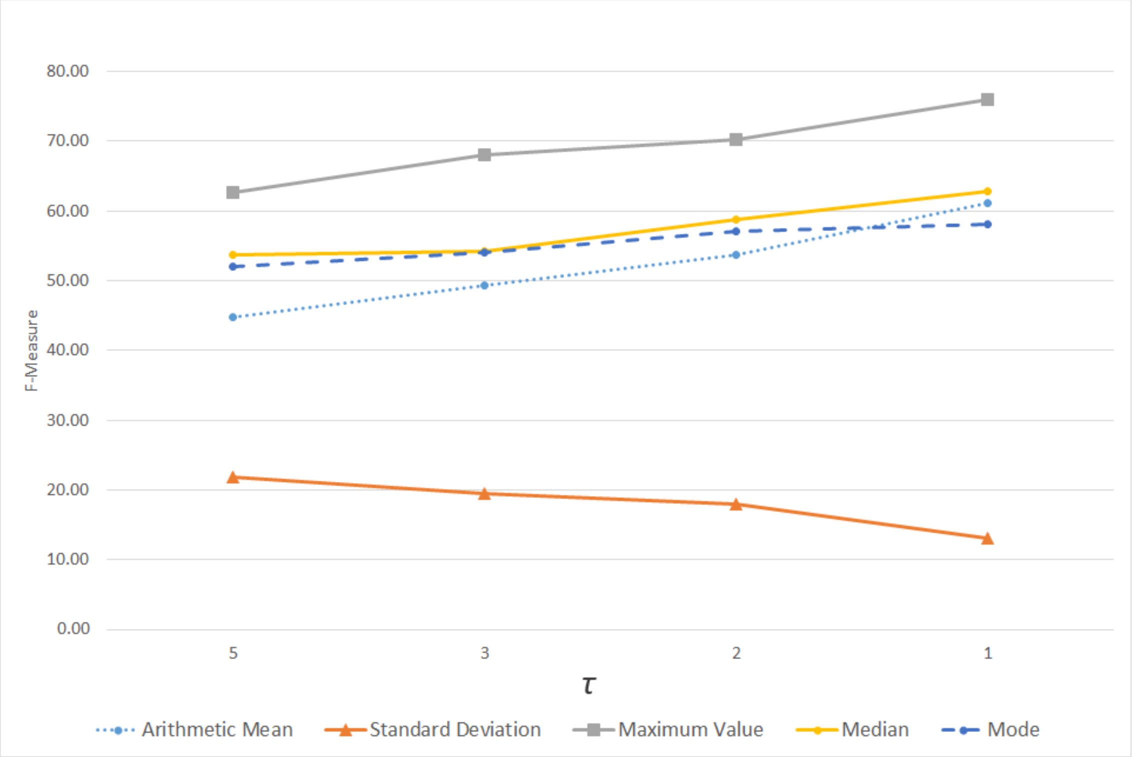


Figure 42 - F-Measure statistics for all stock symbols and time offsets of *τ=1, 2, 3,* and *5*.

Another approach to demonstrate these results is to filter the stock predictive models given a performance threshold. During the experiments with investment simulation, it was observed that the stock predictive models selected using *G-Mean>=55.00* returned high investment results, generating more gains and less losses if compared with F-Measure or AUC as a filtering criteria. This can be explained by the fact that the G-Mean aims to obtain a balance of true predictions, using a geometric mean of true positive and true negative hits.

Table 8 - Classifier results in terms of *G-Mean>=55.00* and F-Measure when predicting a SURGE or NOT RECOMMENDED for a time offset of *τ=1, 2,* and 3 minutes after the news article is released.

Stock Symbol	*τ=3*		*τ=2*		*τ=1*	
	G-Mean	F-Measure	G-Mean	F-Measure	G-Mean	F-Measure
AA					**72.12 (3.8)**	**75.31 (1.3)**
AXP			57.94 (1.3)	51.95 (0.4)	**62.87 (13.4)**	**59.29 (0.6)**
BA			54.95 (0.5)	63.03 (0.1)	**60.57 (1.0)**	**68.92 (0.3)**
DD	**69.61 (0.0)**	**68.09 (0.0)**	57.63 (0.6)	55.34 (2.7)	59.45 (7.7)	60.90 (5.5)
GE					**87.85 (4.6)**	**63.79 (0.2)**
MCD			**60.70 (3.7)**	65.86 (1.9)	59.47 (4.5)	**71.25 (0.3)**
MMM					**92.66 (7.8)**	**60.93 (6.3)**
MRK	68.98 (8.9)	67.26 (4.6)	77.04 (6.6)	**70.27 (0.1)**	**85.52 (0.2)**	65.65 (2.4)
PFE	**60.50 (0.1)**	**63.25 (0.5)**				
TRV					**59.59 (4.5)**	**76.00 (3.0)**
VZ			**61.54 (16.9)**	**63.04 (3.7)**		
XOM					**69.75 (11.1)**	**59.80 (2.2)**

Table 9 – Descriptive statistics for *G-Mean>=55.00*.

Statistical Measure	*τ=3*		*τ=2*		*τ=1*	
	G-Mean	F-Measure	G-Mean	F-Measure	G-Mean	F-Measure
Arithmetic Mean	66.36	**66.20**	62.00	61.57	**70.99**	66.18
Standard Deviation	**6.76**	**3.49**	9.70	6.56	14.37	6.78
Maximum Value	69.61	68.09	77.04	70.27	**92.66**	**76.00**
Median	61.29	**68.09**	58.59	63.12	**70.56**	65.89
Mode	69.00	68.00	58.00	64.00	**85.00**	**74.00**

Table 8 and Table 9 present the stocks with *G-Mean >= 55.00* and their respective F-Measure. The predictive models with *τ=5* were not included because they did not produce any results with *G-Mean >= 55.00*. It can be noticed that the time offset *τ=1* produced more predictive models (10 stocks in total), and the highest values of G-Mean and F-Measure than other time offset configurations. Also, Figure 43 shows a rise of measurements from *τ=3* to *τ=1*. The predictive models with *τ=3* had no variance.

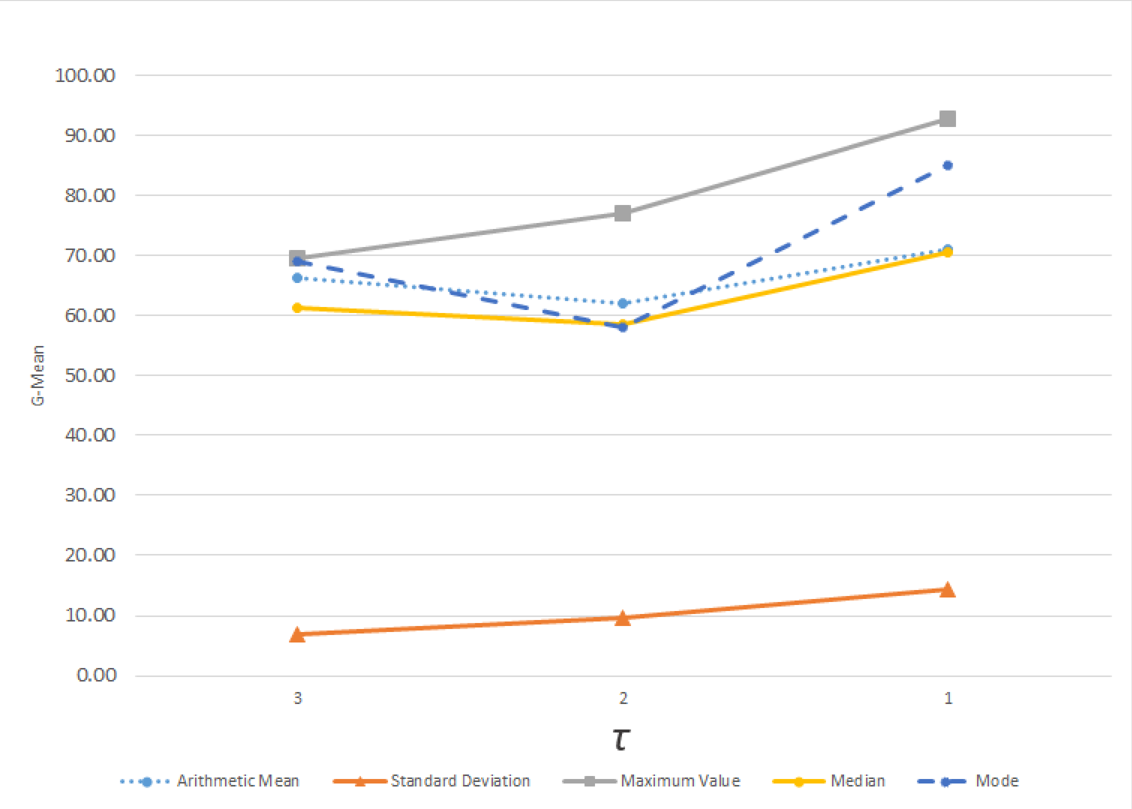


Figure 43 – Descriptive statistics for *G-Mean* >= *55.00*, with the time offsets *τ=1, 2, 3*.

These results demonstrate evidences that the stock prices studied in this work started to be affected by the news articles few minutes after they are published, and that a loss of signal occurs when the news articles are accumulated in a wider period, because there is no mechanism developed to distinguish which news articles are affecting the stock price, making it more difficult to obtain a stable model under these conditions. As a future work, an investigation about the alignment of news articles and stock prices in a wider period will be conducted.

To make a comparative analysis of the results published in the previous literature, the best experiment in this section (*τ=1*) will have its evaluation measures: Accuracy, Precision, Recall, AUC, G-Mean, and F-Measure exposed in the discussion from section 5.4.

In the next section, all the predictive models listed in Table 8 will be submitted to the investment simulation process, to test their significance and the applicability of these models in a real investment scenario.

5.3 – Investment Simulation

In this section, the predictions made by classifier predictions with *G-Mean >=55.00* were applied in an investment simulation engine. These predictions refer to the news articles published between 03/Jun/2013 and 03/Sep/2013 (3 months of test dataset).

As explained in the methodology chapter, the simulation engine uses a very short term investment strategy: if a SURGE prediction occurs, purchase $10,000 of shares from the related stock at *τ-1* minutes after the news article being published (i.e., for *τ =1*, 0 minute, or as soon as possible, for *τ=2*, one minute after the news article being published, etc.). Hold the stock position for *n=3* minutes, if during that *n* minutes the stock can be sold to make a profit of >=2%, then sell it immediately. At the end of three minutes, the stock is sold at the current market price, taking a loss if necessary.

The results of investment simulation using the predictive models for different values of time offset are demonstrated along the tables and charts in this section. These values represent the arithmetic mean of investment simulation, after 10 runs for TradeMiner, 100 runs for Random Trader, and the standard deviation represented between parenthesis. Each run comprises an entire process of investment simulation using the predictions made between 03/Jun/2013 and 03/Sep/2013. All values marked with $ represent US$. An operation, represented in the column "number of operations", means a SURGE prediction that triggered an entire process to buy and sell a stock share by the investment simulator. The cumulative return is the profit (or loss) of an investment over n periods (27), and is represented as a percentage (%).

The values in Table 10, Table 11, and Table 12 represent the arithmetic mean of investment simulation for each stock symbol, after 10 runs, using the predictions generated from TradeMiner, for *τ=1, 2,* and *3* respectively. The bottom lines represent the sum of all columns, and it can be noticed that all three values for *τ* generated some positive cumulative return, with a rise from *τ=3* to *τ=1*, with no more than three stocks presenting loss in all simulations. Specifically for *τ=1*, an expressive cumulative return (if compared with 0.05% from US T-Bond in the same period) of 21.47% was observed. It also can be observed in some stocks, especially for *τ=1*, some large standard deviation in terms of loss and gain, which can be explained by a high standard deviation for G-Mean. In some cases, there is about 40% of loss even with G-Mean above 75.00, this is because the investment strategy counts with any heuristic to leave (sell) the position. Despite of this, in most stocks the strategy simulation showed to be profitable using TradeMiner recommendations.

Table 10 – Average results of three months of investment simulation after 10 runs, by stock symbol, using TradeMiner predictions for *τ=1*.

Stock Symbol	Number of operations	G-Mean	Loss ($)	Gain ($)	Profit ($)	Cumulative Return (%)
AA	4 (0)	72.12 (3.8)	-13.05 (0)	155.45 (0)	142.4 (0)	1.42 (0)
AXP	35 (12)	62.87 (13.4)	-642.43 (227.13)	900.00 (303.03)	257.56 (75.90)	2.57 (0.75)
BA	10 (0)	60.57 (1.0)	-91.63 (0)	170.68 (0)	79.05 (0)	0.79 (0)
DD	19 (2)	59.45 (7.7)	-198.81 (2.20)	612.20 (100.48)	413.38 (102.68)	4.13 (1.02)
GE	17 (0)	87.85 (4.6)	-143.30 (5.17)	325.83 (2.63)	182.53 (2.54)	1.82 (0.02)
MCD	5 (1)	59.47 (4.5)	-79.60 (2.58)	57.75 (6.65)	-21.85 (9.23)	-0.21 (0.09)
MMM	31 (16)	92.66 (7.8)	-214.45 (75.92)	534.78 (100.93)	320.33 (25.00)	3.20 (0.25)
MRK	48 (8)	85.52 (0.2)	-206.02 (52.12)	1198.44 (102.31)	992.41 (50.18)	9.92 (0.49)
TRV	1 (0)	59.59 (4.5)	0.00 (0)	19.67 (16.14)	19.67 (16.14)	0.19 (0.16)
XOM	49 (16)	69.75 (11.1)	-740.69 (261.87)	502.38 (167.97)	-238.31 (93.90)	-2.38 (0.93)
TOTAL	221 (4)	-	-2330.01 (356.16)	4477.20 (342.81)	2147.18 (13.34)	21.47 (0.13)

Table 11 – Average results of three months of investment simulation after 10 runs, by stock symbol, using the TradeMiner predictions for *τ=2*.

Stock Symbol	Number of operations	G-Mean	Loss ($)	Gain ($)	Profit ($)	Cumulative Return (%)
AXP	78 (0)	57.94 (1.3)	-440.41 (0)	642.01 (0)	201.60 (0)	2.01 (0)
BA	28 (0)	54.95 (0.5)	-265.76 (0)	300.51 (0)	34.75 (0)	0.34 (0)
DD	46 (9)	57.63 (0.6)	-306.91 (7.05)	362.70 (3.10)	55.79 (7.38)	0.55 (0.07)
MCD	21 (5)	60.70 (3.7)	-151.33 (38.10)	190.58 (40.82)	39.25 (2.72)	0.39 (0.02)
MRK	27 (5)	77.04 (6.6)	-112.43 (18.07)	426.79 (84.60)	314.36 (66.52)	3.14 (0.66)
VZ	28 (8)	61.54 (16.9)	-408.63 (115.55)	292.16 (103.29)	-116.47 (12.25)	-1.16 (0.12)
TOTAL	228 (17)	-	-1685.46 (206.08)	2214.74 (56.40)	529.27 (149.67)	5.29 (1.49)

Table 12 – Average results of three months of investment simulation after 10 runs, by stock symbol, using the predictions for *τ=3*.

Stock Symbol	Number of operations	G-Mean	Loss ($)	Gain ($)	Profit ($)	Cumulative Return (%)
DD	16 (0)	69.61 (0.0)	-164.97 (0)	246.84 (0)	81.87 (0)	0.81 (0)
MRK	37 (29)	68.98 (8.9)	-210.23 (207.05)	387.70 (243.39)	177.47 (36.33)	1.77 (0.36)
PFE	33 (1)	60.50 (0.1)	-262.67 (37.07)	464.35 (38.02)	201.68 (75.10)	2.01 (0.75)
TOTAL	87 (30)	-	-637.86 (244.13)	1098.88 (205.36)	461.01 (38.76)	4.61 (0.38)

In order to test the significance of predictive models, a random trader was used for investment simulation as a null hypothesis (Neyman & Pearson, 1933), to be compared with the simulation using the recommendation from TradeMiner as an alternative hypothesis. Table 13 demonstrates the arithmetic mean of totals of investment simulation after 10 runs using TradeMiner recommendations, and after 100 runs using the random trader recommendations. From there, it can be noticed that the cumulative returns from TradeMiner are higher than the cumulative return from a random trader. The low results from the random trader presented loss for *τ=3*, and in all cases, the standard deviation is above the mean value, which denotes a high uncertainty when using a random trader for this type of problem. Figure 44 demonstrates the average cumulative return by time offsets, with very low and discrepant results from the random trader, and a rise of positive results from *τ=3* to *τ=1* when using TradeMiner recommendations.

Table 13 – Average of three months of investment simulation using the predictions from TradeMiner and Random Trader, after 10 and 100 runs respectively.

Time offset τ	Source of predictions	Number of operations	Loss ($)	Gain ($)	Profit ($)	Cumulative Return (%)
1	TradeMiner	221 (4)	-2330.01 (356.16)	4477.20 (342.81)	2147.18 (13.34)	21.47 (0.13)
1	Random	244 (15)	-1171.52 (191.22)	1188.65 (197.78)	17.12 (271.04)	0.17 (2.71)
2	TradeMiner	228 (17)	-1685.46 (206.08)	2214.74 (56.40)	529.27 (149.67)	5.29 (1.49)
2	Random	232 (11)	-961.02 (135.69)	1047.94 (138.11)	86.92 (203.26)	0.86 (2.03)
3	TradeMiner	87 (30)	-637.86 (244.13)	1098.88 (205.36)	461.01 (38.76)	4.61 (0.38)
3	Random	78 (8)	-360.02 (77.50)	337.26 (83.30)	-22.75 (111.89)	-0.22 (1.11)

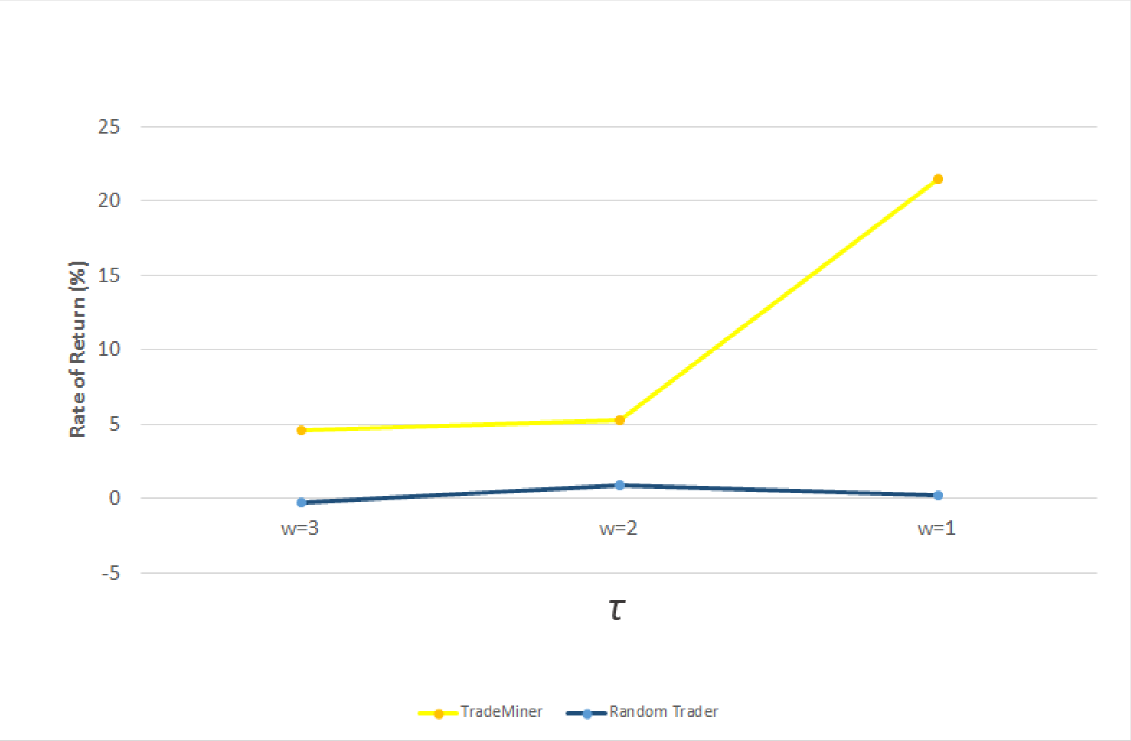


Figure 44 - Summary of average of cumulative return by time offset.

Table 14 summarises all the cumulative returns acquired along the simulations, and respective p-values calculated with one sample t-test. As explained in Chapter 4, section 4.5.3, a cut point of α=2.821 is established (99% of confidence, nine degrees of freedom, one tail). As all the *p-values*>α, the null hypothesis H_0 is rejected (i.e., to use a random trader as a recommendation engine for investment), and the alternative hypothesis H_a is accepted (i.e., to use the TradeMiner as a recommendation engine for investment).

Table 14 - Summary of average cumulative return (%) by time offset and respective p-values.

Origin of Predictions	***τ=3***	***τ=2***	***τ=1***
TradeMiner	4.61 (0.38)	5.29 (1.49)	21.47 (0.13)
Random	-0.22 (1.11)	0.86 (2.03)	0.17 (2.71)
p-value (one sample t-test)	39.46	9.34	504.93

These results show a huge cumulative return if compared with the results from a random trader, showing evidences that the predictive models are stable and profitable, especially for investment decisions predicted one minute after the news articles be released *(τ=1)*. It also worth to mention, due to the very short term investment strategy, that the capital of $10,000 is kept invested for no more than three minutes for each operation, it means for example from Table 10, for *τ=1* and stock AA, the value of $10,000 was hold for only *4 operations . 3 = 12* minutes, with a cumulative return of 1.42%. According to the information in Table 10, the stocks were kept invested for 221 . 3 / 60 ≅ 11 hours, with a cumulative average return of 21.47% after three months. Nevertheless, as mentioned in section 4.5.2, the investment strategy relies in 2 assumptions: the transactions have zero cost, and stocks shares will be available to buy and sell at the moment they are requested. In a real investment scenario, the transactions have a cost, and the capacity to execute an investment order varies greatly, then the cumulative return can be limited to these factors that are beyond the scope of this work.

Since the results of classification and investment simulations are properly demonstrated, the next section will compare and discuss these results with the values published in the related literature.

5.4 – Discussion

In this section, the results demonstrated previously will be compared with the state of the art, followed by a discussion about some good practices and future improvements in this area.

As this chapter demonstrated, the use of proper evaluation measures was crucial to the success of this work. The arithmetic mean of F-Measure (18) was used to adjust the hyperparameters along the modelling process, and G-Mean (19) was used to filter the predictive models for investment simulation, given a threshold above or equal 55.00. According to the initial experiments conducted in this work, it was not possible to achieve good results, using only Accuracy as a measure for algorithm adjustment. Nevertheless, to make a comparison with the results published in the literature, the best experiment from section 5.2 (*τ=1*), will have other evaluation measures (Accuracy, Precision, Recall, AUC) exposed together with G-Mean, and F-Measure along the Table 15.

All values in Table 15 are an arithmetic mean after 10 runs, and the observed standard deviation is represented between parenthesis. Like the F-Measure, the Precision and Recall are represented as an arithmetic mean of positive and negative classes (The results for both positive and negative classes can be verified in Appendix A, Table 18). The maximum values by classification measure are marked in bold face.

Along the Table 15, 20 values of Accuracy above 98.00 can be noticed, and if compared with other measures in the same line, the Accuracy always has the highest values, denoting the most optimist results with no much room for improvement, but if compared with G-Mean and F-Measure, some huge discrepancies among these measurements (e.g., BAC, DIS, JNJ, WMT) can be noticed. According to (Weiss & Provost, 2001), (Ling, et al., 2003), (Weis, 2004), (He & Garcia, 2009) , (He. & Ma, 2013), (Ali, et al., 2013), this is explained because the Accuracy measure lacks the sensitivity to data distributions, and assumes equal costs for positive and negative errors. A good example is a data set consisting of 98 negative examples (the majority class), and two positive examples (the minority class), then a classifier that identifies all data as negative will achieve 98% of Accuracy. Assuming that the minority class represents a rare disease to be detected, this classifier is useless for application in a real-world scenario. Especially in this branch of research, the best moment to invest is a rare and valuable event, then the underlying data for this problem is naturally imbalanced, and Accuracy as a criterion to evaluate the predictive performance must be avoided in all circumstances. In an opposite way, the G-Mean showed the lowest values, followed by the F-Measure. This pessimist behaviour was useful along the process of building 30 predictive models simultaneously, because it denounced and helped to identify problems occurring in the data, algorithms, and hyperparameters.

Table 16 shows the evolution of classification measures and simulation results in chronological order, since the first reported initiative with TMFP (Wuthrich, et al., 1998), until the results from the current work at the bottom line. These results represent the best values published in each work. The CR in the column “Max Simulation Results” means Cumulative Return (27). The entries marked with N/A represent findings not applicable for this comparison, but these respective works brought important insights to this branch of knowledge when analysing the effects of news regarding sentiments and companies’ fundaments.

Still in Table 16, about 50% of the reviewed works published their results only as Accuracy. Despite all the warnings and evidences along all these years, some results in this branch of knowledge continue to be published using Accuracy only, even in the recent years, and just a few of them devoted attention to the class imbalance problem. Another concern is regarding the inappropriate use of n-fold cross validation in some works. Unless the purpose to use cross validation is to adjust a model using the training dataset, and keeping the test dataset untouched, it is not acceptable to use cross validation for prediction in a time series, because information from the future is used to build a model to identify events occurred in the past, giving an unfair advantage to the classifier (Hastie, et al., 2003), with the resulting model becoming useless for application in the real world. However, if the decision is to use cross-validation, some measures must be taken when dealing with time series, as proposed by (Arlot & Celisse, 2010), (Bergmeir, et al., 2015).

Table 15 - More classification measures for $\tau=1$.

Stock Symbol	Accuracy	Precision	Recall	AUC	G-Mean	F-Measure
AA	99.20 (0.1)	74.81 (0.0)	75.98 (2.9)	**67.87 (6.0)**	72.12 (3.8)	75.31 (1.3)
AXP	93.17 (2.0)	62.90 (14.8)	69.22 (6.0)	67.27 (5.1)	62.87 (13.4)	59.29 (0.6)
BA	98.94 (0.0)	69.72 (0.0)	68.19 (0.6)	66.60 (4.0)	60.57 (1.0)	68.92 (0.3)
BAC	99.05 (0.0)	65.66 (1.4)	53.54 (0.1)	63.76 (1.4)	26.85 (0.3)	55.64 (0.0)
CAT	98.49 (0.2)	73.78 (10.5)	63.17 (2.0)	63.67 (1.3)	51.51 (4.4)	65.94 (0.8)
CVX	97.70 (0.1)	59.04 (1.2)	60.32 (2.3)	63.22 (0.8)	46.29 (4.5)	59.62 (1.7)
DD	94.08 (1.3)	58.58 (4.8)	66.51 (5.6)	63.41 (0.7)	59.45 (7.7)	60.90 (5.5)
DIS	99.23 (0.1)	94.88 (11.6)	57.37 (2.7)	63.00 (0.4)	37.93 (6.3)	61.52 (1.5)
GE	98.43 (0.0)	58.64 (0.3)	88.54 (4.3)	63.71 (0.8)	87.85 (4.6)	63.79 (0.2)
HD	98.10 (0.1)	68.82 (1.2)	59.11 (1.5)	63.38 (0.6)	42.98 (4.0)	62.14 (1.7)
HPQ	98.64 (0.1)	77.94 (8.8)	53.84 (0.1)	62.60 (0.1)	27.88 (0.4)	56.47 (0.4)
IBM	99.20 (0.0)	99.60 (0.0)	60.16 (0.4)	62.46 (0.1)	45.07 (0.8)	66.68 (0.5)
INTC	99.70 (0.0)	99.85 (0.0)	60.71 (4.4)	62.47 (0.1)	42.86 (17.5)	67.07 (7.0)
JNJ	98.16 (0.1)	93.75 (13.1)	53.86 (0.0)	61.89 (0.3)	27.89 (0.4)	56.55 (0.3)
JPM	99.13 (0.4)	64.73 (4.1)	58.90 (1.6)	61.75 (0.4)	42.36 (3.9)	60.32 (1.5)
KO	99.01 (0.0)	99.51 (0.0)	55.66 (0.2)	61.52 (0.5)	33.62 (0.7)	59.91 (0.4)
MCD	98.84 (0.1)	78.17 (3.5)	67.67 (2.8)	61.74 (0.4)	59.47 (4.5)	71.25 (0.3)
MMM	89.18 (6.9)	58.18 (2.7)	**92.74 (7.8)**	62.42 (0.1)	**92.66 (7.8)**	60.93 (6.3)
MRK	91.76 (1.8)	61.26 (1.5)	85.78 (0.1)	63.78 (0.5)	85.52 (0.2)	65.65 (2.4)
MSFT	**99.77 (0.0)**	**99.88 (0.0)**	57.14 (2.9)	63.69 (0.5)	34.99 (14.3)	62.19 (5.0)
PFE	98.57 (0.0)	99.28 (0.0)	58.51 (1.4)	63.47 (0.4)	41.07 (3.9)	64.12 (2.2)
PG	98.92 (0.2)	93.79 (14.0)	57.26 (0.3)	63.31 (0.3)	38.21 (1.0)	61.74 (1.2)
T	99.41 (0.1)	78.31 (8.8)	57.28 (0.4)	63.18 (0.3)	38.21 (1.1)	61.41 (1.1)
TRV	99.22 (0.1)	99.61 (0.1)	67.86 (2.9)	63.22 (0.3)	59.59 (4.5)	**76.00 (3.0)**
UNH	98.15 (0.0)	99.07 (0.0)	64.29 (5.8)	63.23 (0.3)	49.49 (20.2)	70.96 (8.7)
UTX	85.41 (5.3)	55.26 (9.9)	60.16 (3.4)	63.21 (0.3)	54.14 (2.8)	52.35 (8.3)
VZ	14.89 (34.3)	27.16 (13.2)	35.22 (8.0)	62.15 (0.3)	8.08 (9.6)	8.82 (19.4)
WMT	85.06 (34.3)	65.13 (11.1)	58.53 (6.0)	62.04 (0.2)	41.05 (14.7)	55.03 (22.1)
XOM	94.77 (1.7)	62.27 (15.2)	73.97 (6.1)	62.46 (0.2)	69.75 (11.1)	59.80 (2.2)

Table 16 – Maximum results published in the related literature.

Reference	**Max Performance**	**Max Simulation Results**	**Evaluation Period (Test Set)**
(Wuthrich, et al. 1998)	Accuracy 46.70	CR 7.5%	3 months
(Lavrenko, et al. 2000)	-	CR 0.23%	40 days
(Peramunetilleke & Wong, 2002)	Accuracy 53.00	-	
(Fung, et al., 2003)	-	CR 6.55%	1 month
(Gidófalvi & Elkan, 2003)	Accuracy 45.00	-	-
(Mittermayer, 2004)	Weighted Recall 60.00	Average return 11%	-
(Werner & Murray, 2004)	-	-	-
(Das & Chen, 2007)	N/A	N/A	-
(Rachlin, et al., 2007)	Accuracy 82.40	Return over investment $23,341	3 months
(Soni, et al., 2007)	Accuracy 56.20	-	-
(Zhai, et al., 2007)	Accuracy 70.01	CR 5.1%	2 months
(Mahajan, et al., 2008)	Accuracy 60.00	-	-
(Tetlock, et al., 2008)	N/A	N/A	-
(Butler & Kešelj, 2009)	Precision 67.80	-	-
(Schumaker & Chen, 2009)	Accuracy 57.10	CR 2.06%	-
(Huang, et al., 2010)	Avg Precision 85.26 Avg Recall 75.37	-	-
(Li, 2010)	Accuracy 67.00	-	-
(Bollen & Huina, 2011)	Directional Accuracy 87.60	P-values < 0.05	19 days
(Groth & Muntermann, 2011)	Accuracy 75.00, **AUC 70.30**	Avg return 12.42% (14.44)	Cross Validation
(De Faria, et al., 2012)	Accuracy 66.17 Precision 66.57 Recall 65.37 F-Measure 65.17	Avg return 33%	337 days
(Hagenau, et al., 2012)	Accuracy 76.00	Avg return per trade 1.1%	-
(Lugmayr & Gossen, 2012)	-	-	-
(Schumaker, et al., 2012)	Accuracy 59.00	CR 3.3%	-
(Siering, 2012)	Accuracy 68.27, Avg Precision 68.45, Avg Recall 64.48, Avg F-Measure 64.40	Return of 0.0585% (0.0028)	Cross validation
(Vu, et al., 2012)	Accuracy 82.93	-	-
(Jin, et al., 2013)	Precision ~ 28.00	-	-
(Makrehchi, et al., 2013)	-	20% over S&P500	4 months, Cross Validation
(Yu, et al., 2013)	N/A	N/A	-

(Crone & Koeppel, 2014)	Accuracy 60.26	-	Cross validation
(Kim, et al., 2014)	F-Measure 65.20	-	-
(Vakeel & Shubhamoy, 2014)	Weighted average: Precision 65.300, Recall 64.00, F-Measure 63.10, AUC 63.80	-	-
(Wong, et al., 2014)	Accuracy 55.70	CR 56%, Sharpe ratio: 0.148	1 year
(Nassirtoussi, et al., 2015)	Accuracy 83.33	-	-
(Yang, et al., 2015)	-	Annualized average return 23.18%, Sharpe ratio 2.92	3 months
(Fehrer & Feuerriegel, 2016)	F-Measure 56.00	-	-
This current work	**Max average value:** **Accuracy 99.77 (0.0),** **Precision 99.88 (0.0),** **Recall 92.74 (7.8),** **AUC 67.87 (6.0),** **G-Mean 92.66 (7.8),** **F-Measure 76.00 (3.0)**	**Avg CR 21.47% (0.13),** **p-value<0.01**	**3 months**

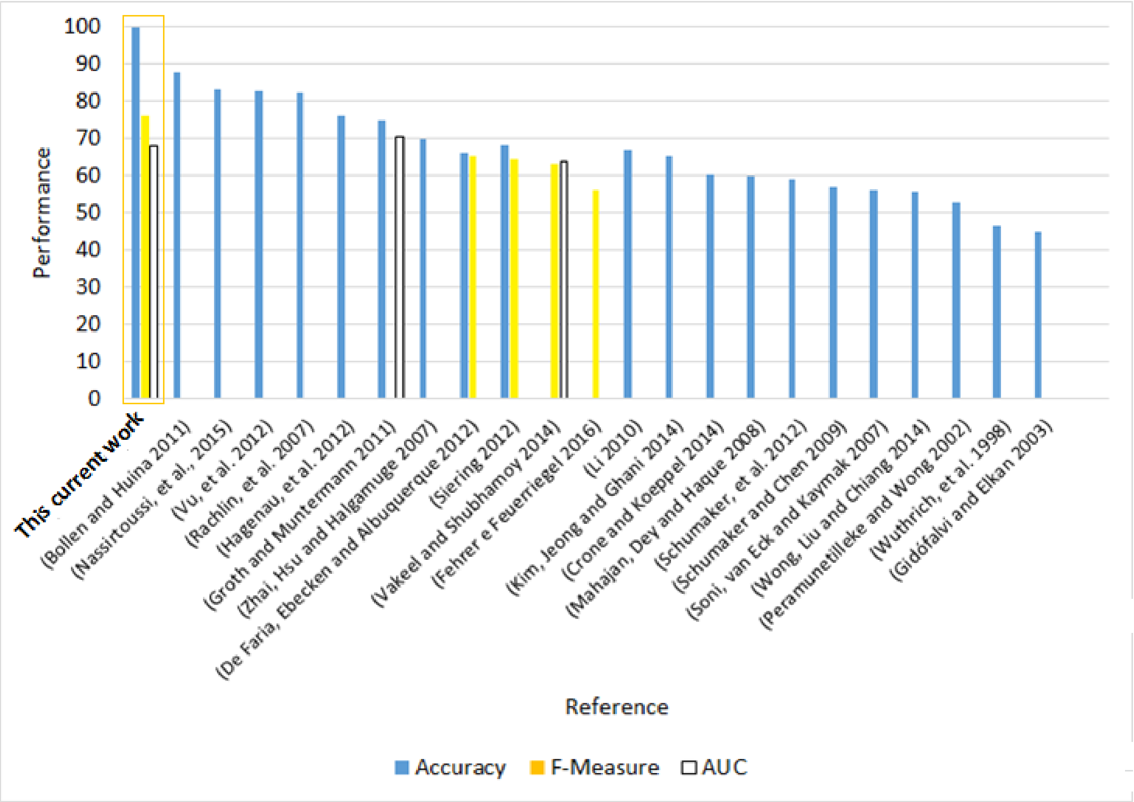


Figure 45 - Results of Accuracy, F-Measure, and AUC by reviewed work.

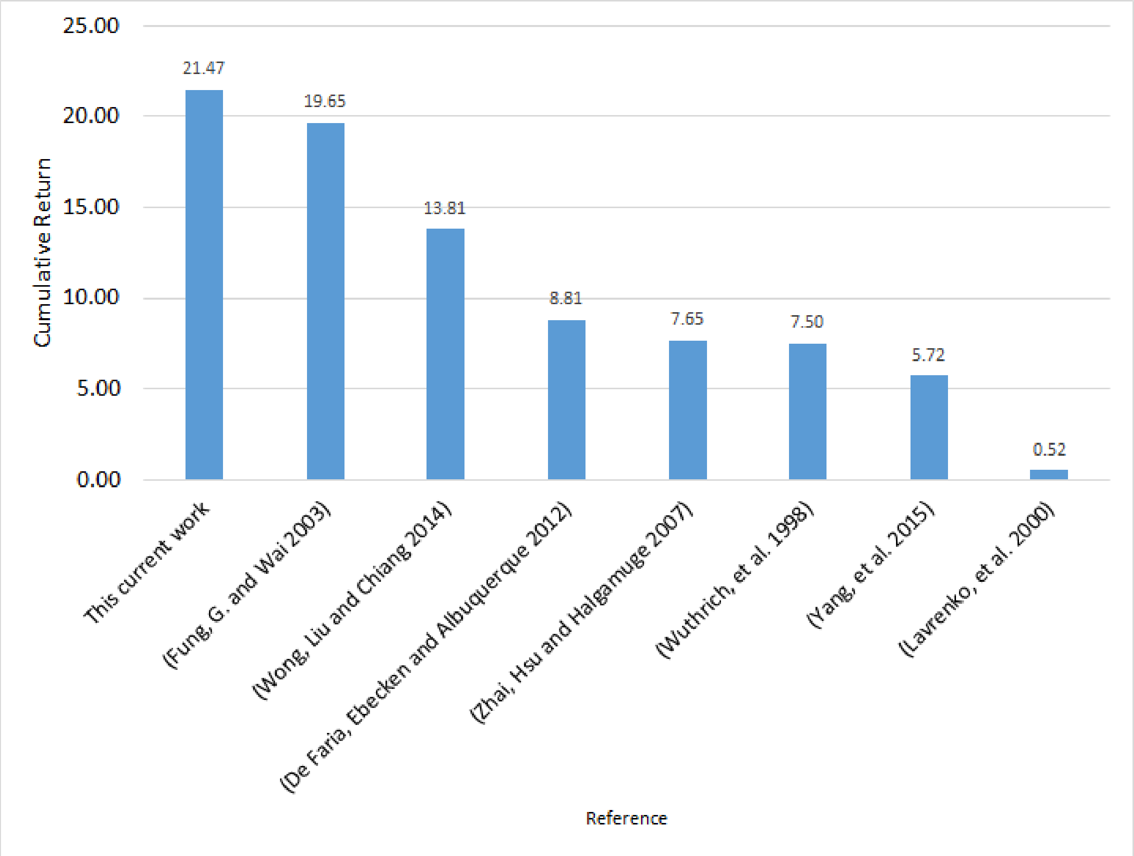


Figure 46 – Comparison of cumulative returns of investment simulation, adjusted for three months, by reviewed work. The reviewed works with no information about the evaluation period were not included in this comparison.

If compared to the maximum results along the Table 16 and Figure 45, with exception of (Groth & Muntermann, 2011) with AUC of 70.30 obtained with cross validation, the current work presents the highest values published. Despite the high values achieved in terms of classification measures in this current work, so far there is no dataset established as a benchmark for TMFP problem, and it is not possible to make a precise comparison to other works, because each of them aims to solve problems in different conditions such as periods of time, markets, exchanges, indexes, datasets, etc. However, considering only the profit (as this is one of the purposes of financial markets), according to Figure 46 the current work presented the best result so far, with a cumulative return of 21.47% after three months of simulation, and p-value<0.01, demonstrating that text mining is applicable as a predictive tool for financial markets.

At last, as a contribution to the state of the art, in addition to the methodology proposed in this work, here are some simple recommendations for researchers in this branch of knowledge: avoid to publish results using only Accuracy, use cross validation with caution as a model selection, and make a good choice of your classification measures. These good practices will help the algorithms to respond and improve, and at the same time you will be fair with your sponsors and audience, and decisively contributing to improve the transparency, predictability, and clear understanding of financial markets.

Chapter 6 - Conclusion

The purpose of this work is to identify possible relationships between textual information and the stock price movements, and it presents a computational framework using data mining and text mining to find patterns between the news articles published and the respective movements in the stock prices, creating a predictive model to forecast the stock prices changes along the day (intraday), for the 30 companies listed in the DJIA. This computational framework can be considered a recommendation system to be used by a high frequency quantitative trading system.

Due to the complex and unstable nature of the financial markets, the machine learning algorithm alone was not capable to make correct predictions, due to the existence of imbalanced data and class overlapping. To solve this problem, this work proposes a new data preparation technique to deal with the imbalanced class problem named KNN-Und, and a classifier ensemble technique using a genetic algorithm named CATS, which is adapted to remove the class overlapping in time series. All the new algorithms proposed in this work were developed in RapidMiner, in an extension called TradeMiner.

The best experiment used a time offset of one minute after the news article being published, and the maximum results in terms of classification measures such as Accuracy (99.77), Precision (99.88), Recall (92.74), AUC (67.87), G-mean (92.66), and F-Measure (76.00) and the cumulative return of 21.47% obtained after three months of investment simulation outperformed the other results found after an extensive literature review. These positive results can be accredited to the precise workflow developed, the proper use of F-Measure and G-Mean as classification measures and process adjusting, and the new algorithms KNN-Und, and CATS, proposed in this work. It was also observed that the classifier performance decreased while the time offset was increased to two, three, and five minutes, but even below five minutes the results were satisfactory, if compared with other results published in the literature. This work is the first one to report successful results with time offsets below five minutes, which is in accordance with the tendencies of high frequency trading.

These results show evidences that the stock prices movement can be predicted using text mining, and indicates that the stock prices started to be affected by the news articles in the few minutes after they are published, and that a loss of signal occurs when the news articles are accumulated in a wider period, because there is no mechanism developed to distinguish which news articles are affecting the stock price, making it more difficult to obtain a stable model under these conditions. Despite the good results presented in the experiments, the association between news article and prices accumulated in a wider period deserves more attention in a future work.

An extensive survey about TMFP was conducted, and it was identified that about 50% of the reviewed works published their results only according to Accuracy, and few works devoted some attention to the data imbalance problem. This practice raises questions about these published results, because the Accuracy measure lacks sensitivity to data distributions. Another concern is regarding the lack of information about how the classifier model was evaluated in 27% of the reviewed works, and the inappropriate use of cross validation in 11% of the reviewed works. These problems raise concerns about the reproducibility and validity of these researches outside a backtesting environment, and diminish the investors' confidence in the application of TMFP in a real investment scenario.

It seems to be a utopic idea, but the correct forecast of price movements and other economic events is something that can change the face of financial markets as we know today, bringing transparency and confidence to this important instrument of human development. With the contributions presented in this work, the authors hope some steps in this direction have been given.

6.1 – Future Work

Beyond the methodology proposed in this work, some other aspects deserve further investigation in a future work.

According to the bibliographic review, this branch of research lacks a standard benchmark dataset. To improve this scenario, and at the same time to provide means of experiment reproduction, the raw data used in this work is available for download at Open Science Framework[11].

The CATS algorithm deserves further improvements regarding the rule optimization process with GA, for example, the use of the last week in the training set, instead to use a cross validation in the entire training set. Another improvement is the use of normalized counting values, new measurements, and other attributes to be included in the decision process. A further investigation about new ensemble strategies to be applied to CATS, and the use of this algorithm in other time series datasets are also valid approaches.

The entire TMFP process developed in this work could be applied to online test, i.e., the predictive model could be constructed with recent historical data, and the classification and simulation could be performed on fresh new data. This approach can be the last and safe step to apply TMFP in a real investment scenario. Nevertheless, there are scalability and time constraints according to the number of news articles to be processed, and the time offset to predict the price movements. One way to overcome this problem is to designate this processing to a server with high availability of memory and CPU cores. Nevertheless, in the case the high capacity server is not enough, the migration of pre-processing and machine learning algorithms to Hadoop environment can be done using the big data extension developed for RapidMiner in (Beckmann, et al., 2014). The use of Hadoop environment in this branch of research is not explored, as there is no mention about this aproach among the reviewed works.

[11] https://osf.io/gc6u6/

The experiments demonstrated that the wider the time offset, the bigger the number of news articles accumulated along that period, and this causes a decay in the classifier performance. One of the possible causes for this problem is the current process that labels the news articles given a stock price, the news alignment algorithm. The current news alignment algorithm is not able to measure how each news article affects the price at the end of the time offset. To overcome this problem, an appropriate news alignment algorithm for wider time offsets must be developed.

The use of t-SNE (Van der Maaten, 2014) together with unsupervised learning and other visualization techniques could be more explored to understand the relationship and meaning of words or group of words given the classifier outcome.

According to the literature review, the use of deep learning algorithms was barely explored in this branch of research, then the use of these new techniques in combination with the methodologies proposed in this work is also recommended. This combination could be used to obtain better predictive models for the 20 companies that did not perform well in this work.

The TMFP process developed in this work can be applicable to other markets like ForEx, and it is also adaptable for sentiment detection and automatic textual content interpretation to be used in fundamental analysis, risk, volume, and the forecast of other economic events.

References

Aizerman, A., Braverman, E. & Rozoner, L., 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control.*

Aldridge, I., 2013. *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems.* s.l.:Wiley Trading.

Ali, A., Shamsuddin, S. & Ralescu, A., 2013. Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl, Vol. 5, No. 3*.

Argentini, A. & Blanzieri, E., 2010. About Neighborhood Counting Measure Metric and Minimum Risk Metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 4(32), pp. 763-765.

Arlot, S. & Celisse, A., 2010. A survey of cross-validation procedures for model selection. *Statistics Surveys, 4,* pp. 40-79.

Aronson, D., 2007. *Evidence-based technical analysis: applying the scientific method and statistical inference to trading signals.* Hoboken, N.J.: John Wiley & Sons.

Balsara, N., Chen, G. & Zheng, L., 2007. The Chinese Stock Market: An Examination of the Random Walk Model and Technical Trading Rules. *The Quarterly Journal of Business and Economics, Spring.*

Barandela, R., Sánchez, J., García, V. & Rangel, E., 2003. Strategies for learning in class imbalance problems. *Pattern Recognition,* 36(3), pp. 849-851.

Beckmann, M., Ebecken, N. & De Lima, B., 2014. *A User Interface for Big Data Using RapidMiner.* Massachusets, Rapid-I.

Beckmann, M., Ebecken, N. & De Lima, B., 2015. A KNN Undersampling Approach for Data Balancing. *JILSA - Journal of Intelligent Learning Systems and Applications,* pp. 7, 104-116.

Bengio, Y., 2009. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning. 2.*

Bergmeir, C., Hyndman, R. & Koo, B., 2015. *A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction,* s.l.: Monash University, Departmnet of Econometrics and Business Statistics.

Bollen, J. & Huina, M., 2011. Twitter mood as a stock market predictor. *Computer, 44*, p. 91–94.

Boriah, S., Chandola, V. & Kumar, V., 2007. *Similarity Measures for Categorical Data: A Comparative Evaluation.* Minneapolis, s.n., pp. 243-254.

Boser, B., Guyon, I. & Vapnik, V., 1992. *A Training Algorithm for Optimal Margin Classifiers,* Berkeley, CA: EECS Department, University of California, Berkeley; AT&T Bell Laboratories.

Breiman, L., 1996. Bagging predictors. *Machine Learning,* 2(24), p. 123–140.

Breiman, L., 2001. Random Forests. *Machine Learning. 45 (1),* pp. 5-32.

Browning, E., 2007. Reading market tea leaves. *The Wall Street Journal Europe. Dow Jones.,* p. 17–18.

Butler, M. & Kešelj, V., 2009. Financial forecasting using character n-gram analysis and readability scores of annual reports. *Advances in artificial intelligence,* p. pp. 39–51.

Cambria, E., Schuller, B., Xia, Y. & Havasi, C., 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems. 28 (2),* pp. 15-21.
Camerer, C. & Loewenstein, G., 2004. *Advances in Behavioral Economics.* s.l.:Princeton University Press.

Carpenter, B., 2007. *LingPipe for 99.99% Recall of Gene Mentions.* Valencia, Spain, s.n.

Chang, C. & Lin, C., 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology,* pp. 2:27:1-27:27.

Chatrath, A., Miao, H., Ramchander, S. & Villupuram, S., 2014. Currency jumps, cojumps and the role of macro news. *Journal of International Money and Finance, 40,* p. 42–62.

Collobert, R. & Weston, J., 2008. *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.* New York, NY, USA, ACM, pp. 160-167.

Cortes, C. & Vapnik, V., 1995. Support-Vector Networks. *Machine Learning,* 09, 20(3), pp. 273-297.

Crone, S. & Koeppel, C., 2014. *Predicting Exchange Rates with Sentiment Indicators: An Empirical Evaluation using Text Mining and Multilayer Perceptrons.* s.l., s.n., pp. 114-121.

Dasarathy, B., 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* s.l.:IEEE Computer Society Press.

Das, S. R. & Chen, M. Y., 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science, 53,* p. 1375–1388.

De Faria, E., Ebecken, N. & Albuquerque, M., 2012. *A Methodology for Bovespa Index Forecasting Using Text Mining,* Rio de Janeiro: Federal University of Rio de Janeiro.

Dhaka, V., Kausar, M. & Singh, S., 2013. Web Crawler: A Review. *International Journal of Computer Applications, Vol 63, No. 2,* pp. 31-36.

Dietterich, T., 2001. Ensemble methods in machine learning. In: *Multiple Classifier Systems.* s.l.:Springer, pp. 1-15.

Dietterich, T., 2002. *Machine Learning for Sequential Data: A Review.* London, Springer-Verlag, pp. 15-30.

Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM, Vol. 55 Issue 10*, October, pp. 78-87.

Donoho, D., 2000. *High-dimensional data analysis: The curses and blessings of dimensionality.* s.l., s.n.

Duda, R., Hart, P. & Stork, D., 2001. Pattern Classification. In: 2nd Edition ed. New York: John Wiley & Sons Ltd., pp. 202-220.

Duman, E., Ekinci, Y. & Tanrıverdi, A., 2012. Comparing alternative classifiers for database marketing: The case of imbalanced datasets. *Expert Systems with Applications, 39,* p. 48–53.

Elder, A., 1993. *Trading for a Living; Psychology, Trading Tactics, Money Management.* s.l.:John Wiley & Sons.

Fama, E., 1965b. The Behavior of Stock-Market Prices. *The Journal of Business, Vol. 38, No. 1,* pp. 34-105.

Fama, E., 1965. Random Walks in Stock Market Prices. *Financial Analysts Journal,* September/October, 21(5), pp. 55-59.

Fama, E., 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance,* Volume 25, p. 383–417.

Fawcett, T., 2004. *ROC Graphs: Notes and Practical Considerations for Researchers,* s.l.: HP Laboratories.

Federighi, E., 1959. Extended Tables of the Percentage Points of Students's t-Distribution. *Journal of the American Statistical Association,* September, 54(287), pp. 683-688.

Fehrer, R. & Feuerriegel, S., 2016. *Improving Decision Analytics with Deep Learning: The Case of Financial Disclosures.* Istanbul, Turkey, s.n.

Fix, E. & Hodges, J., 1951. *Discriminatory analysis, nonparametric discrimination: Consistency properties, Technical Report 4,* Randolph Field, Texas: USAF School of Aviation Medicine.

Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research 3,* pp. 1289-1305.

Frank, E., Hall, M. & Witten, I., 2016. The WEKA Workbench. Online Appendix. In: *Data Mining: Practical Machine Learning Tools and Techniques, 4th Ed..* s.l.: Morgan Kaufmann.

Fung, P. C., G., X. Y. J. & Wai, L., 2003. *Stock prediction: Integrating text mining approach using real-time news.* s.l., s.n., pp. 395-402.

Gama, J. & Brazdil, P., 2000. Cascade Generalization. *Machine Learning, Vol. 41 (3),* pp. 315-343.

Gerig, A., 2015. *High-Frequency Trading Synchronizes Prices in Financial Markets,* s.l.: U.S. Securities and Exchange Commission (SEC), Division of Economic and Risk Analysis (DERA).

Gidófalvi, G. & Elkan, C., 2003. *Using News Articles to Predict Stock Price Movements. Technical Report,* San Diego: Department of Computer Science and Engineering, University of California.

Gimpel, K. et al., 2011. *Part-of-speech tagging for twitter: annotation, features, and experiments.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 42-47.

Go, A., Bhayani, R. & Huang, L., 2009. *Twitter sentiment classification using distant supervision,* s.l.: Stanford University.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston, MA: Addison-Wesley.

Gosset, W., 1908. The Probable Error of a Mean. March, 6(1), pp. 1-25.

Graham, B., Dodd, D., Klarman, S. & Buffett, W., 2008. *Security Analysis, 6th Ed..* s.l.:Mcgraw-Hill.

Griffioen, G., 2003. *Technical Analysis in Financial Markets (March 3, 2003). Available at SSRN: https://ssrn.com/abstract=566882 or http://dx.doi.org/10.2139/ssrn.566882,* Amsterdam: University of Amsterdam - Faculty of Economics and Business (FEB).

Groth, S. & Muntermann, J., 2011. An intraday market risk management approach based on textual analysis. *Decision Support Systems,* pp. 680-691.

Hagenau, M., M., L., Hedwig, M. & Neumann, D., 2012. *Automated news reading: Stock Price Prediction based on Financial News Using Context-Specific Features.* Maui, Hawaii, s.n., pp. 1040 - 1049.

Halls-Moore, M., 2015. *Successful Algorithmic Trading.* London: Quantstart.com.

Hardie, W., Kleinow, T. & Stahl, G., 2008. *Applied Quantitative Finance, 2nd Ed..* 2 ed. Berlin: Springer.

Harris, Z., 1954. Distributional Structure. *Word, 10,* p. 146–162.

Haselton, G., Nettle, D. & Murray, D., 2005. The Evolution of Cognitive Bias. In: *The Evolutionary Psychology Handbook 2nd Ed..* s.l.:Wiley.

Hastie, T., Tibshirani, R. & Friedman, J., 2003. Model Assessment and Selection. In: *The Elements of Statistical Learning, Data Mining, Inference and Prediction.* New York: Springer Series in Statistics, pp. 245-247.

He., H. & Ma, Y., 2013. *Imbalanced Learning: Foundations, Algorithms, and Applications.* 1st Edition ed. s.l.:Wiley-IEEE Press.

He, H. & Garcia, E., 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering, Volume 21 Issue 9,* pp. 1263-1284.

Herring, R., Diebold, F. & Doherty, N., 2010. *The Known, the Unknown, and the Unknowable in Financial Risk Management: Measurement and Theory Advancing Practice.* Princeton, N.J: Princeton University Press.

Holland, J., 1975. *Adaptation in Natural and Artificial Systems.* Cambridge: MIT Press.

Hollingworth, C., Barker, L. & Samson, A., 2016. *The Behavioral Economics Guide,* s.l.: Behavioraleconomics.com.

Hsu, C. W., Chang, C. C. & J., L. C., 2003. *A practical guide to support vector classification,* Taipei, Taiwan: National Taiwan University.

Huang, C. et al., 2010. Realization of a news dissemination agent based on weighted association rules and text mining techniques. *Expert Systems with Applications, 37,* p. 6409–6413.

Iba, W. & Langley, P., 1992. *Induction of One-Level Decision Trees.* Aberdeen, Scotland, Morgan Kaufmann, pp. 233-240.

Irwin, S. & Park, C., 2007. What Do We Know About the Profitability of Technical Analysis?. *Journal of Economic Surveys, Vol. 21, No. 4,* pp. 786-826.

Japkowicz, N., 2003. *Class imbalances. Are we focusing on the right issue?.* Washington DC, USA, s.n.

Jin, F. et al., 2013. *Forexforeteller: Currency trend modeling using news articles.* Chicago, ACM, p. 1470–1473.

Johnson, B., 2010. *Algorithmic Trading & DMA: An Introduction to Direct Access Trading Strategies.* s.l.:4Myeloma Press.

Jurevičienė, D. et al., 2013. Assessment of Corporate Behavioural Finance. *Procedia - Social and Behavioral Sciences,* pp. 432-439.

Kaltwasser, P. R., 2010. Uncertainty about fundamentals and herding behavior in the FOREX market. *Physica A: Statistical Mechanics and its Applications, 389,* p. 1215–1222.

Keerthi, S., Shevade, S., Bhattacharyya, C. & Murthy, K., 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation. 13(3),* pp. 637-649.

Kim, Y., Jeong, S. & Ghani, I., 2014. Text Opinion Mining to Analyze News for Stock Market Prediction. *Int. J. Advance. Soft Comput. Appl., Vol. 6, No. 1.*

Kissell, R., 2013. *The Science of Algorithmic Trading and Portfolio Management 1st Edition.* s.l.:Elsevier.

Kittler, J., Hatef, M., Duin, R. & Matas, J., 1998. On Combining Pattern Classifiers: Methods and Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3),* pp. 226-239.

Kohavi, R. & Quinlan, J., 2002. Decision Tree Discovery, chapter 16.1.3. In: *Handbook of Data Mining and Knowledge Discovery.* s.l.:Oxford University Press, pp. 267-276.

Land, S., 2010. *Approaching Vega: The final descent - How to extend RapidMiner 5.0,* s.l.: Rapid-I.

Lavrenko, V. et al., 2000. *Language Models for Financial News Recommendation.* Washington, DC, ACM New York, pp. 389-396.

le Cessie, S. v. H. J., 1992. Ridge Estimators in Logistic Regression. *Applied Statistics. 41(1),* pp. 191-201.

Leng, G., McGinnity, T. & Prasad, G., 2005. An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. *Fuzzy sets and systems 150 (2),* pp. 211-243.

Li, F., 2010. The information content of forward-looking statements in corporate filings - a naïve Bayesian machine learning approach. *Journal of Accounting Research, 48,* p. 1049–1102.

Ling, C. X., Huang, J. & Zhang, H., 2003. AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In: B. C. Yang Xiang, ed. *Lecture Notes in Computer Science.* Halifax: s.n.

Lintner, J., 1965. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Review of Economics and Statistics, 47 (1),* p. 13–37.

Liou, C., Cheng, C., Lou, J. & Liou, D., 2014. Autoencoder for Words. *Neurocomputing, Volume 139,* p. 84–96.

Lo, A., Mamaysky, H. & Wang, J., 2000. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *Journal of Finance, 55,* pp. 1705-1765.

Lo, A. W., 2005. Reconciling efficient markets with behavioral finance: The adaptive markets hypothesis. *Journal of Investment Consulting.*

Lovins, J., 1968. Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics, Vol. 11,* pp. 22-31.

Lugmayr, A. & Gossen, G., 2012. *Evaluation of methods and techniques for language based sentiment analysis for DAX 30 stock exchange – a first concept of a 'LUGO' sentiment indicator.* s.l., s.n.

Mahajan, A., Dey, L. & Haque, S. M., 2008. *Mining financial news for major events and their impacts on the market.* s.l., s.n., p. 423–426.

Makrehchi, M., Shah, S. & Liao, W., 2013. *Stock Prediction Using Event-based Sentiment Analysis.* s.l., s.n., pp. 337-342.

Malkiel, B., 1973. *A Random Walk Down Wall Street: The Time-tested Strategy for Successful Investing.* New York: W.W. Norton.

McCulloch, W. & Pitts, W., 1943. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics. 5 (4,* p. 115–133.

Merton, R. C., 1973. Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science,* 4(1), p. 141–183.

Mierswa, I., 2006. *Evolutionary learning with kernels: a generic solution for large margin problems.* New York, ACM, pp. 1553-1560.

Mierswa, I. et al., 2006. *YALE: Rapid Prototyping for Complex Data Mining Tasks.* s.l., s.n.

Miller, G. A., 1995. WordNet: A lexical database for English. *Communications of the ACM, 38,* pp. 39-41.

Miller, R. & Shorter, G., 2016. *High Frequency Trading: Overview of Recent Developments,* Whashington, USA: Congressional Research Service.

Miner, G. et al., 2014. *Pratical Text Mining and Statistical Analysis for Non-structured Text Data Applications.* s.l.:Elsevier.

Mittermayer, M., 2004. *Forecasting Intraday Stock Price Trends with Text Mining Techniques.* Hawaii, Big Island, s.n., p. 64.

Moat, H. et al., 2013. Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Scientific Reports. 3,* p. 1801.

Mossin, J., 1966. Equilibrium in a Capital Asset Market. *Econometrica, Vol. 34, No. 4,* p. 768–783.

Nassirtoussi, A., Aghabozorgi, S., Waha, T. & Ling Ngo, D., 2015. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications 42*, p. 306–324.

Nassirtoussi, A., Aghabozorgi, S., Waha, T. & Ngo, D., 2014. Text mining for market prediction: A systematic review. *Expert Systems with Applications 41*, p. 7653–7670.

Neyman, J. & Pearson, E. S., 1933. On the Problem of the most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society,* 1 January.p. A. 231 (694–706): 289–337.

Ng, A. et al., 2010-2012. *Machine Learning.* [Online]
Available at:
http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex8/ex8.html
[Accessed September 2015].

Pang, B., Lee, L. & Vaithyanathan, S., 2002. *Thumbs up? Sentiment classification using machine learning techniques.* s.l., s.n., pp. 79--86.

Pearson, K., 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series*, Volume 5.
Peramunetilleke, D. & Wong, R. K., 2002. Currency exchange rate forecasting from news headlines. *Australian Computer Science Communications, 24,* p. 131–139.

Porter, M., 1980. An Algorithm for Suffix Stripping. *Program, Vol. 14, No. 3,* pp. 130-137.

Prati, R., Batista, G. & Monard, M., 2004. *Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior.* Mexico City, Springer Link, pp. 312-321.

Preis, T., Moat, H. & Stanley, H., 2013. Quantifying Trading Behavior in Financial Markets Using Google Trends. *Scientific Reports. 3: 1684,* p. 1684.

Qiong, G., Cai, Z., Zhu, L. & Huang, B., 2008. *Data Mining on Imbalanced Data Sets.* Phuket, Thailand, s.n., pp. 1020-1024.

Rachlin, G., Last, M., Alberg, D. & Kandel, A., 2007. *ADMIRAL: A data mining based financial trading system.* s.l., s.n., pp. 720-725.

Read, C. et al., 2013. *The Efficient Market Hypothesists.* London: Palgrave Macmillan.

Rish, I., 2001. *An empirical study of the naive Bayes classifier,* New York: IBM Research Division.

Robertson, S., 2004. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation. 60 (5),* p. 503–520.

Samuelson, P., 1972. Proof That Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review, Vol. 6, No. 2,* pp. 41-49.

Schapire, R., 2003. The Boosting Approach to Machine Learning: An Overview. In: *Nonlinear Estimation and Classification.* New York: Springer, pp. 149-171.

Schumaker, R. & Chen, H., 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions in Information System*, 27(2), pp. 1-19.

Schumaker, R. P., Zhang, Y., Huang, C. & Chen, H., 2012. Evaluating sentiment in financial news articles. *Decision Support Systems.*

Shadbolt, J. & Taylor, J., 2013. *Neural Networks and the Financial Markets: Predicting, Combining and Portfolio Optimisation (Perspectives in Neural Computing).* s.l.:Springer.

Sharpe, W., 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance, 19 (3),* pp. 425-442.

Sharpe, W. F., 1994. The Sharpe Ratio. *The Journal of Portfolio Management,* Issue 21, pp. 49-58.

Sidorova, G. et al., 2014. Syntactic N-grams as machine learning features for natural language processing. *Expert Systems with Applications, Vol. 41, Issue 3,* pp. 853-860.

Sidorov, G., Gelbukh, A., Gomez-Adorno, H. & Pinto, D., 2014. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas,* 18(3), p. 491–504.

Siering, M., 2012. *"Boom" or "Ruin" – Does it Make a Difference? Using Text Mining and Sentiment Analysis to Support Intraday Investment Decisions.* s.l., s.n., pp. 1051-1059.

Sokolova, M. & Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45,* pp. 427-437.

Soni, A., van Eck, N. J. & Kaymak, U., 2007. *Prediction of stock price movements based on concept map information.* s.l., s.n., pp. 205-211.
Sprothen, V., 2016. *Trading Tech Accelerates Toward Speed of Light.* [Online]
Available at: http://www.wsj.com/articles/trading-tech-accelerates-toward-speed-of-light-1470559173
[Accessed 14 Nov 2016].

Tetlock, P. C., Saar-Tsechansky, M. & Macskassy, S., 2008. More than words: Quantifying language to measure firms’ fundamentals. *The Journal of Finance, 63,* p. 1437–1467.

Thammasiria, D., Delenb, D., Meesadc, P. & Kasapd, N., 2014. A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications, Vol. 41, Issue 2,* pp. 321-330.

Thawornwong, S. & Enke, D., 2004. Forecasting Stock Returns with Artificial Neural Networks, Chap. 3.. In: *Neural Networks in Business Forecasting.* s.l.:IRM Pres.

Thomson, G., 2007. *Finance Theories Taxonomy,* Nevada USA: Nevada State College.

Tomer, J., 2007. What is behavioral economics?. *The Journal of Socio-Economics 36,* pp. 463-479.

Treynor, J., 1961. *Market Value, Time, and Risk,* s.l.: s.n.

Trippi, R. & Turban, E., 1996. *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance.* New York, NY, USA: McGraw-Hill.

Vakeel, K. & Shubhamoy, D., 2014. *Impact of News Articles on Stock Prices: An Analysis Using Machine Learning.* New York, ACM, pp. 1-4.

Van der Maaten, L., 2014. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research,* pp. 3221-3245.

Van Rijsbergen, C., 1979. *Information Retrieval.* 2nd ed. Massachusets: Butterworths.
Vapnik, V. & Lerner, A., 1963. Pattern recognition using generalized portrait method. *Automation and Remote Control,* Volume 24, p. 774–780..

Vu, T., Chang, S., Ha, Q. & Collier, N., 2012. *An experiment in integrating sentiment features for tech stock prediction in twitter.* Mumbai, India, The COLING 2012 Organizing Committee, pp. 23-38.

Weis, G., 2004. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets Homepage archive Volume 6 Issue 1,* pp. 7-19.

Weiss, G. & Provost, F., 2001. *The Effect of Class Distribution on Classifier Learning: An Empirical Study,* s.l.: Technical Report MLTR-43, Dept. of Computer Science, Rutgers University..

Weiss, S. M., Indurkhya, N. & Zhang, T., 2010. *Fundamentals of Predictive Text Mining.* s.l.:Springer Publishing Company, Incorporated.

Werner, A. & Murray, Z. F., 2004. Is all that talk just noise? The information content of internet stock message boards. *Journal of Finance, 10,* p. 1259–1294.

White, T., 2009. *Hadoop: The Definitive Guide, 1st Ed..* s.l.:O'Reilly.

Whitley, D. & Sutton, A., 2012. Genetic Algorithms — A Survey of Models and Methods. In: *Handbook of Natural Computing.* s.l.:Springer, pp. 637-671.

Widenius, M., Axmark, D. & DuBois, P., 2002. *Mysql Reference Manual.* 1st ed. Sebastopol, CA, USA: O'Reilly & Associates.

Wilson, D. & Martinez, T., 1997. Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research,* Volume 6, pp. 1-34.

Wilson, T. & Hoffmann, P., 2005. *OpinionFinder: a system for subjectivity analysis.* Vancouver, Canada, s.n.

Wong, F., Liu, Z. & Chiang, M., 2014. *Stock Market Prediction from WSJ: Text Mining via Sparse Matrix Factorization.* s.l., s.n.

Wuthrich, B. et al., 1998. *Daily Prediction of Major Stock Indices from textual WWW Data.* New York, NY, s.n., pp. 364-368.

Wu, X., Kumar, V., Quinlan, J. & al., e., 2007. Top 10 algorithms in data mining. Knowlegdment and Information System. *Springer-Verlag*, Volume 1, pp. 1-37.

Yang, S. et al., 2015. The Impact of Abnormal News Sentiment on Financial Markets. *Journal of Business and Economics, Vol. 6, No. 10 ,* pp. 1682-1694.

Young, T., 1991. Calmar Ratio: A Smoother Tool. *Futures (magazine).*

Yu, H., Nartea, G. V., Gan, C. & Yao, L. J., 2013. Predictive ability and profitability of simple technical trading rules: Recent evidence from Southeast Asian stock. *International Review of Economics and Finance, 25,* pp. 356-371.

Yu, Y., Duan, W. & Cao, Q., 2013. The impact of social and conventional media on firm equity value: A sentiment analysis approach. *Decision Support Systems*.

Zhai, C. & Massung, S., 2016. *Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining.* s.l.:ACM.

Zhai, Y., Hsu, A. & Halgamuge, S., 2007. *Combining news and technical indicators in daily stock price trends prediction.* Nanjing, China, Springer-Verlag, p. 1087–1096.

Appendix A

Table 17 - Main aspects of TMFP methodology over the years.

Reference / Year	Source of news	No. of items	Market/ Index/ Exchange	Time-Frame / Alignment offset	Period of news collection	Number of months	Number of Classes / Target prediction	Feature Selection / Representation	Dimensionality Reduction	Learning Algorithm	Training vs. testing	Sliding window	Sentiment Analysis	Semantics	Syntax	Balanced Data
(Wuthrich, et al., 1998)	WSJ, FT, Reuters, Down Jones, Bloomberg	-	Stocks from DJIA, Nikkei, FTSE, HSI, STI	Daily	6/Dec/1997-6/Mar/1998	3	3	BOW/ Binary	Expert dictionary	k-NN, ANN, Naïve Bayes, Rule-based	100 days vs. 1 day	Y	N	Y	N	-
(Lavrenko, et al., 2000)	Yahoo! Finance	38000	Stocks from NYSE, NASDAQ	Intraday/ 1 hour	15/Oct/1999-10/Feb/2000	4	5	BOW	Bayesian Language Models	Bayesian Language Models	3 months vs. 40 days	N	N	N	N	N
(Peramunetilleke & Wong, 2002)	HFDF93 via www.olsen.ch	960	ForEx USD-DEM, USD-JYP	Intraday/ 3 hours	22/Sep/1993-27/Sep/1993	0	3	BOW/ Boolean, TF-IDF, TF-CDF	Set of keywords	Decision tree and rules	22/Sep 12:00-27/Sep 09:00 vs. 9:00-10:00 on 27 Sep.	Y	N	Y	N	Y
(Fung, et al., 2003)	Reuters Market 3000	600000	33 stocks from HIS	Daily	1/Oct/2002–30/Apr/2003	7	2	BOW/ TF-IDF	Stemming, stop words	SVM	6 months vs. last month	N	N	N	N	-
(Gidófalvi & Elkan, 2003)	Yahoo! Finance	6300	12 stocks from NASDAQ	Intraday/ -20 to 20 minutes	14/Nov/1999 -11/Feb/2000	3	3	BOW, Wittenbell smoothing method	Stemming, stop words, highest 1000 words with mutual information	Naïve Bayes	4650 vs. 1650	N	N	N	N	-

(Mittermayer, 2004)	PRSNews-Wire	7002	Stocks from NYSE and NASDAQ	Daily	Year 2002	12	3	BOW/ TF-IDF	Selected 1000 terms	SVM	400 vs. 6602 examples	N	N	N	N	Y
(Werner & Murray, 2004)	Yahoo! Finance, WSJ Raging Bull	1500000	DJIA stocks	Daily	Year 2000	12	3	BOW/ Binary	Minimum information criterion (top 1000 words)	Naïve Bayes, SVM	1000 messages vs. the rest	N	N	N	N	N
(Das & Chen, 2007)	Message boards	145110	Stocks of 24 tech-sectors from MSH	Daily	Jul/2001-Aug/2001	2	Aggregate Sentiment index	BOW/ Triplets, discrete values for each classifier	Predefined dictionaries	Combinatory algorithms	1000 messages vs. the rest	N	Y	Y	Y	-
(Rachlin, et al., 2007)	forbes.com, reuters.com	-	5 stocks from NASDAQ	Daily	7/Feb/2006-7/May/2006	3	5	BOW/ common financial values, TF, Boolean, Extractor SW output	Automatic extraction of most influential keywords	C4.5 decision tree	-	N	N	N	N	-
(Soni, et al., 2007)	Financial Times Intelligence	3493	Stocks of 11 oil and gas companies	Daily	1/Jan/1995-15/May/2006	136	2	Visual coordinates	Thesaurus using term extraction tool	SVM w/ linear kernel	80% vs. 20%	N	N	Y	N	Y
(Zhai, et al., 2007)	Australian Financial Review	216	BHP Billiton Ltd. from ASX	Daily	1/Mar/2005-31/May/2006	14	2	BOW/ Binary, TF-IDF	Top 30 higher level concepts using WordNet	SVM w/ RBF and polynomial kernel	12 months vs. 2 months	N	N	Y	N	-
(Mahajan, et al., 2008)	-	700	Stocks from SENSEX	Daily	Aug/2005-Apr/2008	33	Categorical	LDA/ Binary	Extraction of twenty-five topics	Stacked classifier	Aug/2005-Dec/2007 vs. Jan/2008-Apr/2008	N	N	Y	N	-
(Tetlock, et al., 2008)	WSJ, Down Jones news from Factiva service	350000	Firms future cash flows from S&P 500	Daily	1980-2004	300	Regression	BOW for negative words/ Frequency divided by total words	Harvard-IV-4 psychosocial dictionary	OLS regression	33 trading days prior to an earnings announcement	Y	Y	Y	N	NA
(Butler & Kešelj, 2009)	Reports from companies’ websites	-	1 Year market drift of stocks	Yearly	2003-2008	72	2	BOW / Character n-grams, n-gram frequency	Minimum occurrence per document	CNG distance, SVM	x-1 and x-2 and all vectors vs. testing year	Y	N	N	Y	-

(Schumaker & Chen, 2009)	Yahoo! Finance	2800	S&P 500 stocks	Intraday/ 20 minutes	26/Oct/2005-28/Nov/2005	1	Categorical discrete numeric	BOW / noun phrases, named entities/ Binary	Minimum occurrence per document	SVM	-	N	N	Y	Y	-
(Huang, et al., 2010)	Leading electronic newspapers in Taiwan	12830	TAIEX stocks	Daily	Jun/2005-Nov/2005	6	Significant degree assignment	Simultaneous terms, ordered pairs / Weighted on the index fall/rise	Synonyms replacement	Weighted association rules	Jun/2005-Oct/2005 vs. Nov/2005	N	N	Y	Y	-
(Li, 2010)	Management discussion and Analysis section from SEC Edgar website	140000	(1) Index (2) Quarterly earnings and cash flows (3) Stock returns	Yearly	1994-2007	168	4	BOW, Tone and content / Binary, Dictionary value	Pre-defined dictionaries	Naïve Bayes and dictionary-based	30000 randomly vs. itself and the rest	N	N	N	N	N
(Bollen & Huina, 2011)	Twitter	9853498	DJIA stocks	Daily	28/Feb/2008-19/Dec/2008	10	Regression	Opinion finder	Opinion finder	Self organizing fuzzy NN	28/Feb-28/Nov vs. 1/Dec-19/Dec	N	Y	NA	NA	-
(Groth & Muntermann, 2011)	Adhoc corporate disclosures	423	Stock Market Risk	Intraday/ 15 minutes	1/Aug/2003-31/Jul/2005	24	2	BOW/ TF-IDF	Information Gain and Chi-Squared	Naïve Bayes, k-NN, ANN, SVM	Stratified cross validation	N	N	N	N	N
(De Faria, et al., 2012)	Macro-economic, financial news, social media	174993	Blue chips stocks from BOVESPA	Daily	23/Feb/2010-30/Jun/2011	17	3	BOW/ TF, TF-IDF, TF-CDF	Keep titles only, stop words, stemming, small dictionary	SVM, MLP, RBF, Naïve Bayes	Cross validation	N	N	Y	N	Y
(Hagenau, et al., 2012)	DGAP, EuroAdhoc	14348	Company specific stock	Daily	1997-2011	180	2	BOW/ noun phrases, n-grams / TF-IDF	Chi-Squared + Bi-normal separation for exogenous-feedback.	SVM linear, SVR	-	N	N	Y	Y	N
(Lugmayr & Gossen, 2012)	Broker newsletter	-	Stocks from DAX 30	Intraday/ open,	-	0	3	BOW/ Sentiment value	Stemming	SVM	-	N	Y	Y	N	-

				midday, close												
(Schumaker, et al., 2012)	Yahoo! Finance	2802	Stocks from S&P 500	Intraday/ 20 minutes	26/Oct/2005-28/Nov/2005	1	Regression	Opinion Finder overall tone and polarity / Binary	Minimum occurrence per document	SVR	-	N	Y	Y	N	-
(Siering, 2012)	Down Jones News	11518	DAX blue chips stocks	Intraday/ 15 minutes	06/Apr/2006-08/Apr/2008	24	3	BOW / TF-IDF	Porter stemming, stop words, Info Gain	SVM w/ linear kernel	Cross validation	N	Y	N	N	-
(Vu, et al., 2012)	Twitter	5001460	NASDAQ Stocks AAPL, GOOG, MSFT, AMZN	Daily	1/Apr/2011-31/May/2011 online test: 8/Sep/2012-26/Sep/2012	12	2	Daily number of pos/neg on TST+ emoticon lexicon + PMI / Real number of pos/neg and bullish/bearish anchor words	Pre-defined company related keywords, Named Entity Recognition.	C4.5 decision tree	Previous day vs. current day	Y	Y	Y	Y	-
(Jin, et al., 2013)	General news from Bloomberg	361782	ForEx	Daily	2012	12	Regression	LDA / Each article's topic distribution	Manual top identification by manually aligning news articles with currency fluctuations.	Linear regression model	Previous day vs. a given day	Y	Y	Y	N	-
(Makrehchi, et al., 2013)	Twitter	30M	S&P 500 index	Daily	27/Mar/2012 -13/Jul/2012	2.5	2	BOW / Binary	Mood word list	Rocchio	Cross validation	N	Y	Y	N	Y
(Yu, et al., 2013)	Blogs, forums, news, micro blogs (e.g., Twitter)	52746	AR and CAR from stocks of 824 firms	Daily	1/Jul/2011-30/Sep/2011	3	2	BOW / Binary	-	Naïve Bayes	-	N	Y	Y	N	-
(Crone & Koeppel, 2014)	Reuters MarketPysch	783	ForEx AUD-USD	Daily	4/Sep/2009-4/Sep/2012	36	2	14 built-in sentiment indicators from Reuters	NA	MLP	Cross validation	N	Y	N	N	-

(Kim, et al., 2014)	Naver.com	78216	KOSPI, stocks of 2 media firms	Daily	2011	12	2	BOW / TF	Stop words, automated sentiment dictionary	-	01/01/2011-31/Jul/2011 vs. 1/Aug/2011-31/Dez/2011	N	Y	Y	N	-
(Vakeel & Shubhamoy, 2014)	Times of India, Economic Times	3253	SENSEX Stocks	Pre/Post Election	17/Feb/2014-13/June/2014	4	2	BOW / TF-IDF, n-grams	Information Gain	SVM	80% vs. 20%	Y	N	N	Y	-
(Wong, et al., 2014)	WSJ	-	Stocks from DJIA, S&P 500, NASDAQ	Daily	1/Jan/2008-30/Sep/2013	69	2	Sparse Matrix Factorization + ADMM	Sparse Matrix Factorization + ADMM	Sparse Matrix Factorization + ADMM	2008-2011 vs. validation: 2012 vs. test: 2013	N	N	N	N	-
(Nassirtoussi, et al., 2015)	MarketWatch .com & others	6096	ForEx EUR/USD	Intraday/ 1 hour	2008-2011	48	2	BOW/ TF-IDF, SumScore weighting	Synchronous Target Feature-Reduction, WordNet	SVM, k-NN, Naïve Bayes	Several tests with training data proportion >=0.99	Y	Y	Y	Y	-
(Yang, et al., 2015)	Northern Light business news	678378	S&P 500 index	Daily	13/Jul/2012-16/Oct/2014	27	Regression	BOW/ Daily sentiment score from dictionary	Stemming, stop words	Regression with abnormal sentiment scores	Training = Test	N	Y	Y	N	-
(Fehrer & Feuerriegel, 2016)	Adhoc reports from DGAP	8359	Stocks from German firms	Daily	Jan/2004-Jun/2011	90	3	Neural Networks - Recursive auto encoders	Neural Networks - Recursive auto encoders	Neural Networks - Recursive auto encoders	80% vs. 20%	N	Y	N	Y	-
This current work	**Yahoo! Finance, Google Finance**	**128195**	**DJIA stocks**	**Intraday, 1,2,3, and 5 minutes**	**01/Jan/2013-03/Sep/2013**	**9**	**2**	**BOW/ TF-IDF, n-grams**	**Chi Square, stop words, min/max occurrence per document**	**LIBSVM w/ RBF kernel**	**Last 6 months vs. 1 week, then repeating for 3 months**	**Y**	**N**	**N**	**Y**	**Y**

Table 18 - Average results for both negative (NOT RECOMENDED) and positive (SURGE) classes, using a time offset of one minute ($\tau=1$) after the news article to be released.

Stock Symbol	Precision -	Precision +	Recall -	Recall +	F-Measure -	F-Measure +
AA	50.00 (0.0)	99.61 (0.1)	99.58 (0.0)	52.38 (5.8)	99.60 (0.0)	51.02 (2.5)
AXP	27.14 (29.7)	98.66 (0.2)	94.30 (2.3)	44.16 (14.3)	96.42 (1.1)	22.16 (2.2)
BA	40.00 (0.0)	99.43 (0.0)	99.50 (0.0)	36.88 (1.3)	99.47 (0.0)	38.37 (0.7)
BAC	32.14 (2.9)	99.19 (0.0)	99.86 (0.0)	7.22 (0.2)	99.52 (0.0)	11.76 (0.0)
CAT	48.57 (21.0)	98.99 (0.0)	99.48 (0.2)	26.87 (4.2)	99.23 (0.1)	32.65 (1.7)
CVX	19.15 (2.4)	98.92 (0.1)	98.75 (0.0)	21.90 (4.7)	98.84 (0.0)	20.41 (3.3)
DD	18.81 (9.3)	98.35 (0.3)	95.52 (1.0)	37.50 (10.2)	96.91 (0.7)	24.88 (10.3)
DIS	90.48 (23.3)	99.29 (0.1)	99.94 (0.2)	14.80 (5.6)	99.61 (0.0)	23.44 (3.0)
GE	17.38 (0.7)	99.91 (0.0)	98.51 (0.1)	78.57 (8.7)	99.20 (0.0)	28.38 (0.5)
HD	39.05 (2.3)	98.59 (0.0)	99.49 (0.1)	18.73 (3.1)	99.04 (0.0)	25.24 (3.5)
HPQ	57.14 (17.5)	98.73 (0.0)	99.91 (0.0)	7.78 (0.2)	99.32 (0.0)	13.62 (0.7)
IBM	100.00 (0.0)	99.19 (0.0)	100.00 (0.0)	20.32 (0.8)	99.60 (0.0)	33.76 (1.1)
INTC	100.00 (0.0)	99.70 (0.0)	100.00 (0.0)	21.43 (8.7)	99.85 (0.0)	34.29 (14.0)
JNJ	89.29 (26.2)	98.21 (0.0)	99.93 (0.2)	7.78 (0.2)	99.07 (0.1)	14.03 (0.6)
JPM	29.95 (8.3)	99.50 (0.0)	99.63 (0.4)	18.18 (3.7)	99.57 (0.2)	21.09 (2.8)
KO	100.00 (0.0)	99.01 (0.0)	100.00 (0.0)	11.31 (0.5)	99.50 (0.0)	20.32 (0.8)
MCD	57.14 (7.0)	99.19 (0.1)	99.64 (0.2)	35.71 (5.8)	99.41 (0.0)	43.09 (0.6)
MMM	16.42 (5.1)	99.93 (0.2)	89.05 (6.9)	96.43 (8.7)	94.03 (4.2)	27.83 (8.4)
MRK	23.16 (3.1)	99.36 (0.1)	92.12 (1.9)	79.43 (2.1)	95.60 (1.0)	35.70 (3.9)
MSFT	100.00 (0.0)	99.77 (0.0)	100.00 (0.0)	14.29 (5.8)	99.88 (0.0)	24.49 (10.0)
PFE	100.00 (0.0)	98.57 (0.0)	100.00 (0.0)	17.01 (2.9)	99.28 (0.0)	28.97 (4.4)
PG	88.57 (28.0)	99.00 (0.1)	99.90 (0.2)	14.63 (0.8)	99.45 (0.1)	24.03 (2.4)
T	57.14 (17.5)	99.48 (0.0)	99.92 (0.0)	14.63 (0.8)	99.70 (0.0)	23.13 (2.2)
TRV	100.00 (0.0)	99.22 (0.1)	100.00 (0.0)	35.71 (5.8)	99.61 (0.1)	52.38 (5.8)
UNH	100.00 (0.0)	98.13 (0.0)	100.00 (0.0)	28.57 (11.7)	99.06 (0.0)	42.86 (17.5)
UTX	12.00 (19.6)	98.52 (0.2)	86.40 (5.4)	33.93 (1.5)	91.99 (3.0)	12.71 (13.7)
VZ	3.41 (6.8)	50.91 (19.7)	14.47 (34.8)	55.98 (18.8)	14.66 (34.6)	2.98 (4.2)
WMT	34.38 (13.8)	95.87 (8.5)	85.51 (34.8)	31.55 (22.8)	85.40 (34.6)	24.67 (9.5)
XOM	25.20 (30.5)	99.34 (0.1)	95.34 (1.9)	52.59 (14.0)	97.29 (0.9)	22.31 (3.5)

Table 19 - Glossary of terms and acronyms.

Term	Description
ADMM	Alternating Direction Method of Multipliers is an optimization algorithm suitable for non-convex problems.
AMH	Adaptive Market Hypothesis
AR	Abnormal return, a return of investment above the average and expectations.
Asset	An asset is a resource with economic value that an individual, corporation or country owns or controls with the expectation that it will provide future benefit.
ATS	Automated Trading System
Backtesting	The process of testing a trading strategy or algorithm on historical data to ensure its viability before to apply it in a real investment scenario.

Bearish	Represents a wish or trend for a fall in the price of an asset or market.
Blue Chips	A stock from a reputed and stable company.
BOW	Bag of Words
BOVESPA	São Paulo Exchange
Bullish	Represents a wish or trend for a rise in the price of an asset or market.
CAPM	Capital Asset Price Model
CAR	Cumulative abnormal return
CATS	Cascading Aggregation for Time Series
CR	Cumulative return
Daily	Trading operations with one day of duration.
DAX	Index with the 30 major companies from Germany.
DGAP	German Society for Ad Hoc Publicity
DJIA	Down Jones Industrial Average, is a stock market index that represents 30 large publicly owned companies based in the United States.
ENET	Elastic-net logistic regression
Equity Market	Same as Stock Market
Exchange	A highly-organized market where tradable securities, commodities, foreign exchange, futures, and options contracts are sold and bought.
Financial Instrument	Financial instruments are assets that can be traded.
FA	Fundamental Analysis
ForEx	Foreign Exchange, it is the financial market where currencies are traded.
FT	Financial Times
FTSE	Financial Times Stock Exchange 100 is an index calculated from 100 companies listed on the London Stock Exchange (LSE).
Future Market	A market where the long-term contracts are traded. The parties agreed in the present, a buy and sell price of an asset to be traded in the future.
GA	Genetic Algorithm
Hyperplane	In geometry, it is a representation of *n-1* dimension, being *n* the current number of available dimensions. For example, a 1-dimensional line is the hyperplane in 2 dimension spaces, a 2-dimensional plane is the hyperplane in 3 dimension spaces, and so on.
Hyperparameter	A parameter provided by the user to be applied by the pre-processing and machine learning algorithms.
HKEx	Hong Kong Stock Exchange
HSI	Hang Seng Index, is constituted with the 50 companies from HKEx.
Intraday	Trading operations with less than one day of duration.
KOSPI	Korea Composite Stock Price Index
LDA	Latent Dirichlet Allocation

LSE	London Stock Exchange
MLP	Multi-Layer Perceptron neural network
NASDAQ	National Association of Securities Dealers Automated Quotations, is an American stock exchange, and concentrates the trading of the most important technology companies in the world.
Nikkei 225	The main stock index from Tokyo Stock Exchange (TSE).
NYSE	New York Stock Exchange, is the largest exchange in the world by volume and market capitalization.
Order, Order execution	The command to buy or sell financial instruments sent to an exchange.
PMI	Pointwise Mutual Information
Portfolio	A collection of investments held by an investment company or individual.
POS	Part of Speech, it is used to capture a sentence’s syntactic aspects.
RBF	Radial Basis Function
Roundtrip	The entire operation of to buy and sell a share or other security.
S&P 500	Standard & Poor Index, which aggregates 500 American companies.
SENSEX 30	The stock index from the Bombay Stock Exchange (BSE).
Share	Share is a portion of a stock.
Security	Same as financial instrument, but its legal definition varies according the jurisdiction.
STI	FTSE Straight Times Index is constituted from the top 30 companies from Singapore Exchange (SGX).
Stock	A stock is a type of financial instrument that grants ownership in a corporation and gives the right to claim for part of the corporation's assets and earnings.
Stock Market	A group of buyers and sellers with the common interest to trade shares of stock.
SVM	Support Vector Machine
SVR	Support Vector Machine for Regression
TA	Technical Analysis
TF	Term frequency, number of occurrence of a term in a document, divided by the total number of terms in a document.
TF-IDF	Term frequency-inverse document frequency
TMFP	Text mining applied to financial market prediction.
TSE	Tokio Stock Exchange
TST	Twiter Sentiment Tool
UTC	Coordinated Universal Time
WSJ	Wall Street Journal