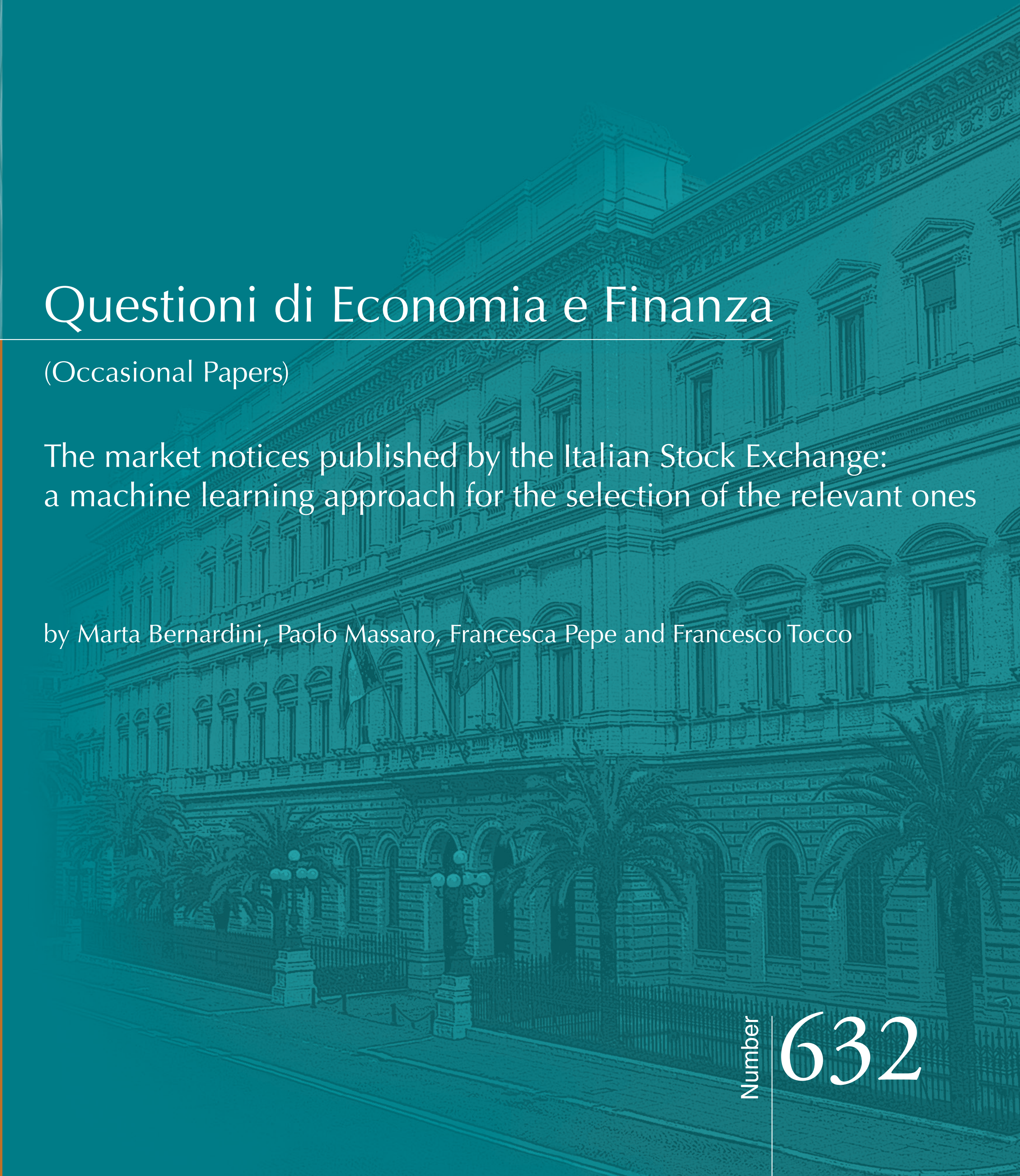
BANCA D'ITALIA
EUROSISTEMA



Questioni di Economia e Finanza

(Occasional Papers)

The market notices published by the Italian Stock Exchange: a machine learning approach for the selection of the relevant ones

by Marta Bernardini, Paolo Massaro, Francesca Pepe and Francesco Tocco

July 2021

Number 632

BANCA D'ITALIA
EUROSISTEMA

Questioni di Economia e Finanza

(Occasional Papers)

The market notices published by the Italian Stock Exchange: a machine learning approach for the selection of the relevant ones

by Marta Bernardini, Paolo Massaro, Francesca Pepe and Francesco Tocco

Number 632 – July 2021

The series Occasional Papers *presents studies and documents on issues pertaining to the institutional tasks of the Bank of Italy and the Eurosystem. The* Occasional Papers *appear alongside the* Working Papers *series which are specifically aimed at providing original contributions to economic research.*

The Occasional Papers *include studies conducted within the Bank of Italy, sometimes in cooperation with the Eurosystem or other institutions. The views expressed in the studies are those of the authors and do not involve the responsibility of the institutions to which they belong.*

The series is available online at www.bancaditalia.it .

ISSN 1972-6627 (print)
ISSN 1972-6643 (online)

Printed by the Printing and Publishing Division of the Bank of Italy

THE MARKET NOTICES PUBLISHED BY THE ITALIAN STOCK EXCHANGE: A MACHINE LEARNING APPROACH FOR THE SELECTION OF THE RELEVANT ONES

by Marta Bernardini[†], Paolo Massaro[*], Francesca Pepe[*] and Francesco Tocco[*]

Abstract

Bank of Italy data managers check the market notices published daily by the Italian Stock Exchange (Borsa Italiana) and select those of interest to update the Bank of Italy's Securities Database. This activity is time-consuming and prone to errors should a data manager overlook a relevant notice. In this paper we describe the implementation of a supervised model to automatically select the market notices. The model outperforms the manual approach used by data managers and can therefore be implemented in the regular process to update the Securities Database.

JEL Classification: C18, C81, G23.
Keywords: machine learning, Securities Database, automatic selection, Italian Stock Exchange.
DOI: 10.32057/0.QEF.2021/0632

Contents

[†] Bank of Italy, IT Development Directorate.

[*] Bank of Italy, Statistical Data Collection and Processing Directorate.

1 Introduction[*]

The availability of complete and high quality statistical information plays a key role for central banks since policy decisions are based upon data. Databases containing the descriptive characteristics of specific entities are a key ingredient in the statistical production of a central bank. Indeed, their importance has progressively increased, hand-in-hand with the introduction of more granular surveys; this is why the quality of such information, in terms of accuracy and completeness, has become pivotal.

This paper focuses on a text mining application to improve the quality of the Securities Database managed by the Bank of Italy, the centralized database containing the attributes of the securities issued on the Italian market and of the foreign securities kept in custody by supervised institutions. In order to maintain and enrich this Database, the Bank of Italy relies on many and diversified sources, such as issuers, placement agents, commercial data providers and market venues. Sometimes, the required data is not available in a structured format[1] and data managers have to undertake lengthy searches through documents containing the relevant information. Some Market Notices published by the Italian Stock Exchange (*Borsa Italiana*)[2] contain relevant information for the timely update of the Securities Database. Hence, every day Bank of Italy data managers check these Market Notices, select those of interest and then use the relevant information to update the Securities Database. The process of "manual" selection is error prone and highly time and resource consuming.

Previous research has shown the importance of machine learning techniques to process textual data for classification and other related tasks.[3] Standard examples in the literature are (automatic) spam detection (Metsis et al., 2006) and the labelled dataset of newswire articles from Reuters[4] (Yang and Liu, 1999). Recent applications of text mining methods carried out in the Bank of Italy also confirm their importance in enhancing data quality and underline the efficiency of the production processes. In particular, Carboni and Moro (2018) demonstrate that machine learning techniques can be applied to infer the nationality of companies from their name and this allows us to enhance the quality of the Foreign Direct Investments item of the Italian Balance of Payments and International Investment Position. Zambuto et al. (2021) apply a model for the efficient processing of requests for confirmations of quality remarks received from reporters during the data quality management process.

[*] We are grateful to Professors Gianluca Cubadda and Alessio Farcomeni (University of Tor Vergata) for their useful comments and fruitful discussions on a preliminary draft of the paper. We also wish to thank the data managers Daniela Giarrusso and Daniela Ruo (Bank of Italy) for their valuable contribution to the detailed description of their daily notice selection activity. The views expressed herein are those of the authors and do not necessarily reflect those of the Bank of Italy

[1] Or the data is only available in a structured format from commercial sources.

[2] https://www.borsaitaliana.it.

[3] An overview is available in (Bishop, 2007).

[4] Reuters-21578, Distribution 1.0, available at http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

The aim of this paper is to explore the possibility of estimating a classifier to automatically find the notices that are useful to update the Securities Database. For this purpose, we use a dataset of notices labelled either "to be selected" or "not of interest" to train a model following a supervised approach. Empirical results show that the performance of our approach is higher than that obtained by following a manual approach. Since April 2020 this approach has been supporting the Bank of Italy's daily procedures for updating the Securities Database and both the data quality and the efficiency of the process have improved.

This paper is organized as follows: Section 2 describes the problem in more detail; Section 3 illustrates the available data and the construction of the dataset; Section 4 briefly gives the background of the machine learning methods that have been applied; Section 5 explains the iterative process followed to improve data quality, evaluates different approaches and selects the final model for the notice selection task; and Section 6 summarizes the main results.

2 The problem

Borsa Italiana regulates the procedures for listing companies and supervises the disclosure of information for listed companies through the publication of around 25 thousand Market Notices every year, of which about 4 thousand include information useful to the Bank of Italy in order to update the Securities Database. On a daily basis data managers check the published notices (over 100 per day) and carry out two separate tasks:

1. notice selection: the data manager assesses whether the notice contains relevant data for the Securities Database;
2. data extraction: once the notice is classified as useful, the relevant data are extracted and collected in the Securities Database.

Both tasks are quite time and resource consuming. In this paper we focus on the automation of the first task, which can be modelled as a binary classification task; the second one, which is more a NLP and text extraction issue, is left to the next stage of our research programme.

The main Market Notices of our interest refer to the following areas:

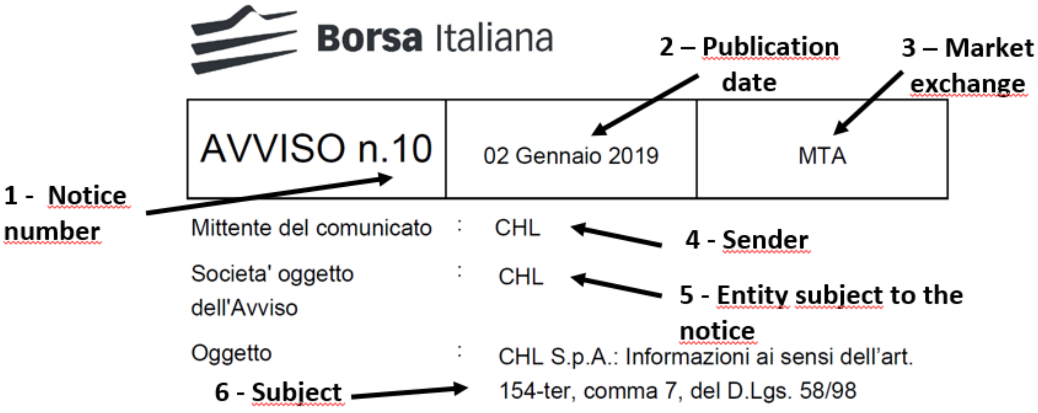
- knock-out events;
- size changes;
- coupon communications related to Italian companies;
- dividend payments related to Italian companies;
- stock splits/reverse stock splits;
- changes in capital;

- conversions;
- listings and de-listings;
- bonus shares allocation;
- press releases focused on securities updates.

The first page of all the Market Notices are based on the same semi-structured format, which contains the following elements:

1. the identification number of the notice;
2. the publication date;
3. the market exchange;
4. the sender;
5. the entities subject to the notice;
6. the subject, i.e. a brief description of the notice content.

Figure 1. An example of a Borsa Italiana Market Notice



Borsa Italiana

AVVISO n.10	02 Gennaio 2019	MTA

Mittente del comunicato : CHL

Societa' oggetto dell'Avviso : CHL

Oggetto : CHL S.p.A.: Informazioni ai sensi dell'art. 154-ter, comma 7, del D.Lgs. 58/98

In practice, the first semi-structured page of the document most of the times bears enough details for a data manager to understand the relevance of a Market Notice. However, when the subject is not very specific[5], the data manager has to skim through the whole notice in order to have a full understanding as to whether it should be selected for further processing. If unsure, the approach adopted is conservative and the notice is selected (positive bias) under the assumption that not collecting a useful notice is worse than selecting a non-useful notice.

[5] For instance, press releases may contain information regarding the quarterly earnings of a company (not of interest for the Securities Database) or the results of an offer of change in capital (of interest).

3 Dataset and feature extraction

Our dataset comprises the notices collected and labelled during 2019. In particular, it contains information on the subject (see below), stock market exchange and the target variable – i.e. a "yes"/"no" label defined by the data managers depending on whether the notice can be used for the information extraction. In our dataset, the target variable assumes the label "yes" in 4,602 notices (17% of total) and "no" in 21,889 cases (83%). The subject of a notice is a short, unstructured text. Textual raw data cannot be an input for most algorithms as they generally work with numerical feature vectors. We relied on scikit-learn (Pedregosa et al., 2011) for the feature extraction and the models estimation. We use a bag-of-words approach whereby texts are described by word occurrences and obtain feature vectors ignoring the order of the words in the subject. A bag-of-words vectorization of a text proceeds in three steps:

- tokenizing the text by assigning an integer to each token; in our case we pick words as tokens;
- counting the occurrences of tokens for each observation;
- normalizing, if necessary, in order to weight the words that occur too often (i.e. prepositions), with diminishing importance. In the following, we will skip the normalization step[6]. In the Appendix we show that the results of the application of the Term Frequency – Inverse Document Frequency (TF-IDF) transformation are similar.

Each individual token frequency, or count in absence of normalization, becomes a feature. With this approach, the subjects of the notices are represented by a matrix with one row per subject and one column per word occurring in all the subjects.

In order to partially preserve some of the local ordering information, we extract also the bi-grams (couples of ordered words) as tokens in addition to the 1-grams (individual words). Finally, we remove the tokens with less than a threshold out of 10 occurrences[7] among all the notices, obtaining a vocabulary of 2,657 elements[8]. Thus, the subject of each notice becomes a numeric vector of length 2,657.

[6] In scikit-learn, it means that we are using a simple count vectorizer.

[7] We also tried to change the threshold. The results are similar in the range [5, 20] and worse elsewhere.

[8] In this case the elements are words (uni-grams) and bi-grams occurring more than 10 times.

4 Classification models

Formally, our problem can be described through a binary classification model predicting a categorical variable Y taking on two values – *Yes=1* and *No=0* – depending on whether the notice is of interest or not. Given the features X, we look for a function f such that $Y \approx f(X)$ where $Y \in \{0,1\}$.

In order to estimate f one can consider various suitable types of parametric and non-parametric classification models. Each of the next three subsections illustrates a model then in Section 5 the best method to solve the notice selection problem is identified on the basis of its empirical performance.

4.1 Logistic regression

Instead of directly finding function f, logistic regression is used to model the probability that Y=1 ("Yes"), i.e. $P(Y = 1|X)$; then, given any value of the input data, we can predict P. The accuracy of the model is minimized, on average, by the *Bayes classifier* which assigns each observation to the most likely class, given its predictor values: in this simple binary case, we classify the notice as of our interest if $P(X) > 0.5$.

Logistic regression is stated as the optimization problem of minimizing the following cost function:

$$\sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1)$$

In our case we used a logistic regression adding l_2 regularization, weighted using the parameter λ:

$$\min_{w,c} \frac{1}{2} w^T w + \lambda \sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1)$$

The regularization term shrinks the estimated coefficients towards zero and, if the parameter λ is properly set, it improves out-of-sample prediction accuracy. Regularization allows to train models on data sets with many features with respect to the number of observations without severe over-fitting, essentially by limiting the model complexity. Then, the problem of determining the optimal model complexity becomes that of finding a good value of the regularization coefficient λ.

4.2 Naive Bayes

Naive Bayes methods are based on the application of Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable:

$$P(Y|X_1, X_2, \dots, X_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

We look for Y such that $P(Y|X_1, X_2, \dots, X_n)$ is maximum. Thus:

$$f(X) = \underset{y}{\operatorname{argmax}} \left(\frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(Y|X_1, X_2, \dots, X_n)} \right) = \underset{y}{\operatorname{argmax}} \left(P(y) \prod_{i=1}^{n} P(x_i \mid y) \right)$$

Despite the strong naive assumption, Naive Bayes classifiers have shown good results in many real-world applications, including the text classification task (Wang and Manning, 2012).

4.3 Random forest

Random forest (Breiman, 2001) is a non-parametric ensemble model based on decision trees. The training data set is repeatedly sampled in a new training set. For every set a decision tree is trained. In order to reduce the correlation among the decision trees, each time a split in a tree is considered, a random sample of *m* predictors is chosen as candidates from the full set of *p* predictors.

5 Model estimation

In order to have an efficient and systematic approach we follow Ng (2018); in particular, we

- establish a single performance metric,
- develop a first model in order to perform an error analysis,
- analyze the misclassification errors of the model in order to assess the quality of the dataset and find the most important ways to improve the model performance,
- evaluate the data manager performance on this task,
- estimate a new enhanced model.

5.1 Single performance metric

Having a single-number evaluation metric of the performance of an individual model is important in order to compare and easily choose among alternative models, as well as to tune its hyper-parameters. In this case the choice is straightforward since the business owner of the process is able to assess the cost of a misclassification. The data managers that manually select the notices consider a false negative ("*a notice that must be selected but that is not*") a more serious error than a false positive ("*a notice that is selected although it does not bear any relevant information*"). We tried to quantify the cost relationship between the two error types with the support of the business owner; the estimated cost of a false negative turned out to be ten times higher than a false positive.

We thus use a weighted version of the accuracy metric. A_w is defined as follows:

$$A_w = \frac{\sum_{i=1}^{N} w_i \mathbb{1}(y_i = \hat{y}_i)}{\sum_{i=1}^{N} w_i} \qquad \text{where: } w_i = \begin{cases} 10 \text{ if } y_i = 1 \\ 1 \text{ if } y_i = 0 \end{cases}$$

with y_i being the real value of the i-th observation and $\hat{y}_i$ its predicted value.

5.2 First models

The dataset is not large enough to introduce complex models. Moreover, as we will show below we can obtain good results by estimating simple models as well. During this phase, we want to focus on the following two goals:

- understanding if the classification task can be tackled using the above mentioned methods;
- obtaining a first prediction to analyze the misclassified observations, thus finding the areas where to improve and check the quality of the labelled data set.

We try the following methods:

- naive Bayes with smoothing parameter equal to 1;
- logistic regression with λ=1;
- random forest with 1000 trees.

The subjects of the notices are vectorized using the number of occurrences and removing the words that appear less than 10 times in the whole training set. The dataset is balanced with a random oversampling of the "yes" notices. Finally, the dataset is randomly split into a "training" and a "test set", representing, respectively, the 85 and the 15% of the 2019 Market Notices.

The weighted accuracy of the estimated models on the test set shows good results among the models. It is worth remarking how the models obtain good performances among others performance metrics.

Table 1. First models performances
(percentages)

Performance metric	Naive Bayes	Logistic regression	Random forest
Weighted accuracy	89,0	**91,6**	90,8
For reference			
Accuracy	*91,9*	*94,0*	***94,7***
Precision	*69,5*	*76,0*	***80,0***
Recall	*86,8*	***89,9***	*88,1*
F1-score	*77,2*	*82,5*	***83,9***

According to the chosen performance measure, the logistic regression is slightly better than the random forest and clearly outperforms the Naive Bayes.

5.3 Error analysis and data managers performance assessment

After a first estimation of the three models, we can perform an errors analysis of the predicted values of the models on the test set, an important tool for improving the overall performance of a machine learning model. This analysis might reveal patterns to improve the model or classification mistakes of the target variable, something quite common to many datasets. Indeed, it is quite often the case that investing in the improvement of the quality of the dataset is a more fruitful strategy than tweaking and optimizing some parameters, especially when the accuracy of the model is already high.

The errors in our dataset can be classified into two categories:

- systematic errors: they might occur if a data manager systematically misclassifies a specific category of notices or when new fields are added to the Securities Database; in the latter case, a type of notice that previously had not been considered now becomes relevant.
- random errors: notices randomly mislabeled by the data managers.

The analysis of the misclassified label gives us useful insights.

First, we find some useful features that is worth adding to the dataset:

- the presence of an ISIN[9] code in the free text of the notice;
- the nationality and the type of the entity the notice refers to;
- the presence of the Italian word *stacco*[10] in the free text of the notice.

Second, in carrying out the error analysis we have also detected two types of systematic errors having a relatively high impact on the model evaluation; we have been able to identify the reasons of

[9] A notice mentioning an ISIN code is often of interest.

[10] "Data *stacco cedola"* is the Italian expression for "ex-dividend date". Notices containing these words are often of interest for our Securities Database.

these errors and, then, we have amended the whole dataset, including the training set, to improve its overall quality.

Once we have completed the error analysis, we can turn to the evaluation of the quality of notice selection performed by the data managers. Having to examine many notices at once, a data manager may misclassify a few of them. On the basis of the analysis of the actual errors made in the past, we estimate the data managers' notice selection weighted accuracy to be 96.6%.

5.4 Final model estimation

As reported in Table 1, the models estimated in section 5.2 are far from reaching data managers' performance. Further efforts are needed to improve the models in order to replace the manual procedure with an automatic approach.

In this section we describe how to improve the classification models introduced in Section 5.2 through the combined effect of two actions:

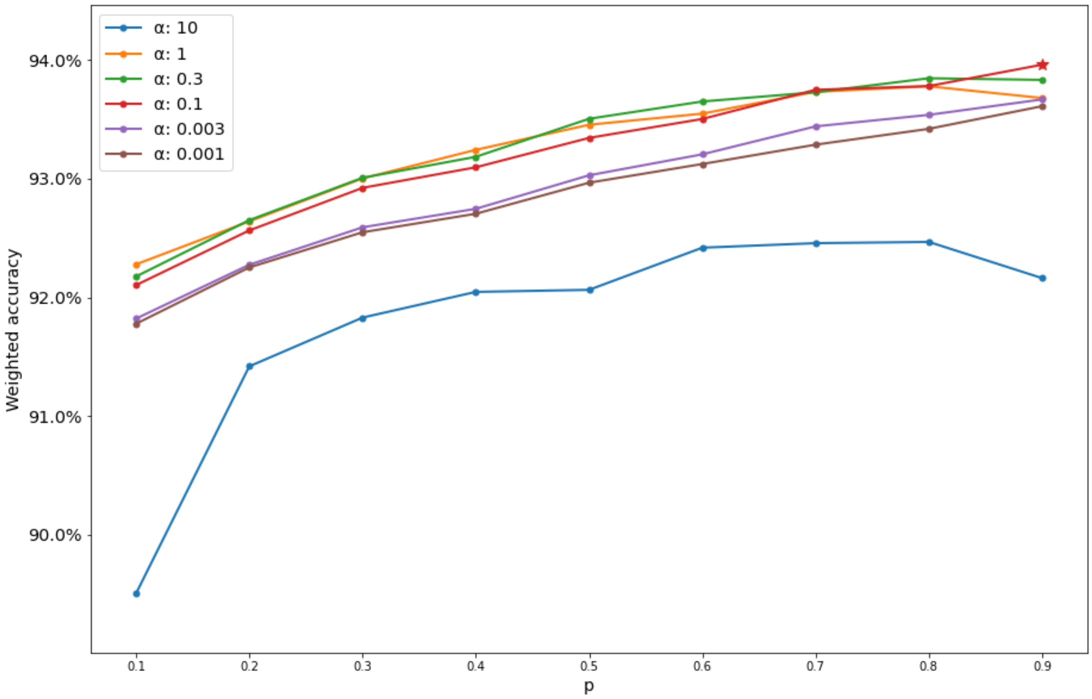
1 – the use of the dataset amended according to the results of the error analysis;

2 – the fine tuning of the models hyper-parameters in order to increase their performance in terms of weighted accuracy; we use a grid search with a 5-fold cross validation. For each model, detailed results are reported in the Appendix.

Regarding the Naïve Bayes model, we perform an optimization through the following parameters:

- α: smoothing hyper-parameter;
- p: prior probability of class "yes".

Figure 2. Weighted accuracy on the validation data of the Naïve Bayes model with different values of the hyper-parameters[11].



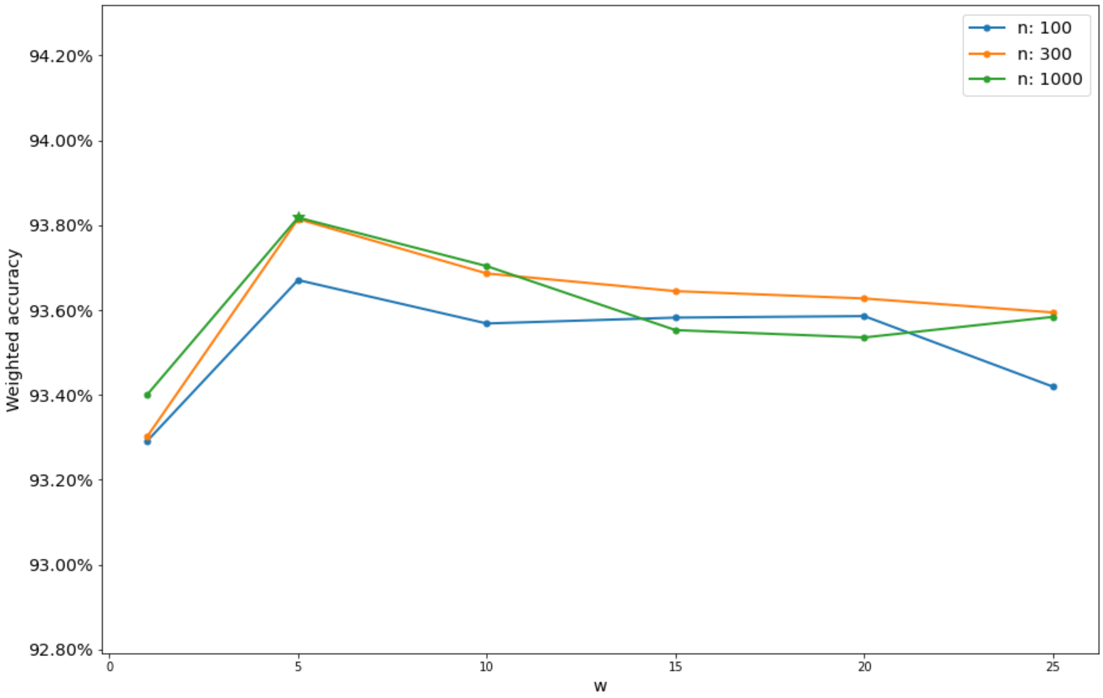
[11] See table A2 in the appendix.

The best obtained model hyper-parameters are α equal to 0.1 and p equal to 0.9.

Regarding the random forest classifier, we perform a grid search looking for a suitable value of the following hyper-parameters:

- w: weight hyper-parameter of class "yes". The other class weight is always 1;
- n: number of trees in the forest.

Figure 3. Weighted accuracy on the validation data of the random forest model with different values of the hyper-parameters[12].



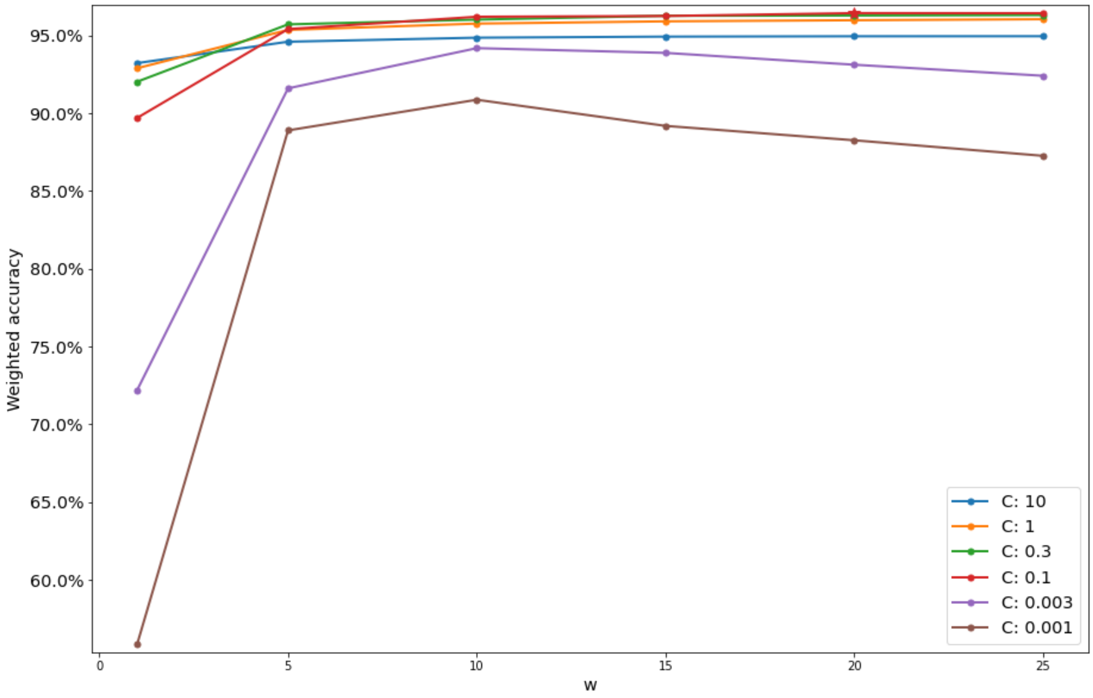
The best performing random forest model is the one with w equal to 5 and n equal to 300.

[12] See table A3 in the appendix.

Finally, for the estimation of the final logistic regression the tuned hyper-parameters are:

- w: weight hyper-parameter of class “yes”. The other class weight is always 1;
- C: regularization term. The lower C, the higher the l_2 regularization.

Figure 4. Weighted accuracy on the validation data of the logistic regression model with different values of the hyper-parameters.[13]



The final, selected model is the one with w equal to 20. As expected, the best w is higher than 1 because false negatives are more serious than false positives. The chosen C is 0.1. The regularization term is essential in this case not only for the improvement of the model performance, but also to make the estimation process succeed: with approximately 20 thousand examples and 3 thousand features, the estimation with no regularization[14] fails because the estimation algorithm could not converge.

Among the three optimized models, the logistic regression is the one with the highest weighted accuracy (97.2%)[15]. It is important to remark that its performance is even slightly better than the benchmark represented by the data managers.

[13] See table A4 in the appendix.

[14] In our notation, no regularization equals to $C \rightarrow \infty$.

[15] For reference, other performance measures are detailed in table A6 in the appendix.

6 Conclusions and future research

The paper describes how a text mining approach can be applied to improve the selection of the Market Notices published by *Borsa Italiana* that serve to update the Bank of Italy's Securities Database.

The importance of our analysis is twofold: first, it increases the overall quality of this crucial database; second, it enhances the efficiency of the whole statistical production process in that a highly time and resource consuming activity is carried out automatically.

More specifically, in the first part of the paper we have shown the benefits of the quick estimation of a set of supervised models and the analysis of the misclassified notices. Firstly, we are able to find new features that should be taken into account as input to further improve the models' accuracy. Secondly, we are able to enhance the quality of the labels of the dataset. A cleaner dataset means that the prediction models perform better.

Then, we show the importance, from the business point of view, of defining a robust measure to evaluate the quality of a model. Having a single performance measure speeds up the model selection and the hyper-parameters tweaking process.

Despite its simplicity, the final logistic regression model returns a weighted accuracy of 97%, which is extremely high; the estimated model is thus able to select the notices of interest with a slightly better performance than that of the data managers.

This model has already been implemented in the production environment and has been used on a daily basis to automatically select the relevant notices since April 2020. In the subsequent three months the data managers monitored the daily output of the model. At the end of the period, the very satisfactory performance of the automated selection process was confirmed and the model ultimately replaced the manual activity. This allowed data managers to focus on other tasks. It is also worth noting that such a procedure also presents the advantage of ensuring a uniform classification over time and across different data managers, hence removing the risk of a potential bias due to the subjective selection made by each individual data manager.

Having resolved the issue of the automated selection of the notices of interest, the paper leaves to future research the task of extracting, through an automatic procedure, the relevant information from individual notices, for later inclusion in the Securities Database.

References

Bishop C.M. (2007), “Pattern Recognition and Machine Learning”. Springer.

Breiman L. (2001), “Random forests”, Machine Learning, 45, 5-32.

Carboni A., Moro A. (2018), “Imputation techniques for the nationality of foreign shareholders in Italian firms, IFC Bulletins chapters”, Bank for International Settlements, External sector statistics: current issues and new challenges, volume 48.

Ng A. (2018), “Machine Learning Yearning”, deeplearning.ai.

Metsis V., Androutsopoulos I., Paliouras G. (2006), “Spam filtering with Naive Bayes – Which Naive Bayes?”, *3rd Conf. on Email and Anti-Spam (CEAS).*

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Bertrand T. (2011), “Scikit-learn: Machine Learning in Python”, *JMLR 12*, pp. 2825-2830.

Wang S., Manning C. (2012), “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification”, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 90-94.

Yang A., Liu X. (1999), “A Re-Examination of Text Categorization Methods”, *ACM SIGIR*, pp. 42-49.

Zambuto F., Arcuti S., Sabatini R., Zambuto D. (2021), “Application of classification algorithms in order to automatically manage confirmations to quality remarks”, Bank of Italy, Occasional Papers.

Appendix

Table A1. Dataset splits. Values: counts.

Year	Training	Test
2019	18543	3974

Table A2. Results of the cross-validation of the Naïve Bayes regression model. Values: weighted accuracy.

p α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.01	91.78%	92.25%	92.55%	92.71%	92.97%	93.12%	93.29%	93.42%	93.61%
0.03	91.82%	92.28%	92.59%	92.75%	93.03%	93.21%	93.44%	93.54%	93.67%
0.1	92.10%	92.56%	92.92%	93.10%	93.34%	93.50%	93.75%	93.78%	**93.96%**
0.3	92.18%	92.65%	93.01%	93.19%	93.51%	93.65%	93.73%	93.85%	93.83%
1	92.28%	92.64%	93.00%	93.24%	93.46%	93.55%	93.73%	93.78%	93.68%
10	89.50%	91.42%	91.83%	92.05%	92.06%	92.42%	92.46%	92.47%	92.16%

Table A3. Results of the cross-validation of the random forest model. Values: weighted accuracy.

w n	1	5	10	15	20	25
100	93.29%	93.67%	93.57%	93.58%	93.59%	93.42%
300	93.30%	93.81%	93.69%	93.64%	93.63%	93.59%

1000	93.40%	**93.82%**	93.70%	93.55%	93.54%	93.58%

Table A4. Results of the cross-validation of the logistic regression model. Values: weighted accuracy.

w *C*	1	5	10	15	20	25
0.01	55.84%	88.90%	90.87%	89.19%	88.26%	87.26%
0.03	72.19%	91.60%	94.19%	93.88%	93.12%	92.41%
0.1	89.70%	95.41%	96.21%	96.25%	**96.44%**	96.43%
0.3	92.03%	95.72%	96.02%	96.27%	96.29%	96.32%
1	92.89%	95.36%	95.77%	95.91%	95.98%	96.05%
10	93.23%	94.60%	94.86%	94.93%	94.95%	94.95%

Table A5. Results of the cross-validation of the logistic regression model with tf-idf as normalization technique. Values: weighted accuracy.

w *C*	1	5	10	15	20	25
0.01	32.27%	79.62%	81.82%	82.61%	83.75%	71.07%
0.03	58.75%	81.97%	85.53%	83.53%	84.57%	85.37%
0.1	82.96%	94.02%	95.69%	95.86%	95.85%	95.69%
0.3	87.76%	95.25%	96.15%	96.36%	96.36%	96.32%
1	91.03%	95.85%	96.22%	96.30%	96.40%	**96.44%**
10	92.94%	95.35%	95.67%	95.82%	95.86%	95.85%

Table A6. Final logistic regression model performance metrics computed on the test set.

Performance metric	Value
Weighted accuracy	97,2%
Accuracy	94,9%
Precision	78,1%
Recall	98,7%
F1-score	87,2%