Predicting incorrect labels: combining (noisy) existing data and text classification

Vera Oosterveen

Statistics Netherlands & Utrecht University

December 11, 2019

Abstract

Insert abstract :)

1 Introduction

National Statistical Institutes (NSI's) aim to produce official statistics to contribute to the public debate and policy development. Business statistics, such as statistics on turnovers and bankruptcies, give insights into different facets of an economy. These business statistics are often presented by industry, groupings of enterprises with similar economic activities. The classification system for economic activities that is used in the European Union is the NACE (Nomenclature statistique des activités économiques dans la Communauté Européenne). The NACE is a hierarchical classification and consists of four-digit codes. National Statistical Institutes (NSI's) may introduce a fifth digit, as Statistics Netherlands did in the *Standaard Bedrijfsindeling* (SBI).

The SBI codes for Dutch enterprises can be found in the general business register (GBR). An enterprise is described by Statistics Netherlands as a 'statistical unit' and can be a combination of multiple legal units. When a 'legal unit' is registered at the chamber of commerce, the SBI code is determined.

However, one may question the quality of the SBI codes found in the GBR. Christensen (2008) found that, considering all NACE categories, 18 percent of a sample of Danish firms with at least 5 employees were misclassified. There are no recent quality evaluations available for all NACE codes in the Netherlands. But, for the Dutch car trade sector, with 9 sub-categories, Van Delden et al. (2016a) found that small enterprises (fewer than 10 employees) in certain sectors had a probability of 50 percent of misclassification.

The SBI codes in the GBR might be incorrect for several reasons. An incorrect code can be appointed either during registration, if the structure of an enterprise changes over time, or if there changes in the activity of an enterprise. Changes in economic activity are rarely registered (Christensen, 2008; Van Delden et al., 2016a) since this is not obligatory. The correctness of the SBI codes is of great importance for the quality of the business statistics. Misclassifications in the SBI codes could lead to inaccurate, biased output statistics.

To avoid costly and time-consuming manual checks, NSI's can use additional data sources to validate or supplement the current data sources. Previous research tried to predict the correct economic activity code using text classification approaches. Textual descriptions of the activity (Caterini, 2018) or website texts (Roelands et al., 2017; Berardi et al., 2015; Kuhnemann, 2019) were deployed to automate the classification of enterprises using machine learning techniques. However, these attempts resulted in low accuracy or high misclassification rates.

Websites are not necessarily created to describe an economic activity, but are aimed at branding, selling products or services. This has two consequences. Firstly, it complicates designing an accurate text classifier. Secondly, the observed label in the GBR might be more informative than the website text. Solely relying on text classification algorithms might not be the optimal strategy to deal with the possibly 'noisy' labels in the GBR.

Instead of predicting the correct label for each case, the focus in this study is on estimating which cases are labeled incorrectly. If the incorrectly labeled cases can be pointed out, these can be corrected with manual editing.

The goal of the current research is therefore to create a method to point out the incorrectly labeled cases. To find these erroneous cases both the observed label and a predicted label, based on a machine learning algorithm, are included in the proposed method. Including two labels for every enterprise leads to the introduction of a variable that indicates whether there is an agreement between those two labels. An agreement between the two labels would lead to more trust in the labels, i.e. a lower probability of being erroneous for that enterprise.

To point out the incorrectly labeled cases, we include the probability of an error. Other studies that tried to locate erroneous cases assumed a random error-pattern (Sigurdsson et al., 2002; Eskin, 2002; Guarnera & Zio, 2013). An average probability of being incorrect was included in the models to account for the noise. However, we believe the probability of an error might depend on characteristics of the enterprise. Van Delden et al. (2016a) found that enterprise characteristics, such as size class and the number of legal units, influence the probability of an classification error. Instead of assuming a random-error pattern, the current research will account for non-random error probabilities.

2 Related work

The effects of noisy labels on commonly-used algorithms (e.g. Support Vector Machines (SVMs), logistic regression, k-nearest neighbours) are well studied (Pechenizkiy et al., 2006; Nettleton et al., 2010). In general, the performance of an algorithm decreases when labels in the training data are noisy. However, some algorithms are less influenced by label noise than others (Frénay & Verleysen, 2014). Furthermore, we can distinguish two main approaches when dealing with label noise (Frénay & Verleysen, 2014): filtering approaches and incorporating a model for the label noise within an algorithm.

With a filtering approach, the label noise is handled in the data preprocessing stage. Labels that are suspected to be noisy are simply removed from the training set (Brodley & Friedl, 1999). Removing the suspected noisy cases from the training set possibly leads to higher classification accuracy, but filtering is problematic when training data is scarce. Also, good classifiers are needed to detect misclassified cases, and learning from noisy labels results in weak classifiers: a chicken-and-egg dilemma as pointed out by Angelova et al. (2005).

Label noise robust variations of common classifiers have been proposed for, among others, logistic regression (Bootkrajang & Kaban, 2012; Rantalainen & Holmes, 2011)), SVMs (Stempfel & Ralaivola, 2009), and neural-networks (Sigurdsson et al., 2002; Sukhbaatar & Fergus, 2014)). Or, with a Bayesian approach, a prior on the mislabeling probabilities is included in the model (see Frénay & Verleysen (2014) for references).

Some approaches for incorporating label noise in a model are interesting in particular, since these are more closely related to the current research. These approaches assume a mixture model. By imposing a mixture model, the population is assumed to consist of multiple subpopulations. A different distribution can thereby be modeled for each subpopulation. A mixture model can be applied to deal with label noise, or to find anomalies or outliers in a dataset.

Guarnera & Zio (2013); Eskin (2002) applied such a mixture model. They assumed that the erroneous data and the error-free data came from different distributions. Detecting the noisy cases is then identical to finding out to which distribution a case belongs.

The current study has a similar motivation as the earlier studies of Guarnera & Zio (2013); Eskin (2002), but without assuming a specific distribution for the erroneous and error-free data. In the current research, the erroneous cases will be selected based on probability estimates, which is explained in depth in Section 3.2. In addition, the previous research on using website texts to predict the SBI codes will be expanded. As Nigam et al. (2000) showed, a classifier can be improved when both labeled and unlabeled data are used for training. By assuming that it is unknown whether an enterprise has the correct SBI code or not, these cases are treated as unlabeled. Therefore, we propose a method where the machine learning model used to predict SBI codes will be iteratively updated by including instances in the training set that are considered correct.

In summary.....

This paper proceeds as follows. In Section 3, we describe the proposed methodology to find erroneous labeled cases. In turn, we will elaborate on the different experiments that we ran to test the method in Section 3.4. Lastly, results and conclusions are given.

3 Methodology

In this research, the following aspects are combined in the proposed method: I don't really know how to open this section...

1. The predicted label, based on an enterprises' website,
2. The observed label, as can be found in the GBR,
3. The agreement between the observed label and the predicted label,
4. The probability of an error.

The predicted label is obtained by fitting a machine learning model, the method is not restricted to a specific classifier. The website text serves as the input for the machine learning model. URLs for websites are obtained from the GBR, or via an external party when a URL was missing in the GBR. Only the main page of each website is scraped. The text from the main page is tokenized, e.g. split into smaller parts that can serve as features. Moreover, the observed label, and the agreement between the observed label and the predicted label will be included in the proposed method.

We assume that characteristics of an enterprise affect the probability of an error. A logistic regression is modeled to relate characteristics of enterprises to the probability of an error. These characteristics are *the size class*, *the number of legal units that together form the statistical unit*, *the number of different activities in the legal units*, *an enterprises' age*, *the SBI codes for the* *main and secondary economic activity of an enterprise.* These characteristics are all available in the GBR.

First of all, all mathematical notation will be introduced in the following paragraph. In Section 3.2 we explain how we derive the probability of an erroneous label for each case, based on the components mentioned above. These probabilities are estimated with an EM algorithm, as explained in Section 3.3.

3.1 Notation

Here, we introduce the necessary notation.

The predicted label. Let $y_i^* = k$ denote the SBI code predicted by the machine learning model for unit i. The input for the machine learning model are the features of the website text, denoted by ζ_i. The parameters of the machine learning model are denoted by θ^M.

The observed label. The SBI code as found in the GBR is denoted by $\hat{y}_i = h$.

The agreement. The predicted code $y_i^* = k$ and the observed code $\hat{y}_i = h$ are compared. Let a denote this agreement, with $a \in \{0, 1\}$. Let $a = 1$ if the two codes agree and $a = 0$ otherwise.

The probability of an error. Finally, we introduce a latent variable, z_i, that represents whether a unit's observed code in the GBR is correct or erroneous. Let $z_i = 0$ if the code in the GBR is the true code and $z_i = 1$ when the code in the GBR is incorrect. The variable z_i can be modeled as a function of enterprise characteristics using logistic regression. The probability of an erroneous label, $P(z_i = 1)$, may depend on characteristics such as the size class or the number of legal units (Van Delden et al., 2016a). Enterprise characteristics are denoted by u_i. Let θ^R denote the parameters of the logistic regression model. The full set of model parameters is therefore $\theta \in \{\theta^M, \theta^R\}$.

Throughout this paper we assume that each unit has a single true (unknown) NACE code which is considered as fixed. That is, the NACE code that is derived when all economic activity information is available without error and the rules to derive a NACE code is applied correctly.

In the next paragraph we explain how a probability of an erroneous case is estimated.

3.2 Deriving the probability of being erroneous

We are interested in the probability that a unit is incorrectly labeled, given the agreement between observed and predicted code, enterprise characteristics, features derived from the website text, and the model parameters. This is defined as τ_i:

$$\tau_i(\hat{a}_i = a) \equiv P(z_i = 1 | \hat{a}_i = a, \hat{y}_i = h, \zeta_i, u_i, \theta). \tag{1}$$

To find a probability for Equation 1, we use *Bayes' Theorem* with multiple conditions:

$$P(A|B, C) = \frac{P(A|C)P(B|A, C)}{P(A|C)P(B|A, C) + P(A^C|C)P(B|A^C, C)}.$$

Substituting elements of Equation 1 into Bayes' Theorem gives:

$$\tau_i(\hat{a}_i = 1) = P(z_i = 1|\hat{a}_i = 1, \hat{y} = h, \zeta_i, u_i, \theta) = \tag{2}$$

$$\frac{P(z_i = 1|\hat{y}_i = h, \zeta_i, u_i, \theta)P(\hat{a}_i = 1|z_i = 1, \hat{y}_i = h, \zeta_i, u_i, \theta)}{\sum_{z=0,1} P(z_i = z|\hat{y}_i = h, \zeta_i, u_i, \theta)P(\hat{a}_i = 1|z_i = z, \hat{y}_i = h, \zeta_i, u_i, \theta)}. \tag{3}$$

The first term in Equation 3, $P(z_i = 1|\hat{y}_i = h, \zeta_i, u_i, \theta)$, represents the expected probability of an erroneous label in the GBR, given the observed label, features of the website, enterprise characteristics, and the model parameters. We assume that the probability of an erroneous label in the GBR does not directly depend on the features of the website or on the parameters of the machine learning model. However, the website text (and therefore the features derived from the website text) can indirectly influence the probability of an error. That is, units for which it is more difficult to derive the SBI code may also have more vague website texts, or websites with less text. The indirect effect of the website text can be captured in u_i; we include for instance the number of words in the website text in u_i. Given these assumptions, we obtain the probability of an error, which we denote as π_i:

$$\pi_i \equiv P(z_i = 1|\hat{y}_i = h, \zeta_i, u_i, \theta = P(z_i = 1|\hat{y}_i = h, u_i, \theta^R). \tag{4}$$

The term $P(\hat{a}_i = 1|z_i = 1, \hat{y}_i = h, \zeta_i, u_i, \theta)$ in Equation 3 stands for the probability of agreement between the observed and predicted codes, given that the label is incorrect. Let us define this probability γ and assume this is a small probability. There is a small probability that a SBI code changes in a given period, when the code before and after was incorrect (Van Delden et al., 2016b). Therefore, an initial estimate for γ can be obtained as the fraction of cases where the predicted NACE code y_i^* and the observed NACE code $\hat{y}_i$ agree.

$$\hat{\gamma} = \sum_i \tau_i \hat{a}_i / \sum_i \tau_i. \tag{5}$$

Furthermore, we consider the probability of an agreement, given that the label is correct: $P(\hat{a}_i = 1|z_i = 0, \hat{y}_i = h, \zeta_i, u_i, \theta)$. This probability can be estimated for the probability that $y_i^* = h$ based on the fitted machine learning model. Therefore, $P(\hat{a}_i = 1|z_i = 0, \hat{y}_i = h, \zeta_i, u_i, \theta) = P(y_i^* = h|z_i = 0, \zeta_i, u_i, \theta^M)$.

Equation 3 can now be rewritten as:

$$\frac{\pi_i \gamma}{(1 - \pi_i)P(y_i^* = h|z_i = 0, \zeta_i, u_i, \theta^M) + \pi_i \gamma}. \tag{6}$$

The equation above implies that when γ and π_i are small, and the machine learning model predicts the right SBI code with a high probability, the probability of an erroneous case is low.

Likewise, there is the probability of an erroneous label, τ_i, when there is disagreement between the observed label, $\hat{y}_i$, and the predicted label, y_i^*.

$$\tau_i(\hat{a}_i = 0) = (z_i = 1|\hat{a}_i = 0, \hat{y} = h, \zeta_i, u_i, \theta) = \tag{7}$$

$$\frac{P(z_i = 1|\hat{y}_i = h, \zeta_i, u_i, \theta)P(\hat{a}_i = 0|z_i = 1, \hat{y}_i = h, \zeta_i, u_i, \theta)}{\sum_{z=0,1} P(z_i = z|\hat{y}_i = h, \zeta_i, u_i, \theta)P(\hat{a}_i = 1|z_i = z, \hat{y}_i = h, \zeta_i, u_i, \theta)} = \tag{8}$$

$$\frac{\pi_i P(\hat{a}_i = 0|z_i = 1, \hat{y}_i = h, \zeta_i, u_i, \theta)}{(1-\pi_i)P(\hat{a}_i = 0|z_i = 0, \zeta_i, u_i, \theta) + \pi_i P(\hat{a}_i = 0|z_i = 1, \zeta_i, u_i, \theta)}. \tag{9}$$

The term $P(\hat{a}_i = 0|z_i = 1, \hat{y}_i = h, \zeta_i, u_i, \theta)$ represents the probability that the observed SBI code and predicted code disagree, given that the label is incorrect. This can be estimated by $1 - \gamma$.

The term $P(\hat{a}_i = 0|z_i = 0, \zeta_i, u_i, \theta)$ represents the probability of disagreement, given that the label is correct. This probability is equal to $1 - P(\hat{a}_i = 0|z_i = 0, \zeta_i, u_i, \theta) = 1 - P(y_i^* = h|z_i = 0, \zeta_i, u_i, \theta^M)$.

Equation 9 can be rewritten as:

$$\frac{\pi_i(1-\gamma)}{(1-\pi_i)(1 - P(y_i^* = h|z_i = 0, \zeta_i, \theta^M)) + \pi_i(1-\gamma)}. \tag{10}$$

This implies that the probability of an erroneous label is large when according to the machine learning model it is likely that the SBI code of unit i differs from the code that is observed in the GBR.

In summary, Equation 6 or Equation 10 is used to estimate the probability of an case being erroneous. If this probability is greater than 0.5 for a unit, this unit is considered erroneous and we set $z_i = 1$. If the probability of being erroneous is smaller than 0.5, a unit is seen as correct and, therefore, $z_i = 0$. In the following paragraph, it is explained how the model parameters are estimated.

3.3 Estimation using the EM algorithm

The model parameters are estimated with the Expectation Maximization (EM) algorithm. The EM algorithm can be used for maximum likelihood estimation in incomplete data problems (Dempster et al., 1977). In our case, the data are seen as incomplete because the values for the latent variable z_i are missing for a part of the data. The EM algorithm provides us with a way of estimating the values for z_i.

The EM algorithm consists of an Expectation step (E-step) and a Maximization step (M-step). Firstly, the algorithm estimates model parameters for the machine learning model, θ^M, and the logistic regression model, θ^R, based on the available information of z_i. In the subsequent E-step the values of τ_i are estimated given the model parameters θ. When iterating, the models are retrained in the M-step. A difference with the previous M-step is that the estimates for τ_i serve as weights while training the machine learning model. In other words, τ_i represents the contribution of a unit to the parameter estimation in the machine learning model.

To initiate the model, ...

The process is presented in Table 1.

Table 1: The EM algorithm.

Initialize	Estimate τ_i for all units in the population
M-step	• Train a machine learning model • Fit a logistic regression model • Estimate γ
E-step	Re-estimate τ_i according to Equation 6 and 10. If $\tau_i > .5$ set $z_i = 1$ and $z_i = 0$ otherwise
Iterate	Repeat the E- and M-step until the changes in τ_i are smaller than 1e-9.

3.4 Experiments

This section describes the experimental evaluation of the method. The goal is to investigate under which circumstances the method is (not) able to find the erroneous cases. Section 3.4.1 describes how a synthethic dataset is created. The different experimental settings are explained in Section 3.4.2. Lastly, Section 3.4.3 elaborates on the evaluations metrics used to select the optimal starting values and to select the best performing model.

3.4.1 Synthetic dataset

The synthethic dataset is based on the real-life data from the GBR. A limited number of SBI codes are selected for the experiments. In total, 25 classes were chosen, some of which are combinations of different SBI codes. One prerequisite was that all classes were large, containing at least 400 enterprises with a known URL in the GBR. On the one hand, within the 25 classes were homogeneous classes that are unrelated to other classes, such as hairdressers. On the other hand, more overlapping, related classes are included, such as wholesale of clothes and shops selling clothes. Both more similar and more unrelated classes are included to see if the model can distinguish between these classes when the labels are manipulated (see Section 3.4.2.

A filter is applied to the 25 selected classes. The objective was to end up with those cases of which the SBI codes are almost certainly correct. For filtering, three different machine learning models are applied to the dataset. These machine learning models are trained on the large enterprises. A website is included in our new dataset G if three predicted labels are all in correspondence with the observed label in the GBR. Next, errors in the SBI code are introduced in some units in set G. The set with introduced errors is called the synthetic dataset H. The true labels for the manipulated cases are known, the knowledge of the true labels can be used to evaluate the model performance.

Set H consists of three parts:

- gold set: this part contains solely true SBI codes, therefore all units in the gold set have $z_i = 0$. Conceptually, this represents the part of the GBR that is checked manually. This part is initially used as the input for the machine learning model.

- false set: for this part all labels are set to erroneous labels, $z_i = 1$. This false set is introduced to serve as input for the logistic regression model, since a logistic regression model needs both incorrect and correct examples to be able to estimate parameters.

  The union of the gold set and the false set form the labeled set.

- noisy set: part of the labels in the noisy set are deliberately changed into a false label. However, for this part the value for z_i is missing. By iterating the EM algorithm, values for z_i will be estimated.

Figure 1 depicts how the synthethic set H is obtained.

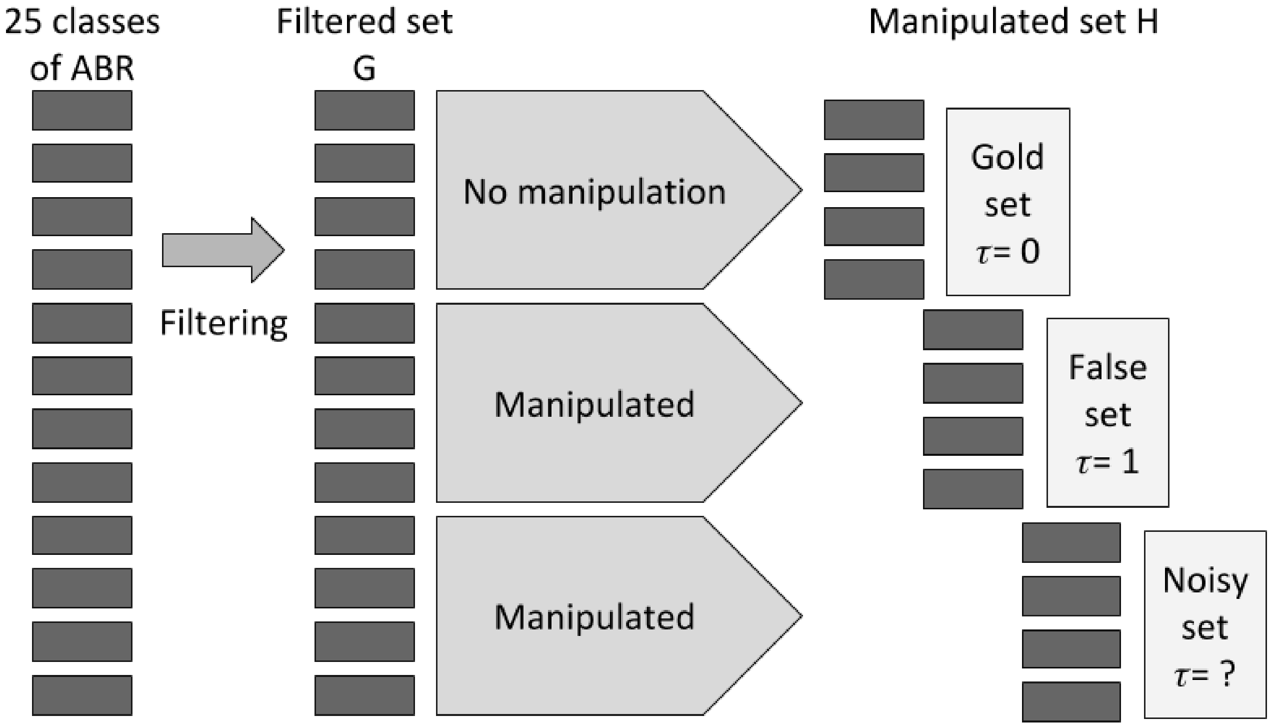

Figure 1: Obtaining set G and set H.

3.4.2 Experimental settings

We aim to test the performance of the algorithm in different settings. The manipulations in set H are varied with respect to: 1) how representative the error-patterns in the labeled set are for the error-pattern in the noisy set, 2) the size of the labeled set compared to the noisy set, 3) the percentage of errors in the noisy set, and 4) the type of errors.

Firstly, the error patterns in the labeled and noisy set could be generated such that they are similar, i.e. cases with certain characteristics have errors. This is referred to as the 'informed labeled set'. In this case, the model should be able to point out the errors in the noisy set. We also generate errors in the labeled set using only part of the enterprise characteristics, the 'limited labeled set'. Also, we can introduce random errors in the labeled set: 'random labeled set'.

Two types of errors can be introduced: obvious and subtle errors. When an observed SBI code is manipulated, one could vary how related the new, manipulated label is to the observed label. A SBI code that is changed to a non-related SBI code is an obvious error. A SBI code that is changed to a more similar, related SBI code is a subtle error.

Moreover, the proposed model, including a machine learning model and a logistic regression model, can be simplified. Fewer predictors can be used in the logistic regression, or the logistic regression could be replaced by simply including an average error probability.

Table 2 gives an overview of the experimental settings.

Table 2: Experimental settings.

Dataset	
Representativity error-pattern	informed labeled set, limited labeled set, random labeled set
Size labeled set	?%, ?%, ?%
Percentage of errors	5%, 10%, 20%
Error type	obvious, subtle
Model	
Modeling π_i	logistic regression to estimate π_i, logistic regression (limited predictors) to estimate π_i, use a fixed, average error probability π

3.4.3 Evaluation metrics

3.4.4 Log-likelihood

In order to select the optimal starting values for the EM algorithm, we use the log-likelihood of the labeled set. Within the labeled set, there is a part with $z_i = 1$ and a part with $z_i = 0$. For the part with $z_i = 1$ we are interested in the estimated probabilities for τ_i. These probabilities are ideally close to 1. For the part of the labeled set where $z_i = 0$, we do know the true SBI code for unit i. We want to extract the probability that the machine learning model predicts this true SBI code. Preferably, this probability is close to 1. The log-likelihood is then given by:

$$log \sum_i z_i \tau_i + \sum_k (1 - z_i) I(y_i = k) \hat{P}(y_i = k) \tag{11}$$

3.4.5 Cross-entropy loss

The cross-entropy loss, also called log loss, is used to measure the performance of the different models considered in the experiments. The log loss is defined on probability estimates and, therefore, takes the uncertainty of the predicted labels into account. The further the value of the log loss deviates from 0, the further away the predicted probability is from the actual class. The log loss can be computed for binary classification problems with the `log_loss` function in the `scikit-learn` package (Pedregosa et al., 2011).

Next, the results of the experiments are discussed. After that, the model is applied to real data and the outcomes are discussed.

References

Angelova, A., Abu-Mostafam, Y., & Perona, P. (2005, June). Pruning training sets for learning of object categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 494-501. doi: 10.1109/CVPR.2005.283

Berardi, G., Esuli, A., Fagni, T., & Sebastiani, F. (2015). Classifying websites by industry sector: A study in feature design. In *Proceedings of the 30th annual acm symposium on applied computing* (pp. 1053–1059). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2695664.2695722 doi: 10.1145/2695664.2695722

Bootkrajang, J., & Kaban, A. (2012, September). Label-noise robust logistic regression and its applications. *Joint European conference on machine learning and knowledge discovery in databases*, 143-158.

Brodley, C. E., & Friedl, M. A. (1999, July). Identifying mislabeled training data. *Journal of artificial intelligence research*, *11*(1), 131–167. Retrieved from http://dl.acm.org/citation.cfm?id=3013545.3013548

Caterini, G. (2018). [DEM Working Papers]. (2018/09). Retrieved from https://ideas.repec.org/p/trn/utwprg/2018-09.html

Christensen, J. (2008, June). Questioning the precision of statistical classifications of industries. *DRUID Conference on Entrepreneurship and Innovation*, 17-20.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38. Retrieved from http://www.jstor.org/stable/2984875

Eskin, E. (2002, April). Detecting errors within a corpus using anomaly detection. *Proceedings of the 1st North American chaper of the Association for Computational Linguistics conference*, 148-153.

Frénay, B., & Verleysen, M. (2014, May). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 845-869. doi: 10.1109/TNNLS.2013.2292894

Guarnera, U., & Zio, M. (2013, December). A contamination model for selective editing. *Journal of official statistics*, *29*. doi: 10.2478/jos-2013-0039

Kuhnemann, H. (2019, May). *Web-based text mining approaches to the classification of economic activity.* (Unpublished Manuscript)

Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010, April). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, *33*(4), 275–306. Retrieved from https://doi.org/10.1007/s10462-010-9156-z doi: 10.1007/s10462-010-9156-z

Nigam, K., Mccallum, A. K., Thrun, S., & Mitchell, T. (2000, May). Text classification from labeled and unlabeled documents using em. *Machine Learning*, *39*(2), 103–134. Retrieved from https://doi.org/10.1023/A:1007692713085 doi: 10.1023/A:1007692713085

Pechenizkiy, M., Tsymbal, A., Puuronen, S., & Pechenizkiy, O. (2006, June). Class noise and supervised learning in medical domains: The effect of feature extraction. *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, 708-713.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rantalainen, M., & Holmes, C. (2011). Accounting for control mislabeling in case–control biomarker studies. *Journal of proteome research*, 5562-5567.

Roelands, M., Van Delden, A., & Windmeijer, D. (2017, November). Classifying businesses by economic activity using web-based text mining [Discussion Paper].

Sigurdsson, S., Larsen, J., Hansen, L., Philipsen, P., & Wulf, H. (2002, May). Outlier estimation and detection application to skin lesion classification. *IEEE International Conference on Acoustics Speech and Signal Processing*, *1*. doi: 10.1109/ICASSP.2002.5743975

Stempfel, G., & Ralaivola, L. (2009). Learning svms from sloppily labeled data. In C. Alippi, M. Polycarpou, C. Panayiotou, & G. Ellinas (Eds.), *Artificial neural networks – icann 2009* (pp. 884–893). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sukhbaatar, S., & Fergus, R. (2014, June). Learning from noisy labels with deep neural networks.

Van Delden, A., Scholtus, S., & Burger, J. (2016a, September). Accuracy of mixed-source statistics as affected by classification errors. *Journal of official statistics*, *32*, 619-642. doi: 10.1515/jos-2016-0032

Van Delden, A., Scholtus, S., & Burger, J. (2016b, November). Exploring the effect of time-related classification errors on the accuracy of growth rates in business statistics. In (p. 21-24).