Detecting incorrect labels in a business register using text classification

Vera Oosterveen
Statistics Netherlands & Utrecht University

December 15, 2019

Abstract

The abstract is included in a later stage.

1 Introduction

National statistical institutes aim to provide 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 contains four-digit codes. National statistical institutes may introduce a fifth digit, as Statistics Netherlands did in the *Standaard Bedrijfsindeling* (SBI).

The NACE 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 NACE code is determined.

However, one may question the quality of the NACE 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 50 percent probability of misclassification.

The NACE 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 are 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 NACE codes is of great importance for the quality of the business statistics. Misclassifications in the NACE codes could lead to inaccurate, biased output statistics.

To avoid costly and time-consuming manual checks, national statistical institutes can use additional data sources to validate or supplement the current data. Previous research tried to predict the economic activity codes 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 selling 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 handle the possibly 'noisy' labels in the NACE classification.

This research expands the previous work on predicting economic activity based on websites. We do this not by focusing on predicting correct economic activities for enterprises, but by focusing on identifying the incorrectly labeled enterprises. When a machine learning model is not able to predict the correct NACE class of an enterprise based on its website, this does not necessarily indicate that the current label of that enterprise is incorrect. Therefore, we consider the current observed NACE code in the GBR, in addition to a predicted economic activity based on a website.

Knowing which enterprises are likely to have an incorrect NACE code, or which industries have the largest fraction of errors, can contribute to the manual editing process. In the manual editing process, specific enterprises can be targeted or prioritized. The objective of this study is to develop an approach to point out incorrectly labeled cases in the GBR.

2 Related work

In the context of text classification and noisy labels, the focus in the literature has mainly been on achieving more accurate classification rates. 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 to handle 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 may lead to higher classification accuracy, but is problematic when training data is scarce. As pointed out by Angelova et al. (2005), good classifiers are needed to detect misclassified cases, and learning from noisy labels results in weak classifiers: a chicken-and-egg dilemma.

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; Biggio et al., 2011), 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 where we want to point out the noisy labels. A mixture model can be applied to deal with label noise, or to find anomalies or outliers in a dataset. By imposing a mixture model, the population is assumed to consist of multiple subpopulations. Different distributions can thereby be modeled for each subpopulation. Guarnera & Zio (2013) and 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 pointed out based on probability estimates, which is explained in depth in Section 3.2.

Typically, researchers assume a random error pattern for label noise. The reason most research is focused on random noise might be due to the simplicity of random noise (Bootkrajang, 2016). However, non-random noise may be more realistic and is encountered more often in real-world data. Only a few studies addressed the non-random noise. For example, Kolcz & Cormack (2009) showed that label noise was class-specific in email spam-filtering data. Certain genres of email were much more likely to confuse than others. In the area of speech recognition, Sarma & Palmer (2004) demonstrate that phonetic similarity between the correct word and the recognised word leads to more mislabeling.

In the current research we will further investigate the non-random label noise. Van Delden et al. (2016a) found that characteristics such as size class and the number of legal units influence the probability of a classification error. Therefore, we assume that the probability of a classification error depends on the characteristics of an enterprise.

This paper proceeds as follows. In Section 3, we describe the proposed methodology to find erroneous labeled cases in depth. In turn, we will elaborate on the different experiments that we conduct to test the method (Section 4).

3 Methodology

In order to point out the erroneous cases, we consider the following four aspects:

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. A non-random probability of an error.

The predicted label is obtained by fitting a machine learning model, such as a Naive Bayes or SVM. The website texts serve as 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. The main page is scraped from each website. The text from the main page is tokenized, i.e. split into smaller parts that can serve as features in a machine learning model.

Moreover, we take note of the observed label. Specifically, we are interested in the agreement between the observed label and the predicted label. An agreement between the two labels leads to more trust in the labels, i.e. a lower probability of an erroneous label. We assume that characteristics of an enterprise affect the probability of an error. With a logistic regression model we can relate characteristics of enterprises to the probability of an error. These enterprise characteristics are *the size class*, *the number of legal units that form the statistical unit*, *the number of different activities in the legal units*, *age of the enterprise*, *the NACE codes for the* *main and secondary economic activity of an enterprise.* The GBR contains these characteristics for all enterprises.

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 four aspects mentioned. These probabilities are estimated with an EM algorithm, as is explained in Section 3.3.

3.1 Notation

In this paragraph we introduce the necessary notation.

The predicted label. Let $y_i^* = k$ denote the NACE 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 NACE 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 error, $P(z_i = 1)$, may depend on enterprise characteristics, which 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 are applied correctly.

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 probability of an error given the observed label, features of the website, enterprise characteristics, and the model parameters. Before, we already stated why the enterprise characteristics influence the error-probability. Furthermore, we assume that the probability of an error 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 NACE 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 NACE 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 incorrect cases among all cases where the predicted NACE code y_i^* and the observed NACE code $\hat{y}_i$ agree:

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

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 NACE 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 NACE 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 NACE 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 a 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, $z_i = 0$.

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). As input, the model requires a labeled set. This is a set of data for which z_i is known. The part where $z_i = 0$ is the gold set: a set where the labels are correct almost certainly. In the real-world GBR data, the large, influential enterprises can be used as the gold set. The large enterprises are checked manually by Statistics Netherlands. The other part of the labeled set is the false set. Here, $z_i = 1$ and this (artificial) set represents the prior knowledge about the errors in the data. This part is needed to initiate the logistic regression model, since both correct and incorrect examples have to be provided to estimate parameters.

Table 1: The EM algorithm.

Input	A labeled set consisting of a gold set ($z_i = 0$) and a false set ($z_i = 1$)
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.

The EM algorithm needs starting values to initialize the model, meaning we provide an initial value for all units for z_i. These starting values can be derived in different ways, based on the labeled set. Then, the algorithm estimates model parameters for the machine learning model, θ^M, and the logistic regression model, θ^R, based on the values of z_i. Only the units with $z_i = 0$ are used for training the machine learning model. 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. The algorithm is presented in Table 1.

It is known that a different set of starting values might lead to a different solution (Shireman et al., 2017). Therefore, we will try several sets of starting values. 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 NACE code for unit i. We want to extract the probability that the machine learning model predicts this true NACE code. Preferably, this probability is close to 1. The log-likelihood is then given by:

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

where the indicator $I(y_i = k) = 1$ if the label is correct, and 0 otherwise, so that we sum the predicted probabilities of the correct labels.

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 4.1 describes how a synthetic dataset is created. The different experimental settings are explained in Section 4.2. Lastly, Section 4.3 elaborates on the evaluations metrics used to select the best performing model.

4.1 Synthetic dataset

The synthetic dataset is based on the real-world data from the GBR. A limited number of NACE codes are selected for the experiments. In total, 25 classes were chosen, some of which are combinations of different NACE 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 4.2.

A filter is applied to the 25 selected classes. The objective was to end up with those cases of which the NACE 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 NACE 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 NACE codes, therefore all units in the gold set have $z_i = 0$.
- 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, as explained before.
  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 synthetic set H is obtained.

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, 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’. An informed labeled set should make it easier for the model to find the erroneous cases in the noisy set. We also generate errors in the labeled set using only part of the enterprise

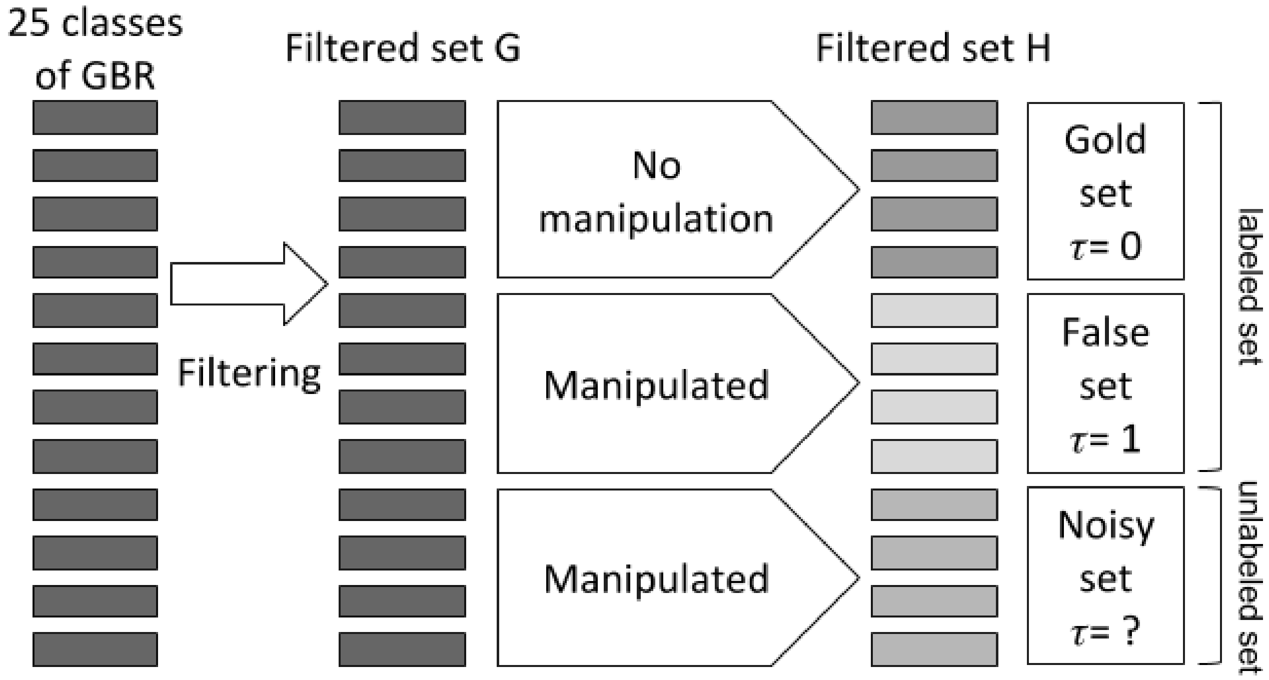

Figure 1: Obtaining set G and set H.

characteristics, the 'limited labeled set'. Also, we can introduce random errors in the labeled set: 'random labeled set'.

The size of the labeled set varies with 50, 100 or 150 cases per NACE code. The percentage of errors introduced in the noisy set varies with 5%, 10% or 15%. Two types of errors are introduced: obvious and subtle errors. When an observed NACE code is manipulated, we vary how related the new, manipulated label is to the observed label. A NACE code that is changed to a non-related NACE code is an obvious error. A NACE code that is changed to a more similar, related NACE 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. When an average error probability is included, we ignore that the noise might be non-random.

Table 2 gives an overview of the experimental settings.

4.3 Evaluation metrics

To evaluate the model performances, we investigate if the model indeed points out the manipulated cases in the noisy set as erroneous and estimates the non-manipulated cases in the noisy set as correct.

In a confusion matrix we find the number true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). The predicted value for z_i is compared with the observed value for z_i. With the confusion matrix, the precision, recall and accuracy are computed.

Furthermore, we look at the cross-entropy loss, also called log loss. The cross-entropy loss is defined on the probability estimates for z_i, that are given by τ_i. The cross-entropy loss takes the

Table 2: Experimental settings.

Dataset	
Representativity error-pattern	informed labeled set, limited labeled set, random labeled set
Size labeled set	50 per class, 100 per class, 150 per class
Percentage of errors	5%, 10%, 15%
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

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 loss function takes the average of all individual cross-entropies:

$$-\frac{1}{N}\sum_i z_i log(p_i) + (1 - z_i) log(1 - p_i), \tag{12}$$

where N is the total number of enterprises. Further, z_i is the true label and indicates either an erroneous case or a correct case. Here $p_i = P(z_i = 1) = \tau_i$, this denotes the estimated probability of being an erroneous case.

4.4 Results

Here, the results of the experiments will be presented. We will discuss how the model performed under different circumstances. The best performing model(s) will be applied to the real-world data.

5 Application to real-world data

The model(s) will be implemented on the enterprises in the GBR.

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

Biggio, B., Nelson, B., & Laskov, P. (2011, January). Support vector machines under adversarial label noise. *Journal of Machine Learning Research - Proceedings Track*, *20*, 97-112.

Bootkrajang, J. (2016). A generalised label noise model for classification in the presence of annotation errors. *Neurocomputing*, *192*, 61–71.

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

Kolcz, A., & Cormack, G. V. (2009). Genre-based decomposition of email class noise. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining* (pp. 427–436).

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

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.

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].

Sarma, A., & Palmer, D. D. (2004, May 2 - May 7). Context-based speech recognition error detection and correction. In *Proceedings of HLT-NAACL 2004: Short papers* (pp. 85–88). Boston, Massachusetts, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N04-4022

Shireman, E., Steinley, D., & Brusco, M. J. (2017, Feb). Examining the effect of initialization strategies on the performance of gaussian mixture modeling. *Behavior Research Methods*, *49*, 282–293. doi: 10.3758/s13428-015-0697-6

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. *arXiv preprint arXiv:1406.2080*, *2*(3), 4.

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).