# Who's Behind That Website? Classifying Websites by the Degree of Commercial Intent

Michael Färber[1], Benjamin Scheer[2], and Frederic Bartscherer[1]

[1]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
michael.faerber@kit.edu
[2]1&1 IONOS SE, Karlsruhe, Germany
benjaminscheer.bs@googlemail.com

**Abstract.** Web hosting companies strive to provide customised customer services and want to know the commercial intent of a website. Whether a website is run by an individual person, a company, a non-profit organisation, or a public institution constitutes a great challenge in website classification as website content might be sparse. In this paper, we present a novel approach for determining the commercial intent of websites by using both supervised and unsupervised machine learning algorithms. Based on a large real-world data set, we evaluate our model with respect to its effectiveness and efficiency and observe the best performance with a multilayer perceptron.

**Keywords:** document classification, web, text mining, machine learning

## 1 Introduction

Web hosting companies, such as 1&1 IONOS,[1] GoDaddy, and HostGator provide hosting services to millions of users ranging from individuals and non-profit organisations with no or little commercial intent to businesses with clear commercial intent. Apart from the size of the contract, web hosting companies are interested in cross-selling paid services with individual recommendations, such as SSL certificates or marketing services.

Websites can be clustered automatically given the readily available information on websites. Specifically, website classification can be considered as a document classification task, for which numerous methods have been proposed. However, no approach has been proposed to identify the commercial intent of websites on a large scale. In particular, applying document classification methods to websites is challenging as websites might have few words and coherent text structure compared to news articles, Wikipedia articles or research papers.

In this paper, we propose a novel approach to categorise websites based on its textual content into one of the following classes: *profit-oriented company*, *non-profit organisation*, *private website*, and *public institution*. To the best of

[1] This work was carried out in cooperation with the web hosting company 1&1 IONOS.

our knowledge, our approach is the first one which can identify the commercial intent of websites on a large scale and, thus, is particularly useful for web hosting companies that want to improve their customer experience. Based on a large data set covering over 30,000 websites, we apply both supervised and unsupervised machine learning methods and evaluate them with respect to effectiveness and efficiency.

Overall, our main contributions are as follows:

- We propose a new classification schema for commercial intent that applies to any website.
- We present several machine-learning-based methods for content-based website classification.
- We evaluate our approaches with a large data set of 30,000 websites in the German language.
- We publish both implementation and data sets for subsequent research.[2]

The remainder of the paper is structured as follows: In Sec. 2, we give an overview of related works and argue for an approach based on the commercial intent. In Sec. 3, we introduce our classification schema, followed by describing the data preparation steps and evaluation data set in Sec. 4. Our applied approach and the evaluation results can be found in Sec. 5. Finally, we conclude the paper with an outlook in Sec. 6.

## 2 Related Works

Previous works differ either in the domain and used categories for website classification or in the used machine-learning-based approaches. In the following, we provide a detailed overview of *website classification schemas* and *website classification methods*.

**Website Classification Schemas.** Lindemann and Littig [1] identified a limited set of website categories by analysing textual data present on websites. They derived the following categories for websites by applying a task-specific algorithm: *academic, blog, community, corporate, information, nonprofit, personal*, and *shop*. This classification schema partly overlaps with the classes introduced in this paper. In contrast, we propose readily available, general-purpose approaches for website classification.

Thapa et al. [2] introduced the four non-topical categories *public, private, non-profit and commercial franchise* in the food domain. Although the four classes are similar to our classification schema, we follow a cross-domain approach that is applicable to the entirety of the web.

Kanaris and Stamatatos [3] used seven categories for classifying websites: *blog*, *e-shop*, *FAQs*, *online newspaper*, *listings*, *personal home page*, and *search page*. However, these labels only describe some elements of a website and are not designated to indicate the commercial intent. For instance, blogs can be

[2] See `https://github.com/michaelfaerber/website-classification/`.

run in a *commercial* and *non-commercial context*. Furthermore, other important categories such as *corporate websites* are not included in this schema. The proposed categories might be sufficient for a benchmark data set, but cannot be used to categorise all websites on the web.

Meyer zu Eissen and Stein [4] used eight categories for website classification, such as *help*, *article*, *shop* and *non-private portrayal*. Note, that the categories are not driven by commercial intent. For instance, *non-private portrayal* contains websites of businesses and non-profit organisations.

**Website Classification Methods.** Bruni and Bianchi [5] applied machine-learning-based approaches to identify the commercial intent of websites. For each website, they aggregated multiple web pages into a single document for document classification and applied support-vector machines and random forests. Although similar to our approach, the scope is limited to a binary classifier for the e-commerce domain determining whether a website offers goods and services or not.

Studies using support-vector machines have been carried out by Sun et al. [6] in the academic domain and by Thapa et al. [2] in the food domain. The latter consider multi-label classification on a small balanced data set with about 100 websites. In contrast, we follow a single-label approach on a large data set and analyse the results of multiple machine-learning algorithms and imbalanced training data sets.

Sahid et al. [7] compared various algorithms for the task of website classification as well as different ways to weigh the given input texts. Specifically, they analysed the performance of Naive-Bayes, support-vector machines and multilayer perceptrons for classifying the industry of e-commerce websites.

AbdulHussien [8] studied the suitability of random forests for website classification of health websites and provided an outlook of the potential benefits of neural networks. Note, that we do not use stemming in data preparation due to potential information loss. Xhemali et al. [9] explore the benefits of neural networks for website classification of training course websites and compare the results with other machine learning algorithms, such as Naive-Bayes and decision trees.

## 3 Website Categories

In this paper, we propose the following four categories for website classification having a distinct level of commercial intent. We argue that this classification is sufficient to categorise the entirety of the web.

**Profit-oriented Company.** *(commercial intent: high)* A company or business is an economic, financial and legal entity acting according to economic principles. Their goal is to realise financial gain; as such, they are also referred to as for-profit organisations (FPO). Example websites are `fahrschuleanik.de` and `dietz-fruchtsaefte.de`.

**Public Institutions.** *(commercial intent: medium)* Public institutions are established on the basis of public law. Websites from public institutions include

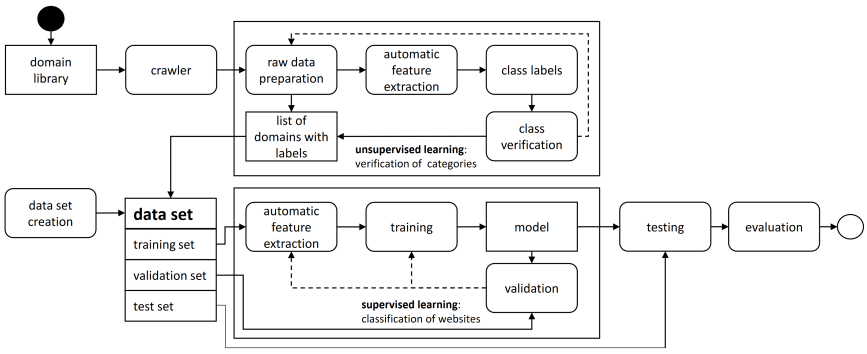


Fig. 1: Process of website classification

pages operated by federal and state governments as well as public institutions, municipalities, universities or state schools. Example websites are `kit.edu` and `stuttgart.de`.

**Non-profit Organisation.** *(commercial intent: low)* Following the notion of the *International Classification of Nonprofit Organisations* [10], a standard to classify non-profit organisations (NPO), an NPO fulfills the following criteria: (1) organised, (2) private, (3) self-governing, (4) non-profit-distributing, (5) voluntary. Example websites are `tc-mudau.de` and `adac.de`.

**Private Websites.** *(commercial intent: none)* A private website usually follows a private objective of an individual without commercial intent. Although the boundaries to other categories are sometimes ambiguous, we define a private website according to the following criteria: (1) No paid advertisement, such as Amazon affiliate links (2) No contact information or imprint, as this is required by law for German websites (3) The site is operated by an individual or a group of individuals. Example websites are `fester.de` and `edithundsven.de`.

## 4 Data Sets and Feature Extraction

In the following, we describe our data set, the required data preparation steps, and the feature extraction methods. Given an imbalanced distribution of classes, we consider three different training data sets and experiment with multiple feature extraction methods. An overview of the entire process, including training and testing, is provided in Fig. 1.

**Data Sources.** We start with a collection of websites, the *domain library*, consisting of two subsets: (1) The *directory-based subset* contains websites that are labelled automatically according to the type of directory and the information provided by the directory. As the directory listings might not match exactly and contain websites of multiple classes, the labels were reviewed manually to a large extent. The websites of all four categories are retrieved from relevant pages

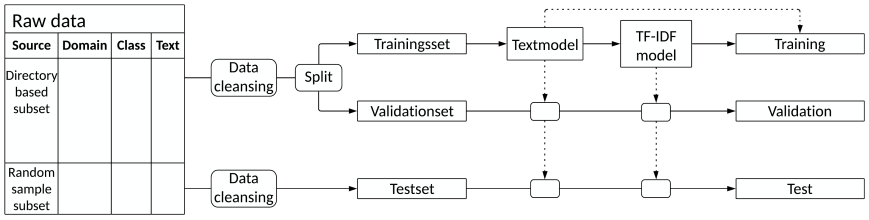


Fig. 2: Data preparation steps

dedicated to German websites such as *DMOZ*,[3] *project Curlie*,[4] *NPO Manager*,[5] and *Schulliste*.[6] (2) We use the *random sample subset* as a test data set that consists of a random sample of 1,500 domains with the German top-level domain *.de*, whereof only websites with useful content are considered.[7] All in all, we keep 1,109 websites and label them manually.

**Data Cleansing.** For each website, we crawl up to 30 pages and extract the textual information into a single document[8]. We remove non-visible textual information such as HTML markups, as well as special characters, non-German letters and numbers. Furthermore, we omit documents with less than 100 characters, as they are mostly error and domain parking pages.

**Class Distribution.** As the performances of some classification algorithms require knowledge of the a priori probabilities of the classes, we analyse the distribution of our four classes. Based on our *random sample subset*, we extrapolate the distribution of classes to be 73.2% commercial, 16% non-commercial, 9.1% private and 1.7% public institutions. Given the sample size and a total of approximately 16 million `.de` domains registered at DENIC, we derived a confidence level of 99% and a standard deviation of 4%.

**Training & Validation Data Sets.** Due to the imbalance in the class distribution, we experiment with three different data sets as depicted in Fig. 2. An overview of the subsets is given in table 1. Note, that each data set is split into training and validation set with a ratio of 3 : 1.

1. **Balanced Data Set.** Each class is weighted similarly.
2. **Distribution Data Set.** Each class is weighted according to the distribution of the *random sample subset*.
3. **Quality Data Set.** Similar to the distribution data set, but considering only documents whose class labels were reviewed manually.

[3] `https://dmoz-odp.org/World/Deutsch/`, accessed on 2019-10-24
[4] `https://curlie.org/de/Gesellschaft/Menschen/Pers%C3%B6nliche_Homepages`
[5] `http://www.npo-manager.de/vereine/`, accessed on 2019-10-24
[6] `http://www.schulliste.eu/`, accessed on 2019-10-24
[7] We remove unavailable domains or domain parking pages, i.e., websites with default content provided by the domain name registar.
[8] We consider only static visible textual information as input for classification, hence no HTML markups, meta tags or JavaScript.

Table 1: Absolute frequency of classes in the different data sets

| Data set | Split | Comp. | NPO | Priv. | Publ. | Total |
|---|---|---|---|---|---|---|
| Full DS | Total | 16,735 | 8,679 | 3,571 | 1,567 | 30,552 |
| Balanced DS | Total | 950 | 950 | 950 | 950 | 3,800 |
| | Training | 703 | 697 | 747 | 703 | 2,850 |
| | Validation | 247 | 253 | 203 | 247 | 950 |
| Distribution DS | Total | 10,450 | 1,306 | 2,283 | 239 | 14,278 |
| | Training | 7,827 | 966 | 1,740 | 175 | 10,708 |
| | Validation | 2,623 | 340 | 543 | 64 | 3,570 |
| Quality DS | Total | 2,100 | 1,500 | 1,500 | 930 | 6,030 |
| | Training | 1,600 | 1,000 | 1,000 | 600 | 4,200 |
| | Validation | 500 | 500 | 500 | 330 | 1,830 |
| Test DS | Total | 842 | 113 | 144 | 10 | 1,109 |

**Test Data Set.** In all cases, the *random sample subset* is used as the test data set to establish a consistent basis for comparison.

**Data Preparation Method.** We analyse multiple feature extraction methods w.r.t. their suitability for website classification. The basis for all features are n-grams extracted from documents. We consider only n-grams that occur at least 1% and no more than 50% of the documents.

We consider the following *feature extraction methods*:

1. **Full Vocabulary without Weights.** We consider all words but stop words. Our list of stop words is based on the R package *stopwords* [11] for the German language that we extend by common words occurring in error messages, such as HTTP status codes.
2. **Full Vocabulary with Weights.** We consider all words and use weights based on tf-idf. We do not remove stopwords.
3. **Reduced Vocabulary with Weights.** We consider only the words of the 5,000 most frequent features and use weights based on tf-idf.
4. **1- & 2-grams with Weights.** We use n-grams with size 1 and 2 as features and use weights based on tf-idf.

Note, that *convolutional neural networks* follow a different approach. Instead of a bag-of-words representation, they are based on word embeddings. For our experiments, we choose a sequence length of 2,000 words and consider only the (i) 25,000 and (ii) 50,000 most common word embeddings of each data set.

**Discussion.** We publish the implementation and data sets online for subsequent research[9]. As shown in table 1, the full data set contains 30,552 websites. The training and validation data sets are randomly chosen from the full data set and documents with less than 100 characters are omitted in the *data cleansing* step. The distribution of character counts is shown in Fig. 3.

[9] The data sets are freely available for research purposes at `https://github.com/michaelfaerber/website-classification/`.

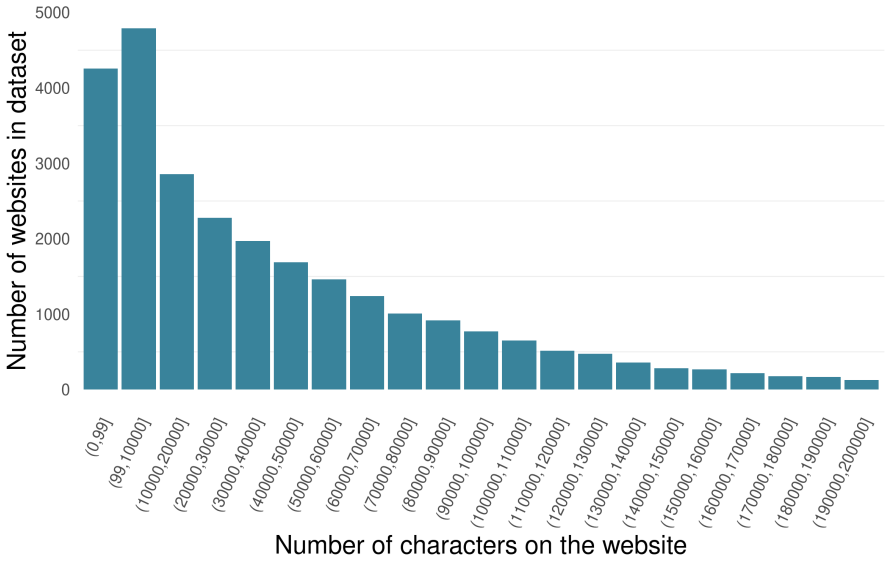


Fig. 3: Histogram of character count

## 5 Evaluation

### 5.1 Approach

As outlined in Table 2, 3, and 4, we use abbreviations to describe algorithms, data sets, and data preparation methods and introduce the following notation: $\text{model}^{\text{preparation method}}_{\text{training data set}}$. For instance, $\text{NB}^{T}_{B}$ describes a Naive-Bayes classifier trained on the balanced data set with td-idf as feature weights.

### 5.2 Website Classification using Unsupervised Algorithms

In Sec. 3, we argued that our four classes are sufficient to categorise the entirety of the web. Considering textual information, we show that unsupervised learning algorithms can distinguish these classes, too.

For a better visualisation, we analyse a subset of the *balanced data set* with *full vocabulary with weights* as the feature extraction method. For each class, we choose 300 documents and cluster them with the following methods:

- **k-means** is often used for partitioning data. We set the number of clusters manually to $k = 4$ and achieved an accuracy of 0.65 and an $F1$-score of 0.64.
- **DIANA** is a hierarchical, divisive clustering algorithm. It achieved the best results with six clusters, consisting of four large clusters that represent our four classes. When we disregard the two small clusters, we achieve an accuracy of 0.71 and an $F1$-score of 0.71.

Table 2: Abbreviations of algorithms

| Abbrev. | Model |
|---|---|
| NB | Naive-Bayes |
| RF | random forest |
| GB | gradient boosting |
| SVO | support-vector machine one-versus-one |
| SVR | support-vector machine one-versus-rest |
| MP[i] | multilayer perceptron nr. i |
| CN[i] | convolutional neural network nr. i |

Table 3: Abbreviations of data sets

| Abbrev. | Training data |
|---|---|
| B | Balanced DS |
| D | Distribution DS |
| Q | Quality DS |

Table 4: Abbreviations of preparation method

| Abbreviation | Variant |
|---|---|
| U | Full vocabulary without weights |
| T | Full vocabulary with tf-idf weights |
| R | Reduced vocabulary (5,000 most popular words) with tf-idf weights |
| 1G | Using 1-grams with tf-idf weights |
| 2G | Using 2-grams with tf-idf weights |
| 25k | Vocabulary with the 25,000 most popular word embeddings (CNN) |
| 50k | Vocabulary with the 50,000 most popular word embeddings (CNN) |

Both clustering methods confirmed that the introduced four classes can be found using unsupervised learning algorithms. We plot the data in Fig. 4 and conclude that, besides two negligible clusters (yellow and orange), the four classes are sufficient to classify the entirety of the web. Furthermore, we determine a strong overlap between company and private websites. The distinction between these classes turns out to be difficult using solely textual information. For instance, many of the red dots in the upper-right quadrant turn out to be private instead of company websites. As discussed in detail in Sec. 5.6 this is due to similar vocabulary in ambiguous cases, whereas a more distinctive vocabulary makes separation clearer for the other classes.

### 5.3 Evaluation Setup

In the following, we outline how we evaluated *seven machine learning methods* for website classification. We trained and evaluated all models using a server with 40 CPU cores, 565 GB RAM, Python 2.7 and R version 3.6. The training was conducted on a single GPU with 32 GB, model NVIDIA Tesla V100.

- **Guessing.** As a simple baseline, this method makes random guesses concerning the class assignment, using either the class distribution a priori or the most popular class as a fixed assignment.
- **Naive-Bayes.** We choose the Naive-Bayes classifier as one of our baselines.

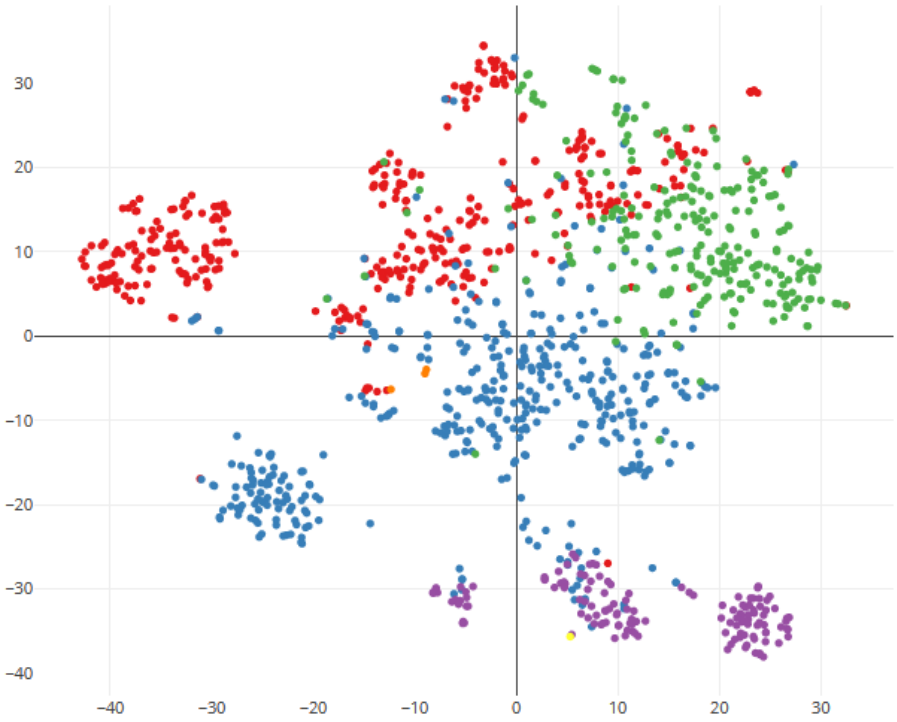


Fig. 4: Result of divisive clustering algorithm *DIANA*: company (red), NPO (blue), private websites (green), public institutions (purple)

- **Random Forest.** We use the R package *randomForest* with parameters $ntree = 500$ and $mtry = 150$, following the advice of Liaw & Wiener [12] for cases where only relevant features are to be found.
- **Gradient Boosting.** We use the R package *xgboost* with the booster *gblinear* and the parameters $n = 250$ and $k = 15$ for all models. All training was terminated before reaching *nrounds* rounds, when no improvements were observed. The standard value of $max.depth = 6$ was reduced in certain training variants and chosen between $[2\,;6]$. The results show that model performance is affected by the composition of training data.
- **Support-vector Machine.** We run and evaluated both the SVM one-versus-one as well as SVM one-versus-rest variant using the R package *e1071*. A linear kernel and $cost = 200$ were used for all models.
- **Multilayer Perceptron.** We evaluated the following four MLP architectures:
  1. Two hidden layers with 10 neurons each.
  2. Two hidden layers with 30 and 15 neurons.
  3. Two hidden layers with 100 neurons each.
  4. Three hidden layers with 50, 10, and 50 neurons, i.e. the second layer acts as an artificial bottleneck.

We chose *sigmoid function* for all hidden layers as it is suitable for text classification tasks and quick to calculate using backpropagation [13]. *Softmax* is used for all output layers. Due to the high dimensionality of our input, the highest amount of the neurons is located in the input layer. Therefore, most edge weights exist between input and the first hidden layer. During training, we used *Adam optimizer* [14] to achieve significantly faster run-times during training.

– **Convolutional Neural Network.** We evaluated the following two architectures, inspired by Chollet [15]: (i) Three convolutional layers with 128 filters each and kernel sizes of 3, 2 and 3. (ii) Two convolutional layers with a kernel size of 9. Both variants have an input layer of 2,000 neurons and a fully connected layer with 100 neurons feeding into an output layer with 4 neurons.

  Convolutional neural networks (CNN) do not use one-hot encoded inputs, but rather rely on vectorised contiguous text extracts of the same length (2,000 words in our case). The vectors are created using *fastText embeddings* [16]. A longer vector increases the number of trainable parameters drastically and may lead to *overfitting* and *longer training times*. Thus, we analyse whether a shallow CNN with a larger context window, i.e. kernel size, will lead to performance increases and reduced overfitting.

### 5.4 Evaluation Results

In the following, we present our evaluation results.[10]

**Guessing.** A simple classifier always guessing the most popular class achieves with 10.000 guesses an accuracy of 0.76 and a macro-F1 score of 0.215 on the test data set, representing the relative frequency of the company class. Another classifier that considers the class distributions for guessing achieves a lower accuracy of 0.60 and a macro-F1 score of 0.25. This shows, that the performance of guessing is highly dependent on the class distribution in the test data set.

**Naive-Bayes.** The results underline the dependence of performance on the training data. We achieve the best results using the balanced or (rather balanced) quality data sets with an accuracy of at least 0.72. The results using the (imbalanced) distribution data set were significantly lower.

The best performing model was $\mathrm{NB}_Q^R$, with the highest micro-F1 score of 0.78 and the highest macro-F1 score of 0.57 as well as the second-highest accuracy of 0.75. Predictions for the classes companies and NPO were notably accurate with a score of 0.94 and 0.7, respectively.

**Random Forest.** In contrast to Naive-Bayes, we achieved the lowest scores with the balanced data set, whereas the model $\mathrm{RF}_Q^R$ achieved the highest score. The classes company and NPO are labelled with an accuracy of 0.94 and 0.86, respectively. The overall accuracy of 0.84 outperforms the Naive-Bayes classifier.

[10] We published the confusion matrices for each model at `https://github.com/michaelfaerber/website-classification/`.

Table 5: Overview of the best models for each MLP architecture

| Model | Accuracy | Macro-F1 | Micro-F1 |
|---|---|---|---|
| $MP1_D^T$ | 0.866 | 0.689 | 0.870 |
| $MP2_D^T$ | 0.861 | 0.679 | 0.867 |
| $MP3_D^R$ | 0.849 | 0.676 | 0.855 |
| $MP4_D^T$ | 0.861 | 0.710 | 0.869 |

Note, that the $RF_D$ models could not label a single website of the class public institutions, possibly due to insufficient training data in the distributed data set.

A deeper look at the decision trees of each model shows that most private websites are classified following the exclusion principle, i.e. the trees split on words that are distinctive for a class. If none of the splits apply, the document is classified as a private website. This explains why even the best random forest models perform poorly classifying private websites.

**Gradient Boosting.** The best gradient boosting model ($GB_Q^R$) is trained using the quality set with the reduced weighted vocabulary (accuracy: 0.82, macro-F1: 0.66, micro-F1: 0.83). Similar to previous models, the distinction between private and company websites proves to be a difficult task. More than half of the websites classified as private are websites of companies or NPOs.

**Support-vector Machine.** The models $SVO_D^T$ and $SVR_D^T$ (accuracy: 0.86, macro-F1: 0.68, micro-F1: 0.86) achieve the best scores on the distribution data set and thus are chosen as best-performing variants. The difference between both models is marginal.

The output of an SVM using one-versus-rest can be interpreted as the confidence score of a class label. With this, we were able to analyse the effects of various thresholds for confidence values. Fig. 5 shows the relationship between a given threshold, accuracy, and percentage of classified websites. About half of the websites can be classified with a threshold of 0.94, increasing the accuracy to 0.97. The idea behind this analysis reflects real-world settings, where particular difficult websites might be labelled manually.

**Multilayer Perceptron.** A summary of the best performing models for all four architectures is presented in Table 5, with the best overall model being $MP1_D^T$, achieving the highest accuracy and micro-F1 score.

Models trained on the quality and distribution data set achieve similar results, though no variant performs best in all metrics. As the largest data set $D$ has slightly better results, we conclude that the size of the training data has a strong influence on the performance.

Similar to SVMs, we can interpret the output of each classification as a confidence score for classification and analyse the effect of a manual threshold (as depicted in Fig. 5) for model $MP1_D^T$. A threshold of 0.92 allows for 75% of websites to be classified, increasing the accuracy to 0.94. Note, that raising the threshold does not lead to all classes being omitted equally. For instance, classifications of the classes private, NPO and public institutions are discarded

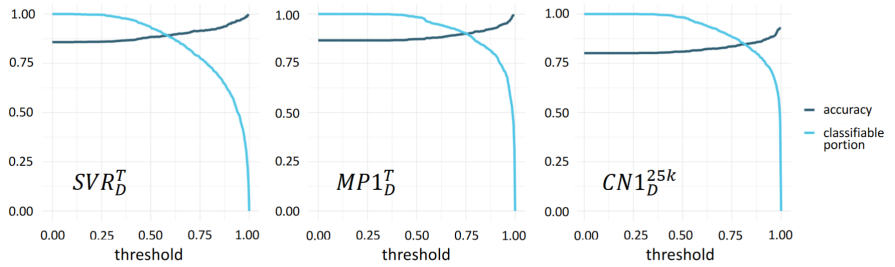


Fig. 5: Accuracy and ratio of classifiable websites depending on threshold value for $\text{SVR}_D^T$, $\text{MP1}_D^T$ and $\text{CN1}_D^{25k}$.

earlier than the class company due to their relative frequency. Therefore, when choosing the ideal threshold value, the distribution of classes must be considered.

**Convolutional Neural Network.** $\text{CN1}_D^{25k}$ is the best performing CNN variant and achieves an accuracy of 0.80 and a micro-F1 score of 0.80. It achieves a macro-F1 score of 0.55, the second-highest of all CNNs. The larger context window of the shallow CNN does not provide any improvement. This implies that classes are defined rather by individual words than longer coherent sentences.

The analysis of output thresholds for the best performing variant $\text{CN1}_D^{25k}$ is depicted in Fig. 5. Considering a threshold of 0.92 the model can classify 75% of all websites and achieves an accuracy of 0.87. With a threshold of 0.99, about half of all websites can be classified with accuracy increasing to 0.92.

### 5.5 Comparisons

In the following, the best performing models of each algorithm are compared according to accuracy, micro-F1 and macro-F1 scores as well as the run-times of training and classification. Afterwards, we will discuss the shortcomings and difficulties faced.

**Effectiveness.** A summary of the results can be found in table 6.

We achieve the best evaluation results for classification using a multilayer perceptron with a simple architecture. Experiments with dropout layers did not improve the performance of our models. The model $\text{MP1}_D^T$ achieves the highest scores in all three metrics. Similar performances are achieved by the SVM with one-versus-rest implementation, achieving only insignificantly lower scores compared to the MLP.

The Naive-Bayes classifier achieves the worst performance in comparison, though no parameters need to be optimised. Furthermore, it only requires very few training data and features. On top of that, the algorithm works well with balanced data, meaning no previous knowledge of class distribution is necessary. The classifier is therefore useful for a first analysis to determine the suitability of machine-based classification for a specific domain.

Despite successes in the latest researches on text classification, our results with CNN and pre-trained word embeddings did not yield good results. Other

Table 6: Overview of the best performing models for each algorithm

| Model | Accuracy | Macro-F1 | Micro-F1 |
|---|---|---|---|
| **$MP1^{T}_{D}$** | **0.866** | **0.689** | **0.870** |
| **$SVR^{T}_{D}$** | **0.857** | **0.678** | **0.861** |
| $SVO^{T}_{D}$ | 0.854 | 0.676 | 0.858 |
| $RF^{R}_{Q}$ | 0.844 | 0.552 | 0.840 |
| $GB^{R}_{Q}$ | 0.821 | 0.664 | 0.834 |
| $CN1^{25k}_{D}$ | 0.796 | 0.550 | 0.797 |
| $NB^{R}_{B}$ | 0.736 | 0.571 | 0.762 |
| Guessing | 0.759 | 0.216 | 0.655 |

Table 7: Overview of training and classification run-time

| Model | Training (sec.) | Testing (sec.) |
|---|---|---|
| **$MP1^{T}_{D}$** | **46.0** | **0.5** |
| $SVR^{T}_{D}$ | 6,223.0 | 36.7 |
| $SVO^{T}_{D}$ | 917.0 | 20.1 |
| $RF^{R}_{Q}$ | 815.1 | 0.3 |
| **$GB^{R}_{Q}$** | **2.0** | **0.1** |
| $CN1^{25k}_{D}$ | 94.2 | 0.2 |
| $NB^{R}_{B}$ | 2.0 | 65.4 |

algorithms consistently achieve higher accuracy and F1-scores under similar training conditions. We conclude that for our use case models benefit rather from finding meaningful keywords within the text than interpreting coherent sentences.

We achieved similar results to the work done by Lindemann and Littig [1]. They also had difficulties to distinguish private websites from the categories "blog"[11] and "corporate."[12] They achieved an accuracy of 0.84 and a micro-F1 score of 0.84, which we surpassed with our MLP as well as SVM approaches.

Thapa et al. [2] achieve the best results (macro-F1: 0.74, micro-F1: 0.73) with an SVM classifier and multi-label approach on a balanced data set with about 100 websites. Although they consider additional features such as structural information and URIs, our model $MP1^{T}_{D}$ using a simple multilayer perceptron architecture (macro-F1: 0.69, micro-F1: 0.87) shows that basic textual information as a feature is sufficient for comparable performance on a large, imbalanced data set.

As depicted in Fig. 5, the outputs of the models $MP1^{T}_{D}$, $SVR^{T}_{D}$ and $CN1^{25k}_{D}$ can be interpreted as confidence scores and thus allow experimentation with threshold values for classification. The performance of our MLP can be improved to 0.94, whilst still able to classify 75% of websites.

**Efficiency.** If a model is implemented in a real-world setting and productive system, regular retraining on large data sets is required. Therefore, training time is an important metric. Considering the hardware configuration described Sec. 5.3, an overview of the run-times of our implementations with average training and testing time of the best models is given in table 7.

The training times of the SVMs and the random forests are noticeably high. The longer training time for $SVR^{T}_{D}$ over $SVO^{T}_{D}$ was unexpected because fewer

[11] "Blogs" fall under the categories of *private* or *company* according to our defined classes from Sec. 3.

[12] This is a subset of our *company* class.

SVMs need to be trained [17], though they were implemented differently ($SVR_D^T$ as a wrapper and $SVO_D^T$ using the R package *e1071*).

The gradient boosting models exhibit the fastest training and testing, though many more pairs of hyperparameters need to be evaluated beforehand to determine the optimal setup, which is not accounted for in pure run-time analysis.

The Naive-Bayes classifier is the only algorithm with a higher run-time during testing compared to training. Because of its slow classification, it is better suited for cases where only a few classifications need to be made like local spam filters that must be retrained every time a new pattern emerges.

We conclude that a multilayer perceptron with a bag of words approach is the most promising solution to the task of website classification. Besides the best results, MLPs have a short classification run-time which can be easily improved through parallel processing with multiple GPUs.

**Feature Extraction Method.** A comparison between the four proposed training sets shows that a prior weighting of features through tf-idf is the most reasonable approach. No model achieved the best performance using non-weighted features. The average accuracy of all models using non-weighted full vocabulary reached 0.777, whereas the average accuracy of all models using weighted full vocabulary reached 0.798. This confirms results achieved by Sahid et al., in which weighting through tf-idf proved to be superior to non-weighted input [7].

Furthermore, a smaller vocabulary does not seem to necessarily lower performance scores. This effect is especially prominent for Naive-Bayes, random forest, and gradient boosting, where reduced vocabularies lead to the best results. A size of 5,000 words proves to be sufficient for our task at hand. All models trained using reduced vocabularies reached an accuracy of 0.794 on average. Finally, using 2-grams instead of 1-grams did not increase performance in our case. All models using 2-grams averaged an accuracy score of 0.789.

### 5.6 Classification of Private Websites

Our evaluation shows that both supervised and unsupervised algorithms cannot distinguish easily between private websites and company websites because private websites sometimes use commercial vocabulary in a non-commercial context. For instance, websites of musicians might be labelled as a private website in case of a school band whereas the portrayal of a singer might have a clear commercial intent. In some cases, this might be a challenge even during manual labelling. We conclude that the diversity of private websites creates a large feature space, leading to many cases where private websites are not classified correctly. A solution for this might be a multi-label classification approach as described by Thapa et al. [2]. However, in our case, a single-label approach was chosen to clearly define a distinct business strategy for the web hosting company.

### 5.7 Main Findings

1. We showed that there are many websites containing only few words and that distinguishing between private and company classes is a non-trivial task. Therefore, robust methods are required for website classification.
2. Our four proposed categories proved to be sufficient to cover the entirety of the web. As each class can be mapped to a target audience, we provide a real-world application for web hosting companies for determining their relationship and communication strategy with their customers.
3. Our work with unsupervised learning algorithms confirms the existence of our four proposed clusters. As for supervised learning, an MLP with a simple two hidden layer architecture proved to be the most suitable model for the task. Although SVMs achieved similar results, MLPs have a short classification run-time and, in general, run-times can be improved easily by parallel processing with multiple GPUs.
4. CNNs did not deliver superior results as performance is influenced rather by individual words than by longer coherent sentences.

## 6 Conclusion and Outlook

In this paper, we proposed four categories that can be used for website classification of the entirety of the web. We implemented various unsupervised as well as supervised machine learning algorithms for the purpose of automatic website classification. Furthermore, we discussed the efficiency and effectiveness of each method in a real-world setting. All in all, we achieved the best performance (accuracy: 0.866, macro-F1: 0.689, micro-F1: 0.870) using a multilayer perceptron that was trained on a data set with real-world distribution of classes using tf-idf as feature weights.

Experiences and insights gathered from this work could be applied to classifying other document types, categorization schemas, and languages. However, language-specific features might influence results, such as the required declaration of legal forms in Germany. Subsequent research can use our published implementations and data sets and, besides textual content, might consider additional features to improve our results, such as URIs and images.

## References


1. Lindemann, C., Littig, L.: Classification of Web Sites at Super-Genre Level. In Mehler, A., Sharoff, S., Santini, M., eds.: Genres on the Web. Springer (2011) 211–236
2. Thapa, C., Zaïane, O.R., Rafiei, D., Sharma, A.M.: Classifying Websites into Non-topical Categories. In: Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery. DaWaK'12 (2012) 364–377
3. Kanaris, I., Stamatatos, E.: Learning to recognize webpage genres. Information Processing & Management **45**(5) (2009) 499–512

4. zu Eissen, S.M., Stein, B.: Genre Classification of Web Pages. In: Proceedings of the 27th Annual German Conference on AI. KI'04 (2004) 256–269
5. Bruni, R., Bianchi, G., et al.: Robustness analysis of a website categorization procedure based on machine learning. Technical report n. 04-2018 DIAG (2018)
6. Sun, A., Lim, E., Ng, W.K.: Web Classification Using Support Vector Machine. In: Proceedings of the Fourth ACM CIKM International Workshop on Web Information and Data Management. WIDM'02 (2002) 96–99
7. Sahid, G.T., Mahendra, R., Budi, I.: E-Commerce Merchant Classification using Website Information. In: Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics. WIMS'19 (2019) 5:1–5:10
8. AbdulHussien, A.A.: Comparison of Machine Learning Algorithms to Classify Web Pages. International Journal of Advanced Computer Science and Applications (IJACSA) **8**(11) (2017)
9. Xhemali, D., Hinde, C.J., Stone, R.G.: Naïve bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages. International Journal of Computer Science Issues **4**(1) (2009)
10. Salamon, L.M., Anheier, H.K.: The International Classification of Nonprofit Organizations. Jossey Bass Publishers (1996)
11. Benoit, K., Muhr, D., Watanabe, K.: stopwords: Multilingual Stopword Lists. (2019) R package version 1.0.
12. Liaw, A., Wiener, M., et al.: Classification and Regression by RandomForest. R news **2**(3) (2002) 18–22
13. Amajd, M., Kaimuldenov, Z., Voronkov, I.: Text Classification with Deep Neural Networks. In: International Conference on Actual Problems of System and Software Engineering (APSSE). (2017) 364–370
14. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference on Learning Representations. ICLR'15 (2015)
15. Chollet, F.: Deep Learning with Python. 1st edn. Manning Publications Co., Greenwich, CT, USA (2017)
16. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning Word Vectors for 157 Languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018). (2018)
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Information science and statistics. Springer (2007)