Learning from data: Assignment1

Emilio Oldenziel
s2509679
e.oldenziel.1@student.rug.nl

1 Introduction

A significant amount of product purchases are nowadays performed online. In most cases, the customer can leave a review of the bought product to let other customers know what their opinion about the product is. These reviews frequently not only contain a numerical rating but also a textual opinion. When extracted and labelled, these textual opinions can be used to train a classifier to predict its sentiment or to classify its subject. In this report, the experiments from the first lab assignment are discussed. In the experiments a model is created to classify product reviews.

2 Data

In the experiments a dataset is used, this dataset contains 6000 lines (instances) of text. The text is a review of a product. Each instance has 2 corresponding labels. The first label is the product's category, the classes are; music, dvd, software, books, camera or health. The second label represents the review's sentiment, this is a binary label which can be positive (pos) or negative (neg). A slice of the dataset can be observed in table 1

As can be observed in Table 2 and Table 3 the distribution of instances per class is not equal in any of the classes. In the sentiment the imbalance of the dataset is not very significant with a difference of 64 over 6000/2 but is not perfectly balanced. The dataset's balance with the category labels has a maximum difference of 41 over 6000/6 instances per class.

Class	Count
pos	2968
neg	3032

Table 2: dataset distribution for sentiment

Category	Senti ment	text
health	pos	895.txt the product was a ... science of superheroes , ” - reviewer
music	pos	791.txt this is a wondrful cd ... with your current work .
camera	pos	3.txt this was the best ... without it getting scratched
music	neg	220.txt i liked the santana - ... i will stay away .

Table 1: A slice from the dataset

Class	Count
music	1027
dvd	1012
software	994
books	993
camera	988
health	986

Table 3: dataset distribution for categories

3 Methods

To create a model that can classify reviews we first have to assemble a few steps.

3.1 Pre-proccessing

Before the training process can be performed, the data has to be pre-processed to a representation that is suitable for learning. Raw text data is not suitable to input into the used classifier. Therefore, we have have to transform the raw text to a count vector representation. This transform is performed using a Count Vectorizer which transforms the texts to a sparse matrix. In this matrix the features (columns) are all words that occur in the dataset (Answer to 1.5e). The rows (instances) are a vector representation instance's word count. This representation can optionally be extended to a TF-IDF representation where the instances are a representation of the importance of each word in the text.

3.2 Classifier

The experiments were performed using a Multinominal Naive Bayes classifier. This classifier is very suitable to be used for document classification using discrete features (Schütze et al., 2008). The classifier uses posterior probabilities calculated by percentage the feature i occurs for a class $p(i|c_j)$ and the percentage of instances having a class $p(c_j)$ which is called prior. The posterior probability for instance i being of class c_j is calculated by:

$$p(c_j|i) = \frac{p(i|c_j)p(c_j)}{p(c_j)p(i|c_j) + p(\neg c_j)p(i|\neg c_j)} \quad (1)$$

The class c_j that returns the highest probability is assigned to the feature.

3.3 Validation

A K-fold cross validation was used to validate the model. To score the predictions made by the model, the class-wise and average; recall, precision and f-1 scores were calculated after the training process in each fold.

3.4 Parameters

To be able to obtain the best model the (hyper) parameters of the pre-proccessing, classifier and validation have to be tuned.

- sentiment: If set to True, the sentiment labels will be used as labels otherwise (False) the categories are used as labels. Results of both labels will be discussed.

- tfidf: We can choose if we want a count vector representation (False) as pre-processing the data or use a TF-IDF representation (True). In our experiment we will use both options to compare.

- fit_prior: This parameter is by default set to True by Sklearn. We want to learn the priors to be stored in the model.

- K: The amount of folds in the cross validation process. We use 5 and 10 as values in the experiment.

4 Results

Averaging method	TF-IDF	Average f1
Micro	False	0.89
Macro	False	0.89
Micro	True	0.89
Macro	True	0.89

Table 4: Average f1-score average using the Category labels over the folds with $K = 5$.

Averaging method	TF-IDF	Average f1
Micro	False	0.79
Macro	False	0.79
Micro	True	0.79
Macro	True	0.79

Table 5: Average f1-score average using the Sentiment labels over the folds with $K = 5$.

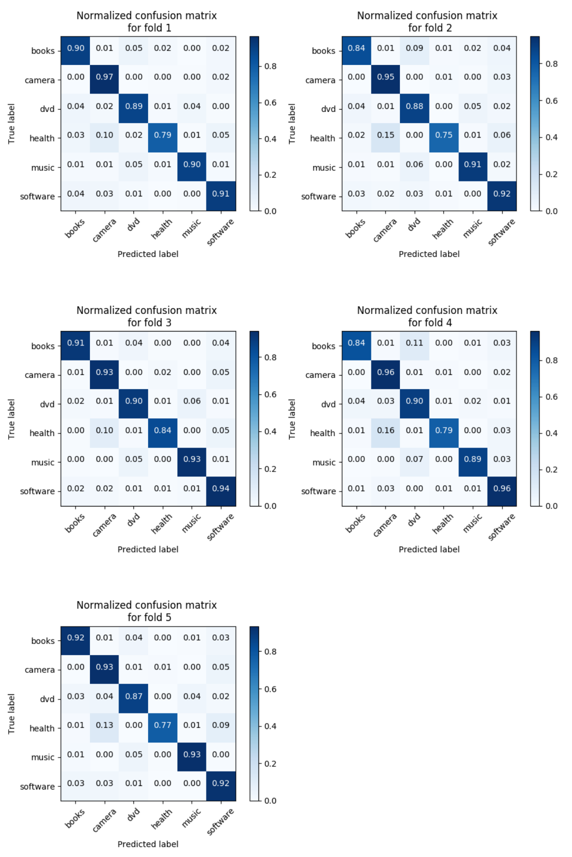

Figure 1: Confusion matrix for each fold, $K = 5$, $sentiment = False$, TF-IDF=$False$.

5 Discussion

From the results of the experiments we can observe in Table 4 and Table 5 that the results of each of the labelsets are uniform. The use of the TF-IDF vector and the count vector yield the same average- micro and macro f1-scores. The micro average for the category labels show us that the overall classification is quite decent. The macro average shows that the score which is obtained per class is in balance. (Oldenziel and Melchert, 2016) (Marina Sokolova, 2009). The scores from the sentiment labels are lower while there are fewer classes. This does suggest that the classification of sentiment is significantly harder than classification of the categories. Furthermore, in Figure 1 we can see the confusion matrices for each of the folds. We can observe that all classes are classified with around 90% and higher accuracy. The only outlier is the class Health, which is mistaken for Camera in 10% to 16% of the cases.

6 Conclusion

In these experiments we build two classifiers to classify product reviews. The classification of sentiment turns out to be a hard problem compared to determining the category. Overall, a Naive Bayes classifier is successful in building models for classification on this dataset.

7 Questions

- (a) One should not look at the test set, because the validation should be independent of what the content of the test set is. By doing so, one can not be biased what the model should look like to better fit the test set. This assures an objective validation of the model.
- (b) The dataset is split in K equal slices where K rounds of training and validation. In each round $K - 1$ slices are used for training and 1 as dev/test set. This assures the validation is performed on all projections of the dataset.
- (c) A simpler classifier could be used as baseline to see what the classification gain in for the more complex model. For the binary class problem, assigning the majority class label to all instances can be used as baseline. The majority class consists of 3032 out the 6000 instances which will correspond to 51% accuracy. For the six-class problem this corresponds to 17% accuracy which is a very low number and quite useless. Therefore, 50% and higher is a number where the model become useful or a simpler classifier can be used to create a baseline.
- (d) As mentioned in the discussion. The confusion matrix is effective for observing where misclassifications/errors occur and what classes are assigned to the errors.
- (e) See section 3.1

References

Guy Lapalme Marina Sokolova. 2009. A systematic analysis of performance measures for classification tasks. *Neural Computation 21, 35323561.*

Biehl Oldenziel and Melchert. 2016. Multi-class classification of functional data.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press.