SAKI SS19 Homework 1

Author: Jannis Wolf

Program code: https://github.com/JannisWolf/transaction_classifier

Summary

Through the more and more overarching digitalization, the demand for data scientists in industry has been steadily rising to take advantage of the flood of data that is recorded every day. For example, financial transactions involving physical money have largely been replaced by digital transactions between finance institutes. This opens up the opportunity for data mining and machine learning algorithms to be integrated into industry software applications.

In the following, it is evaluated how a multinomial naive bayes classifier can be trained to classify a small data set of real transactions (sample size = 209) into six categories (finance, income, leisure, living, private, standardOfLiving). Each sample of the set has ten properties and a label. General characteristics like job account number, dates of payment, and even the amount did not qualify as relevant features as they hold very little information to describe transactions. The features contained in posting text, reason for payment, and payer / recipient were extracted using a common bag of words approach to represent the text data as numerical vectors. Automatic filter-based feature selection like chi square test or TF.IDF were discarded because of the small vocabulary retrieved from the the data set. Furthermore, the performance of the naive bayes classifier is fast enough and did not benefit from a dimension reduction of the feature set. But a manual wrapper method like feature selection tested combinations of posting text, reason of payment, and payer / recipient to optimize the prediction capabilities. The best setting was found by concatenating all three properties and use it as one document in the count vectorizing process.

Eventually, a prediction score of 90% in the F1 measure could be achieved. A detailed evaluation and discussion of the results is presented below.

All results can be exactly reconstructed through the provided Jupyter Notebook, as the seed for every random numbers is set at the beginning of all calculations.

Evaluation

The evaluation is structured in three parts. A description of the used metric, a presentation of the results and a dedicated discussion eventually followed by a conclusion.

Metric: A crucial part in evaluating the performance of machine learning models is choosing an appropriate metric. To deal with a multiclass classification problem which has an uneven class distribution, more complex measures than accuracy have to be utilized. A very common one is the F1 score that relies on precision and recall. An intuitive explanation is possible through the visualisation of a multiclass confusion matrix M. With N classes, it is a N×N matrix, with the vertical axis showing the true class and the horizontal axis showing the class predicted by the classifier. Each element i,j of the matrix would be the number of items with true class i that were classified as being in class j.

Recall calculates how many of the actual positives the model capture through labeling it as positive (True Positive). It is widely used when there is a high cost associated with false negative, like in fraud detection.

$$Recall_i = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{M_{ii}}{\sum_j M_{ij}}$$

Precision gives a measure how many of the predicted Positives are actual positive. It is a good measure to determine, when the costs of False Positive is high, like in email spam detection.

$$Precision_i = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{M_{ii}}{\sum_j M_{ji}}$$

F1-Score combines these to measurement if a balance between precision and recall is seeked..

$$F1\,Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Results and discussion: After training the multinomial naive bayes classifier with 75% of the data set, a weighted average f1-score of 0.9 is achieved predicting the unseen test set (Table 1 on the following page). Three classes show perfect precision, while three classes show perfect recall, respectively. The category 'income' even obtained a f1-score of 1. By looking at the raw data, it can be clearly seen why 'income' has a high recall and precision considering that the posting text and reason of payment are very well distinguishable of other classes. A chi square statistic between features and target variables was conducted. The ten highest ranked features are depicted in Figure 2. The top 3 shows features exclusively belonging to the class 'income' ('kg','gehalt','adorsys'). The high recall of the class 'leisure' could derive from the fact that it has the largest prior probability. Ideally, a naive bayes classifier is trained on the same amount of samples per class.

On the other end, 'private' is relatively easy mispredicted (recall = 0.67), for example as finance. A closer look into the data set reveals, that the recipient of 'finance' transactions ('Isabel Anna') has the same surname as the recipient of some 'private' transactions ('Anna Fein'). This uncovers the overall problem this classifier suffers from. It is heavily trained on certain keywords that are derived from very person dependent data. If this is put into perspective the very high overall f1-score has to be interpreted with caution, as the model will very likely generalize poorly on transaction records of different people.

Conclusion: In summary, this task showed how powerful a simple naive bayes classifier in combination with some text mining strategies like the bag of word method can perform. Speaking in terms of machine learning, where huge amounts of data is needed, the given data set was extremely small. Regardless, a high overall f1-score was achieved, while the calculations executed incredibly fast.

However, there is a drawback which need to be mentioned. The model fits the training data set very accurately, but will fail for data from different person as the trained sets are very person dependent. Coming back to the topic of industry applications, the above described approach lacks of more diverse labeled data for a reasonable implementation, which works on more than one user.

Screenshot

Figure 1:

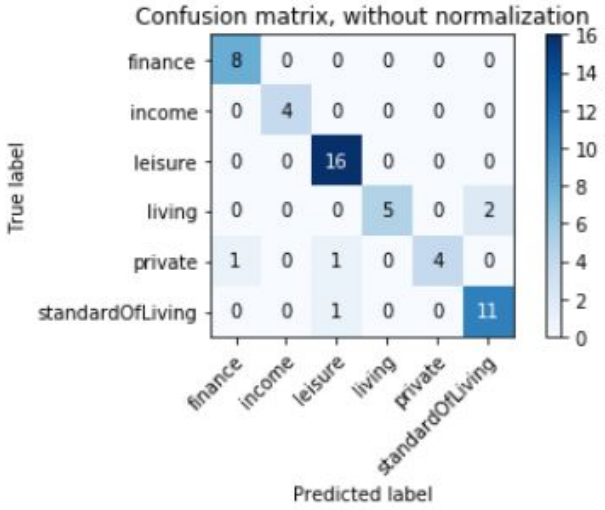

Table 1:

	precision	recall	f1-score	support
finance	0.89	1.00	0.94	8
income	1.00	1.00	1.00	4
leisure	0.89	1.00	0.94	16
living	1.00	0.71	0.83	7
private	1.00	0.67	0.80	6
standardOfLiving	0.85	0.92	0.88	12
accuracy			0.91	53
macro avg	0.94	0.88	0.90	53
weighted avg	0.91	0.91	0.90	53

Figure 2: