SAKI SS19 Homework 1

Author: Daniel Seitz

Program code: https://github.com/Anto4ka/transaction-classifier

Summary

For a given set of financial transactions, the challenge is to develop a classifier, that categorizes every entry into one of the following categories:

- Income
- Private (cash, deposit, donation, presents)
- Living (rent, additional flat expenses, ...)
- Standard of living (food, health, children, ...)
- Finance (credit, bank costs, insurances, savings)
- Leisure (hobby, sport, vacation, shopping, ...)

The chosen approach is a Multinomial Naive Bayes classifier, which was trained and validated on a dataset containing 209 entries. Training data and Validation data were split 75% to 25%.

Every raw data entry contains the nine attributes “Auftragskonto”, “Buchungstag”, “Valutadatum”, “Buchungstext”, “Verwendungszweck”, “Beguenstigter/Zahlungspflichtiger”, “Kontonummer”, “BLZ”, “Betrag” and “Waehrung”. For lack of relevant information every attribute was dropped, besides “Buchungstext”, “Verwendungszweck” and “Betrag”.

The “Buchungstext” attribute is being transformed by a pipeline, consisting of an Imputation transformer and an Encoder. The former fills in missing values, while the latter one maps the different values to numerical categories.

The “Verwendungszweck” attribute is being transformed by a pipeline, consisting of a “Bag-of-words”-transformation, which converts text into numerical features, followed by a term-frequency normalization.

Lastly, the “Betrag” attribute is being cleaned up and normalized, to be put into a transformation pipeline. It consists of an Imputation transformer and a “3-Bins-Discretizer”, which maps the value to one of three uniformely sized value-ranges.

The software’s architecture uses multiple cascading Pipelines, granting the opportunity of automated model selection. This was used in the development process, to empirically find the best paramters. An example would be, that it was found, that 3 bins are optimal for the discretization of the numerical “Betrag”-attribute.

It also simplifies the feature selection process, by offering an easy interface to implement new features. This was used in the analysis of the “Buchungstag”-attribute. After mapping the date to a weekday, the resulting attribute could easily be added to the pipeline handling the “Buchungstext”-attribute. Even though this indeed increased the performance score, the improvement was too insignificant, to be able to justify adding an other dimension to the feature vector.

The decision to choose a Multinomial Naive Bayes classifier is a result of the idea to vectorize the “Verwendungszweck” with the “Bag-of-Words” approach. This kind of classifier is based on a multinomial event model and works best with feature vectors that can be represented as histograms, counting the number of times a event occurred. Following this decision, the other two attributes “Betrag” and “Buchungstext” were converted to similar, discrete feature, to fit the classifier model.

Evaluation

For the evaluation of the results, a weighted F1 score was chosen. Generally speaking, the traditional F1 score is the harmonic mean of the Precision and the Recall.

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \qquad Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)}$$

Which results in the following formula:

$$F1 = \left(\frac{Precision^{-1} + Recall^{-1}}{2}\right)^{-1}$$

Even though this metric is not as easy to understand as intuitively as the raw ration between correct and incorrect classifications (Accuracy), it offers great benefits by taking both false positives and false negatives into account.

Now this definition is only applicable to binary classification problems. To scale this metric to a multiclass classification problem, we need to chose a strategy to calculate a scalar value out of the F1 scores of each class. An example would be to calculate the metric for each label, and to find their unweighted mean. The problem with this approach is, that it does not take label imbalance into account. Since our training data is highly imbalanced, this would be less than optimal.

A more appropriate approach is to again calculate the F1 score for every label, and to weigh the scores with their corresponding label-frequency.

```
Accuracy score:  0.9056603773584906
Precision score(weighted):  0.9318658280922432
Recall score(weighted):  0.9056603773584906
F1 score(weighted):  0.9041520910054082
```

The decision to use a Multinomial Naive Bayes classifier panned out great, regarding multiple reasons. The classifier converges quicker than other models (e.g logistic regression), which reduces the quantity of training data that is needed. Perfect for out small data set.
It also required a negligible amount of time to train the model (less than 5 seconds).

Possible Improvements

The biggest limiting factor in the development of the transaction classifier was the small data set. Besides the general saying “More data is better data”, this lack of information directly affected the two attributes “Verwendungszweck” and “Buchungstag”.

With a data set of this size it is rather hard to develop a significant “Bag-of-Words”-Vocabulary, to cover all the various reasons for a fincanial transaction. Since this is a matter of textual data, there is also no way to extrapolate the classifier to never seen words of unknown domains.

But while the “Verwendungszweck” attribute still succeeded in offering plausible estimations, the “Buchungstag”-attribute really suffered from the narrow training data. Intuitively information like the weekday, beginning/end of month or even time of year could offer great insights into the financial behaviour of an actor. A simple example could be, that most people tend to spend money on leisure activitys right after their paycheck arrived (sudden spike in money) or at the end of the month (“There is money left”). Sadly this data set does not offer enough transactions to find such patterns.

Another opportunity to improve the performance of this classifier, would be to model the K-Bins-Vectorization of the “Betrag” attribute in greater detail. For example, an ascending size of bin-ranges could help to better categorize payments in small, medium and big transactions.