SAKI SS 2021 Homework 1

Author: Jan Degen

Program code: <https://github.com/jandegen/saki-exercise-1/tree/master>

# Summary

The goal of this task is to implement a Naive Bayes classifier for bank transactions. Using a given data set, the transactions are to be classified according to the six classes "finance", "income", "leisure", "living", "private" and "standardOfLiving".

The available data contains eleven features and is also already labeled for training.

|  |  |
| --- | --- |
| Column | Description |
| Id | Continuous unique Id of the transaction |
| Auftragskonto | Numerical account number of the client |
| Buchungstag | Date of booking |
| Valutadatum | Valuta date |
| Buchungstext | Booking text |
| Verwendungszweck | Purpose of the booking as text |
| Beguenstigter/Zahlungspflichtiger | String or numeric value of the beneficiary or the payer. |
| Kontonummer | Numeric account number or IBAN of the recipient/sender |
| BLZ | Bank sort code |
| Betrag | Negative/positive amount to two decimal places |
| Waehrung | Currency oft he amount |
| label | Predefined label of the transaction |

Using a Naive Gaussian classifier, which is based on Naive Bayes networks, the probabilities of the class memberships of a transaction are calculated. In this variant, it is assumed that the individual features have no dependencies on each other.

Since a Gaussian classifier can only work with numerical values, the individual features must be prepared. For this the features order account, bank code, posting text, account number and currencies are converted into numeric categories. Since the purpose and the beneficiary/payee contain alphanumeric characters and have strong differences, occurrences of the words are used here. This is done using a CountVectorizer, which computes a given text into a matrix of word frequencies. Before this processing, however, the text is optimized by converting the entire text to lowercase and removing non-relevant characters (dots and special characters) as well as **g**erman stopwords (und, oder, etc.) based on the Natural Language Processing Kit. Furthermore, strings without content like "End-To-End" or "Notprovided" are removed. With the resulting dataset, a GaussianNB is trained and can thus be used for classification.

The provided data set included 209 transactions, which were used with an 80/20 ratio for training and testingsets. The achieved precision of the classifier is 92.85%.

# Evaluation

For the evaluation of the classifier and its predictions, the metrics accuracy, null accuracy as well as a confusion matrix of the classes and the distribution of the predicted classes by the model are considered.

**Confusion Matrix**

Figure 1 shows the confusion matrix and the distribution of the predicted classes given over the actual classes. It can be seen that out of the 42 elements in the test set, 39 were correctly identified and three of the elements were assigned to incorrect classes. It can also be seen that the classes "leisure"(11) as well as "standardOfLiving"(10) occur most frequently in the test set.

The confusion matrix can also be used to determine the classification error, which indicates the probability of the model making an incorrect prediction. This is defined by:

**Accuracy & Null Accuracy**

The classification report of Figure 3 shows the precision, recall, F1 scores and support of the model for each class. The precision indicates the ratio of correctly predicted elements in a class, while recall or sensitivity indicates the rate of predicted true positives out of all predicted true positives. The F1 score is the harmonic mean between precision and recall and indicates the effectiveness of the classification. Support indicates the number of elements that were assigned to this class by the model, regardless of whether this was correct or incorrect. Finally, the accuracy of a model results from the correctly classified elements over all elements of the set.

The accuracy describes the probability of a correct prediction of the class of a transaction. The accuracy of 92.85% is already very high, but the quality of the predictions can only be determined in comparison with the null accuracy which describes the probability of a correct prediction, if one always predicts the most frequent class. This is calculated by dividing the absolute frequency of the most frequent class by the number of all elements in the set.

The test set contains 42 elements of which 11 transactions belong to the class "leisure". This results in a null accuracy of 26,19%.

This shows that the model is able to predict much better than if it would always predict the most frequent class.

**Histogram of predicted probabilities**

The histogram shows the distribution of the probabilities per class of a prediction by the model. It can be seen immediately that the probabilities of the classes differ greatly. The probabilities are either close to 1 or 0 and there are no intermediate values. This suggests features that have a very strong and unique effect on the prediction.

# Screenshot

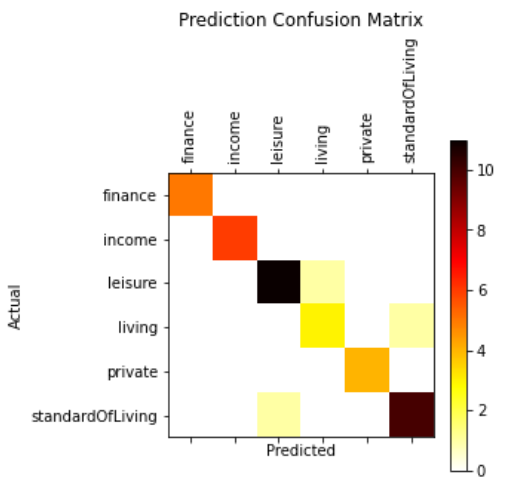


Figure : Prediction Confusion Matrix

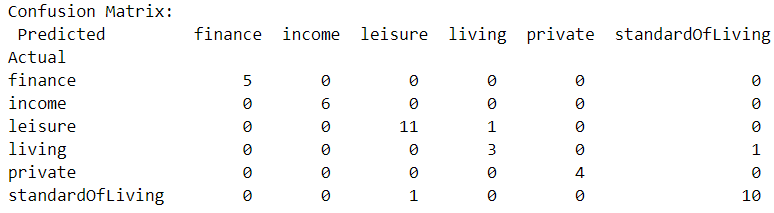


Figure : Confusion Matrix with values

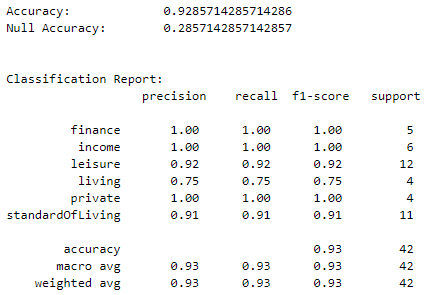
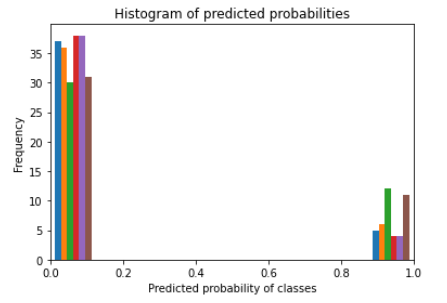


Figure : Classification Report

Figure 4: Histogram of predicted probabilties