184.702 Machine Learning 2019SS

Assignment 1: Classification Experiments

Aleksandra Radecka (e11847505)

Niklas Speckle (e1515258)

Nazika Moeininia (e1535524)

The following report summarizes the results of machine learning experiments on classification. Three different types of algorithms were tested on four different datasets (4x3) to get to know the nature of the classifiers. The aim was to analyse algorithms with different underlying mathematical fundamentals. Therefore, Naive Byes, K-Nearest-Neighbours and Random Forest was chosen. The experiments delivered insights on how the algorithms work and perform on datasets with different characteristics. Also, the impact of different parameter settings and pre-processing steps was investigated.

Data Description

1. Drugs Dataset

This dataset contains information about the frequency of drug use depending on different demographic and psychological attributes. It contains 1885 observations on 32 variables, whereas one is the ID and 19 can be considered as target variables as they contain information on the frequency of drug use on different kind of drugs. We decided to only work on one specific drug (cannabis). Therefore, the dataset consists of 12 features. The features are both numerical and categorical. The target variable has seven different values (never used the drug - used it yesterday), therefore it can be considered as a multiclass problem. The target variable is unequally distributed as most people either never consumed cannabis or used it within the last day (both > 20%). In between the values vary about 10%. Most people are from the UK or the USA and are rather young (< 25 years). Almost all participants have a degree, indicating that the study was conducted at a university. The psychological attributes are approximately normally distributed. There are only few outliers and no missing values. Beside that there are no peculiarities. A detailed description of the dateset can be found in Assignment 0.

1. Cancer Dataset

This Data Set contains information about different properties of breast cancer and if a recurrence event occurred or not. It contains 285 observations on 32 variables, whereas one is the ID and one is the target variable "class". Therefore, the dataset consists of 30 features. All features are floating point values and the target variable has two different values. Therefore, it can be considered as a binary classification problem.  
The target variable is slightly unequally distributed as the proportion for B is about 66%. Generally, all features have similar distribution patterns with several outliers to the top. There are only very few extreme outliers. Beside that there are no peculiarities.

* 1. Kick Dataset
  2. Amazon Dataset

The decision to work on these datasets (Drugs Dataset and Kick Dataset) was based on their variety of characteristics. They differ in the number of samples, features, target classes and in presence of missing values. Therefore, they fit perfectly to analyse the performance of different algorithms. The different characteristics of the datasets are summarized in Tab. 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tab. 1. Dataset characteristics** | | | | | |
| **Name** | **Number of samples** | **Number of features** | **Data types of features** | **Number of target classes** | **Presence of missing values** |
| Drugs Dataset | 1.885 | 12 | Categorical and numerical | 7 | None |
| Kick Dataset | 72.983 | 33 | Categorical, numerical and date | 2 | 149.271 present in 20 features |
| Cancer Dataset | 285 | 30 | Numerical | 2 | None |
| Amazon Dataset |  |  |  |  |  |

Classifiers and Performance Measurements

The three classifiers were primarily assessed by accuracy and its standard deviation. On top of that precision and recall were considered. These metrics were calculated after performing a 10-fold cross validation. Another important performance measurement was the time it took to fit the model and to score predictions. Why these classifiers and performance measurements were chosen is stated below.

1. Classifiers

As mentioned above Naive Byes, K-Nearest-Neighbours and Random Forest were investigated. The decision to analyse these algorithms is based on different aspects. One of the main reasons is their difference in underlying algorithmic methods. Naive Bayes is a so-called probabilistic model and is based on Bayes’ theorem. Conditional probabilities are calculated behind the scenes on which predictions are based. K-Nearest-Neighbours on the other hand does not even fit a model but looks for similar observations in the dataset and predicts according to the k most similar observations. Once the neighbours are found majority voting decides about the prediction. Random Forest is based on decision trees where a specific set of decision rules decide on the prediction. A certain number of decision trees is calculated where each does not include all features but a random subsample. Prediction is again determined by majority voting of all single decision tree predictions. Another reason why these classifiers were chosen is the fact that these are very well-known algorithms and their fundamental ideas are also influencing other techniques. Also, the question appears if very simple algorithms like Naive Bayes and K-Nearest-Neighbours can compete with more modern and sophisticated approaches like Random Forest. On top of that these algorithms can be adapted by a number of parameters.

1. Parameter Settings

In the experiments we varied parameters for each algorithm. For Naive Bayes we decided to vary the assumptions made regarding the distribution of the conditional probabilities. Gaussian, Bernoulli and Multinomial distribution assumptions were made. For K-Nearest-Neighbours we obviously decided to vary the number of neighbours. In addition to that we tested two different kind of algorithms to compute the nearest neighbours. Firstly, we used brute-force search, where all data points are considered as neighbours. Secondly, KDTree was tested, where the dataset gets split into several subspaces. For predictions only neighbours within the subspace where the unseen data is located is considered. This speeds up the prediction time what might be important for K-Nearest-Neighbours as it does not build a model beforehand. Moreover, we used both distance and uniform weights for the voting procedure. Regarding Random Forest we mainly varied the number of trees but also the functions to measure the quality of a split. Precisely gini index and entropy was used. In addition to that we varied the minimum number of samples required to split a node. This causes the trees to not perfectly fit every single data point and therefore avoids overfitting. For each classifier we tried different pre-processing steps. For example, Z-score standardisation and one hot encoding was conducted and sometimes the dataset was discretized.

1. Performance measurements

The most important metric obviously is accuracy. It measures the ratio of correct predictions to the total number of predictions made, hence the higher the better. This can be considered as the overall performance of the model and is therefore obligatory to consider. Still accuracy can be misleading if the frequency for each class in the target variable is unequally distributed. Therefore, accuracy needs to be treated with caution. To see if accuracy is stable standard deviation is needed especially to compare results. This measure gives an insight on how likely it is to achieve such an accuracy again. Here a low standard deviation is better as this means that the result is reliable. To get a deeper insight into the models’ prediction behaviour recall and precision is considered as well. Recall indicates how many percent of one specific class was predicted correctly. This can be interpreted as conditional probability of the prediction being true if the class is actually true. Recall gives an insight on how well the model performs on one specific class and helps to counter the misleading aspect of the accuracy described above. Precision measures how many percent of predictions of one specific class were actually correct. Again, this can be interpreted as conditional probability of the class actually being true if the prediction for the class was true. In combination these performance measurements can be very powerful if interpreted correctly. All metrics were calculated by averaging them after a 10-fold cross validation. This is one of the most common and sufficient ways to validate predictive machine learning models. Lastly runtimes are considered as well to assess the different classifiers. Regarding runtime one have to keep in mind that this is depending on the machine the experiments are run on, but growth trends can be considered as consistent.

Classifications

All four algorithms were used to predict the same scenarios. For the drugs dataset predictions were done on the frequency of cannabis use. The target variable has seven different values like “Never Used”, “Used in Last Year” and “Used Last Day”. Features are either psychological metrics or demographical attributes. Regarding cancer dataset predictions were done on whether a recurrence event occurred after a breast cancer disease. Different cancer attributes are taken into account. The kick dataset is used to figure out if a used car bought at an action can be sold to customers or not. Features are different car attributes like age, model and producer.

Similar description for Amazon data set. And additional information on kick (which features will be used?). Aleks said she wont use all features.

1. Naive Bayes
   1. Drugs Dataset

If Guassian Naive Bayes is applied, one has to keep in mind that categorical datatypes are treated as continuous variables. If Multinomial or Bernoulli Naive Bayes is used continuous variables are treated as categorical. Obviously, this leads to a bad performance of the classifier when used on the original data with an accuracy of 36.69% (5.79) for Gaussian, 36.27% (4.38) for Multinomial and 34.35% (3.57) for Bernoulli. But this is also due to the nature of the dataset as described in K-Nearest-Neighbours below. One solution for this problem might be to discretize the numerical features in five equally distributed classes (20% quantiles). This increases the accuracy of both Multinomial and Bernoulli algorithms whereas the performance for Gaussian decreased minimally. When all variables are discretized and one hot encoded Gaussian Naive Bayes falls far back with an accuracy of 19.09% (1.81) whereas the others perform close to 40%. Runtimes do not differ between the classifiers but with the input data.

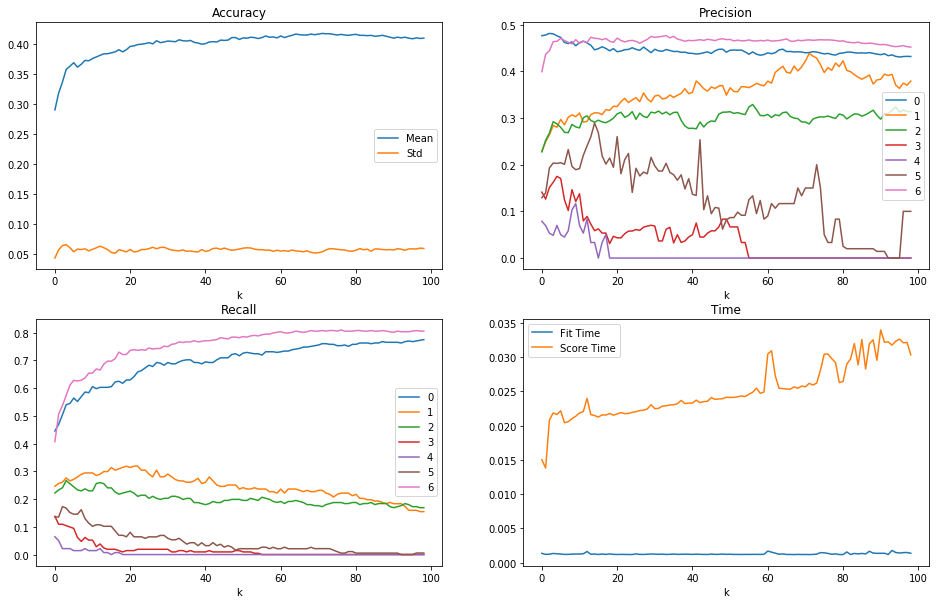
* 1. Cancer Dataset

As this dataset only contains numerical data it perfectly fits Gaussian Naive Bayes. For both Multinomial and Bernoulli the features need to be discretized again. This is also shown in the experiments. Gaussian performed best on the original numerical and the others on the one-hot encoded dataset. Discretization was again done by cutting each feature in five equally distributed classes (20% quantiles). This dataset was then used to create a one hot encoded dataset. The following matrix shows the accuracies with standard deviation for all algorithms. Runtimes did not vary.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Gaussian NB** | **Multinomial NB** | **Bernoulli NB** |
| **Source Data** | 96.54% (3.08) | 90.86% (3.25) | 66.33% (1.05) |
| **Discretized Data** | 94.09% (5.33) | 75.76% (7.87) | 75.73% (5.49) |
| **One hot encoded Data** | 93.05% (6.14) | 94.80% (4.42) | 94.80% (4.42) |

1. K-Nearest-Neighbours
   1. Drugs Dataset

Overall the Classifier did not perform very well on this dataset with an accuracy varying from 25-40%. One reason might be the unequal distribution of target classes as mentioned in the dataset description. When taking a closer look to accuracy and recall one realises that especially recall is high (up to 80% for a higher number of neighbours) for “Never Used” and “Used Last Day” but very low for the other values. This is also true for precision, but the value varies only about 40% for both classes. This indicates that this algorithm mainly decides on whether someone never took the drug or did it within the last day. This means a bad performance for the model as almost 50% of the observations fall within these two classes anyway. This is an issue for all classifiers tested. In general recall for the two classes and accuracy increase with a higher number of neighbours and precision slightly decreases. Both fitting time and score time were very fast. The best results were achieved when using min-max scaling on numerical and one hot encoding for categorical data and applying distance weights (Fig. 1.). But this model also delivers a higher standard deviation for accuracy. The classifier performed very similar when KDTree was used to determine the neighbours. Interestingly it took more time to fit the model using KDTree but also to conduct a prediction. Generally, the performance increased with a higher number of neighbours and improved when features were Z-scaled. The following table shows some representative experiments.

  
Fig 1: kNN on one hot encoded (categorical) and min-max scaled data with distance weights

|  |  |  |  |
| --- | --- | --- | --- |
| **Setting** | **Accuracy** | **Standard Deviation** | **Runtime** |
| Data: source  Weights: uniform  Algorithm: brute | 24.94% (k = 2)  30.48% (k = 10)  35.88% (k = 50)  36.52% (k = 100)  logarithmic growth | 3.33 (k = 2)  3.27  3.92  3.66  constant | Fit Time:  < 0.0025  constant  Score Time:  0.015 – 0.025  linear growth |
| Data: min-max  Weights: distance  Algorithm: brute | 29.66% (k = 2)  37.12% (k = 10)  39.19% (k = 50)  38.28% (k = 100)  logarithmic growth  slightly decreases for k > 40 | 3.63  5.03  4.96  4.46 | Fit Time:  < 0.0025  constant  Score Time:  0.015 – 0.03  linear growth |
| Data: dummy (for categorical variables) and min-max  Weights: distance  Algorithm: brute | 28.95% (k = 2)  37.19% (k = 10)  41.00% (k = 50)  40.99% (k = 100)  logarithmic growth | 4.25  5.79  5.63  5.84 | Fit Time:  < 0.0025  constant  Score Time:  0.015 – 0.035  linear growth |

* 1. Cancer Dataset

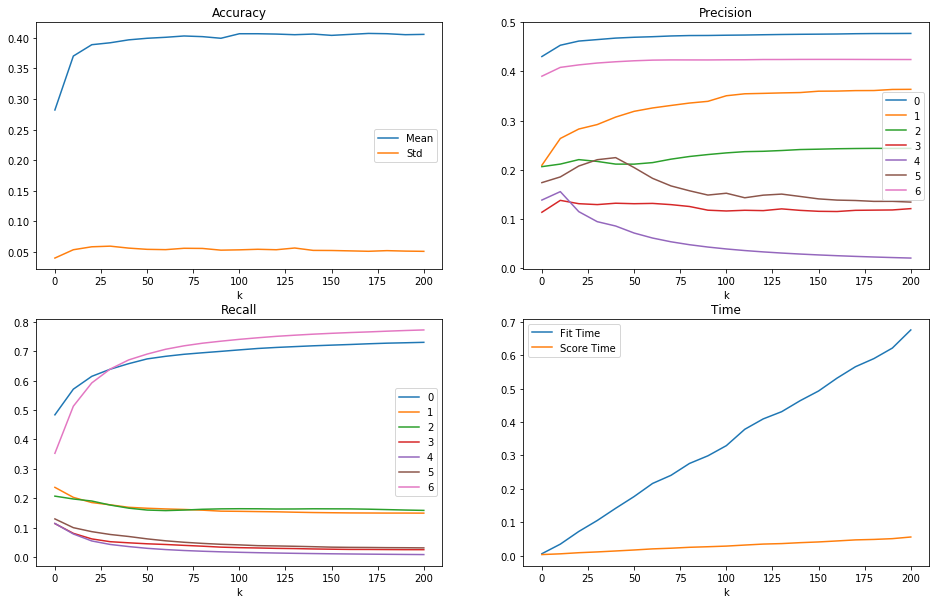
K-Nearest-Neighbours perfectly fit this dataset as all variables are numeric. In all experiments the classifier achieved an accuracy greater than 90%. Anyway, one can see a decent impact of the pre-processing here as performance increased when the data was scaled. Opposed to the Drugs Dataset the accuracy decreased with a larger number of neighbours but increased for up to 13 neighbours. When distance weights were used accuracy was greater and did not decrease as rapidly for a higher number of neighbours. Surprisingly using KDTree to determine the neighbours did not have an impact on accuracy and a minor impact on runtime. For all experiments recall for class B was greater than for class A. This might be again due to an unequal distribution of both classes (66% for B). Precision is higher for A. The gap in recall and precision increases with the number of neighbours. The best results were achieved using min-max scaled data and distance weights with 10-15 neighbours. The following table again shows some representative experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Setting** | **Accuracy** | **Standard Deviation** | **Runtime** |
| Data: source  Weights: uniform  Algorithm: brute | 92.25% (k = 2)  92.28% (k = 10)  92.26% (k = 50)  89.44% (k = 100) | 4.14  3.44  3.78  3.59 | Fit Time:  0.0015  constant  Score Time:  0.003 – 0.005  linear growth |
| Data: source  Weights: distance  Algorithm: brute | 93.67% (k = 2)  92.98% (k = 10)  92.25% (k = 50)  92.26% (k = 100) | 4.07  3.48  4.94  3.78 | Fit Time:  0.0015  constant  Score Time:  0.003 – 0.004  linear growth |
| Data: min-max  Weights: uniform  Algorithm: brute | 96.87% (k = 2)  98.28% (k = 10)  94.79% (k = 50)  89.80% (k = 100) | 3.26  2.31  2.89  4.27 | Fit Time:  0.001  constant  Score Time:  0.003 – 0.005  linear growth |
| Data: min-max  Weights: distance  Algorithm: brute | 97.56% (k = 2)  98.62% (k = 10)  95.43% (k = 50)  91.55% (k = 100) | 3.50  2.29  2.25  3.62 | Fit Time:  0.001  constant  Score Time:  0.003 – 0.004  linear growth |

1. Random Forest
   1. Drugs Dataset

Random Forest performed similarly as the other classifiers. Again, precision and recall show the same pattern as mentioned above (Fig. 1. and Fig. 2.). All experiments run on Random Forest delivered similar results. Only the number of estimators had a bigger impact as accuracy increased logarithmically with a higher number of trees. Surprisingly standard deviation did not decrease. As expected, the minimum sample size to split a node only had an impact on models with a small number of trees as it avoids overfitting. Different scaling methods did not affect the results. Accuracy increased with a higher number of estimators so did runtimes. Especially the time to fit the model increased linearly. The following table again shows some representative experiments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Setting** | **Accuracy** | **Standard Deviation** | **Runtime** |
| Data: source  Criterion: Gini  Minimum Sample Split: 2 | 25.10% (n = 1)  34.53% (n = 10)  36.39% (n = 20)  38.02% (n = 50)  38.40% (n = 100)  38.93% (n = 200) | 2.81  4.49  4.91  5.62  5.42  5.04 | Fit Time:  0.01 – 0.6  linear growth  Score Time:  0.01 – 0.06  linear growth |
| Data: source  Criterion: Gini  Minimum Sample Split: 10 | 28.63% (n = 1)  37.23% (n = 10)  38.29% (n = 20)  39.50% (n = 50)  39.61% (n = 100)  40.25% (n = 200) | 4.16  5.65  4.57  4.84  5.04  5.30 | Fit Time:  0.4 – 0.5  linear growth  Score Time:  0.01 – 0.06  linear growth |
| Data: source  Criterion: Entropy  Minimum Sample Split: 2 | 27.79% (n = 1)  34.13% (n = 10)  36.64% (n = 20)  38.06% (n = 50)  39.08% (n = 100)  39.51% (n = 200) | 3.50  4.28  5.56  4.00  4.59  5.19 | Fit Time:  0.01 – 0.85  linear growth  Score Time:  0.01 – 0.08  linear growth |
| Data: source  Criterion: Entropy  Minimums Sample Split: 10 | 28.20% (n = 1)  37.02% (n = 10)  38.89% (n = 20)  39.93% (n = 50)  40.68% (n = 100)  40.57% (n = 200) | 3.96  5.32  5.81  5.39  5.31  5.06 | Fit Time:  0.01 – 0.7  linear growth  Score Time:  0.01 – 0.07  linear growth |

  
Fig 2: Random Forest on source data with entropy as criterion and 10 as minimal sample split

* 1. Cancer Dataset

Again, Random Forest performs similarly with all parameter settings. As expected, scaling methods did not have a significant impact. Overall the accuracy varied from 94% up to almost 99%. Surprisingly the number of estimators only had a minor impact in this dataset on accuracy and standard deviation. The minimum sample size to split a node also did not even significantly affect models with a small number of trees. Again, there is the same pattern for recall and precision as described above for the cancer dataset. Runtime increases linearly, especially the time to fit the model increases and takes up to 0.3 seconds.

|  |  |  |  |
| --- | --- | --- | --- |
| **Setting** | **Accuracy** | **Standard Deviation** | **Runtime** |
| Data: source  Criterion: gini  Minimum Sample Split: 2 | 96.85% (n = 1)  96.17% (n = 10)  97.91% (n = 20)  97.19% (n = 50)  97.56% (n = 100)  97.91% (n = 200) | 2.42  2.89  2.80  2.63  2.73  2.34 | Fit Time:  0.01 – 0.30  linear growth  Score Time:  0.01 – 0.025  linear growth |
| Data: source  Criterion: gini  Minimum Sample Split: 10 | 96.46% (n = 1)  97.56% (n = 10)  97.56% (n = 20)  97.53% (n = 50)  97.56% (n = 100)  97.56% (n = 200) | 2.78  2.73  3.14  2.75  2.73  2.73 | Fit Time:  0.01 – 0.27  linear growth  Score Time:  0.01 – 0.04  linear growth |
| Data: source  Criterion: entropy  Minimum Sample Split: 2 | 96.85% (n = 1)  96.50% (n = 10)  97.54% (n = 20)  97.88% (n = 50)  97.91% (n = 100)  98.25% (n = 200) | 3.19  2.70  2.80  2.37  2.37  2.37 | Fit Time:  0.01 – 0.25  linear growth  Score Time:  0.01 – 0.04  linear growth |
| Data: source  Criterion: entropy  Minimum Sample Split: 10 | 95.12% (n = 1)  96.84% (n = 10)  97.19% (n = 20)  97.19% (n = 50)  97.56% (n = 100)  97.56% (n = 200) | 4.68  2.90  2.63  2.63  2.73  2.73 | Fit Time:  0.01 – 0.25  linear growth  Score Time:  0.01 – 0.04  linear growth |

Summary

In general, the information content of the dataset seems to make the biggest difference as all algorithms performed similarly on the datasets.

1. Naive Bayes

The experiments approved what was expected from theory but still some unexpected results occurred. Gaussian Naive Bayes works best for numerical data and also surprisingly well on multileveled data but performs badly on one-hot encoded data. Multinomial and Bernoulli Naive Bayes performed better on categorical data. Both achieved higher accuracies and lower standard deviations when the features where discretized or one-hot encoded. Surprisingly Multinomial Naive Bayes sometimes works better for binary or numerical variables as for multiclass variables. Runtimes increase with the size of the dataset but still stay small.

1. K-Nearest-Neighbours

For this algorithm the accuracy improved with more neighbours considered up to a certain number. As expected, this ceiling is higher for datasets with a bigger sample size. With additional neighbours the performance might decrease again. Surprisingly the standard deviation of accuracy seemed to be independent from the number of neighbours. Scaling methods are highly recommended as it equalizes the impact of features. According to the experiments run it is best to apply min-max scaling to numerical data and one-hot encoding to categorical data. Weighting the considered neighbours by distance seem to slightly improve the performance. Surprisingly KDTree did not have an impact on performance. In all experiments the score time increased linearly whereas fitting time was constant. This is due to the reason that this algorithm actually does not build a model beforehand.

1. Random Forest

This algorithm achieved similar accuracies as the other classifiers but has advantages regarding standard deviation. Similar to K-Nearest-Neighbours the model becomes better with a higher number of estimators up to a certain ceiling. Opposed to KNN the model does not become worse by adding additional trees. Surprisingly drugs dataset took more estimators to ceil as cancer dataset. This is contrary to the fact that drugs dataset has a lower dimension. Criterion does not significantly influence the results. It takes some time to fit the models especially for a higher number of trees. Surprisingly in general the standard deviation did not decrease with a higher number of estimators. Minimum sample split only had an impact on drugs dataset for small number of trees. Scaling had no impact.

Overall the conclusion can be drawn that most classification performance is mainly dependent on the information content of the dataset. Although naive, Naive Bayes can perform pretty well if applied correctly by assuming the correct distribution of conditional probabilities. This is also true for K-Nearest-Neighbours where one has to find the appropriated number of neighbours. Equalizing the impact of the features by scaling is essential here. Even when those simple algorithms are able to compete with the random forest this algorithm might be the best as it is very stable and easy to apply. Scaling the data is not required and to be sure one can use rather use more estimators. There are also tiny advantages for random forest regarding standard deviation.