DV2599 – Assignment 2

Simon Lindqvist
Blekinge Institute of Technology
Karlskrona, Blekinge
siln22@student.bth.se

Abdalrahman Mohammed
Blekinge Institute of Technology
Karlskrona, Blekinge
abmm22@student.bth.se

I. Introduction

Spam detection is an important machine learning task that can be applied to filter for example unwanted emails. This assignment compares the performance of three supervised learning algorithms for spam classification using the Spambase dataset. The comparison focuses on training time for computational performance, and accuracy as well as F1 score for their predictive performance. Using stratified ten-fold cross-validation, the algorithms are evaluated and then the Friedmann and Nemenyi statistical tests are used to determine any significant differences. The goal is to understand which algorithms are the most effective and efficient for spam detection tasks.

II. Classifires

A. Decision Tree

Decision Tree is a supervised learning algorithm used for classification and regression tasks. It splits the dataset into subsets based on feature values, creating a tree-like structure of decision rules. Decision Trees are easy to interpret and can handle both categorical and numerical data but may be overfit without proper pruning.

B. K-Nearest Neighbor (KNN)

KNN is a simple classification algorithm that predicts the class of a data point based on the majority class of its "k" closest neighbors in the feature space. It assumes similar instances exist close to each other and work well for smaller datasets but can be computationally expensive for large ones.

C. Naïve Bayes

Naïve Bayes is a probabilistic classification algorithm based on Bayes' Theorem. It assumes that features are conditionally independent given the class label. Despite its simplicity, Naïve Bayes is effective for text classification and other applications, especially when feature independence approximations hold true.

III. Statical Analysis

A. Friedman Test

The Friedman test is a non-parametric statistical test used to detect differences in performance across multiple algorithms or treatments over several datasets or conditions. It ranks the performance of each algorithm for each dataset, then assesses whether the average ranks significantly differ. It is commonly used in machine learning to compare multiple models when the data does not meet parametric test assumptions.

1) Friedman's Test Formula

The formula for Friedman's test is as follows:

$$\overline{R} = \frac{1}{nk}\sum_{i,j} R_{ij} = \frac{k+1}{2} \quad (1)$$

$$n\sum_j (R_j - \overline{R})^2 \quad (2)$$

$$\frac{1}{n(k-1)}\sum_{i,j}(R_{ij} - \overline{R})^2 \quad (3)$$

Where n is the number of data sets, k is the number of algorithms, $\overline{R}$ is the average rank, R_j is the average rank of the j-th algorithm, and R_{ij} is the rank of the j-th algorithm of the i-th data set [1].

2) Friedmann test and critical value

Calculating the critical difference value for the Friedmann tests was done by using table 2 for N values less than 15, with alpha = 0.05, k = 3, and n = 10, as compiled by Martin *et al* [2].

B. Nemenyi Test

The Nemenyi test is a post-hoc statistical test used after the Friedman test when significant differences are detected. It identifies which pairs of algorithms differ significantly by comparing their average ranks. The test uses a critical difference threshold to determine significance, providing a clear view of pairwise algorithm performance comparisons.

1) Nemenyi's Test Formula

The formula for Nemenyi's test is as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \quad (4)$$

Where CD is critical difference, and $q_\alpha$is the studentized range q for alpha = 0,05. k = 3, and infinite degrees of freedom, which is 2,343. K is the number of algorithms, and n is the number of data sets. [3]

IV. Method

The data was split into test and training data using the stratified k-fold split method and a stratified k-fold evaluation was performed where training time, f1-score, and accuracy was saved for evaluation. This was done for each of the different chosen classifiers. The results were then compiled into a table using the same layout as example 12.4 in the coursebook [4].

A Friedmann test was then performed by ranking the results per column, calculating the results for formula (1), (2), and (3) to obtain the Friedmann statistic, and calculating any significant difference using the critical value for comparison against the Friedmann statistic. The results were

then compiled into a table using the same layout as example 12.8 in the coursebook [3]. This was done for each statistic time, accuracy, and f1-score.

Lastly a Nemenyi test was performed. The critical difference was calculated as in formula (4) and was used together with the difference between all the classifiers to determine if there was a significant difference between the different classifiers or not.

V. Results

A. Validation and Friedmann tanking tables

TABLE I. Cross Validation Results for accuracy

Fold	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.826	0.807	0.907
2	0.841	0.787	0.891
3	0.826	0.793	0.911
4	0.811	0.830	0.911
5	0.846	0.813	0.917
6	0.826	0.848	0.920
7	0.839	0.822	0.928
8	0.780	0.800	0.922
9	0.780	0.796	0.891
10	0.826	0.787	0.926
avg	0.820	0.808	0.912
std	0.022	0.019	0.012

TABLE II. Cross Validation Results for F1-score

Fold	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.810	0.753	0.883
2	0.827	0.732	0.864
3	0.815	0.740	0.886
4	0.800	0.780	0.886
5	0.830	0.758	0.893
6	0.814	0.800	0.901
7	0.826	0.776	0.908
8	0.769	0.753	0.904
9	0.775	0.749	0.864
10	0.813	0.718	0.905
avg	0.808	0.756	0.889
std	0.020	0.023	0.015

TABLE III. Cross Validation Results for training time (s)

Fold	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.007	0.004	0.057
2	0.006	0.002	0.056
3	0.006	0.004	0.066
4	0.006	0.003	0.070
5	0.006	0.005	0.062
6	0.005	0.003	0.058
7	0.005	0.003	0.051
8	0.005	0.002	0.061
9	0.006	0.003	0.061
10	0.006	0.003	0.060
avg	0.006	0.003	0.060
std	0.001	0.001	0.005

TABLE IV. Friedmann rankings table for accuracy

Data set	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.826 (2)	0.807 (3)	0.907 (1)
2	0.841 (2)	0.787 (3)	0.891 (1)
3	0.826 (2)	0.793 (3)	0.911 (1)
4	0.811 (3)	0.830 (2)	0.911 (1)
5	0.846 (2)	0.813 (3)	0.917 (1)
6	0.826 (3)	0.848 (2)	0.920 (1)
7	0.839 (2)	0.822 (3)	0.928 (1)
8	0.780 (3)	0.800 (2)	0.922 (1)
9	0.780 (3)	0.796 (2)	0.891 (1)
10	0.826 (2)	0.787 (3)	0.926 (1)
avg rank	2.4000	2.6000	1.0000

TABLE V. Friedmann rankings table for f1-score

Data set	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.810 (2)	0.753 (3)	0.883 (1)
2	0.827 (2)	0.732 (3)	0.864 (1)
3	0.815 (2)	0.740 (3)	0.886 (1)
4	0.800 (2)	0.780 (3)	0.886 (1)
5	0.830 (2)	0.758 (3)	0.893 (1)
6	0.814 (2)	0.800 (3)	0.901 (1)
7	0.826 (2)	0.776 (3)	0.908 (1)
8	0.769 (2)	0.753 (3)	0.904 (1)
9	0.775 (2)	0.749 (3)	0.864 (1)
10	0.813 (2)	0.718 (3)	0.905 (1)
avg rank	2.0000	3.0000	1.0000

TABLE VI. Friedmann rankings table for training time

Data set	*Naive Bayes*	*KNN*	*Decision Tree*
1	0.007 (2)	0.004 (1)	0.057 (3)
2	0.006 (2)	0.002 (1)	0.056 (3)
3	0.006 (2)	0.004 (1)	0.066 (3)
4	0.006 (2)	0.003 (1)	0.070 (3)
5	0.006 (2)	0.005 (1)	0.062 (3)
6	0.005 (2)	0.003 (1)	0.058 (3)
7	0.005 (2)	0.003 (1)	0.051 (3)
8	0.005 (2)	0.002 (1)	0.061 (3)
9	0.006 (2)	0.003 (1)	0.061 (3)
10	0.006 (2)	0.003 (1)	0.060 (3)
avg rank	2.0000	1.0000	3.0000

B. Friedmann and Nemenyi tests

The critical value for Friedmann was 6.2, meaning that exceeding this with the calculated Friedmann statistic leads to null-hypothesis rejection, i.e. there is a significant difference between models. For accuracy the Friedmann statistic was 15.2, for f1-score it was 20.0, and for time it was 20.0. All of these are above the critical value, so all reject the null hypothesis and there is significant difference.

The critical difference for Nemenyi was calculated to 1.0478, meaning exceeding this with the difference between two models means a significant difference. For accuracy Naive Bayes versus Decision Tree and KNN versus Decision were significantly different, and for both f1-score and time only KNN versus Decision Tree were significantly different.

VI. References

[1] P. Flach, “Machine Learning: The Art and Science of Algorithms that Make Sense of Data,” Cambridge University Press, p.355, 2012

[2] L. Martin, R. Leblanc, and N. K. Toan, “Tables for the Friedman rank test,” *Canadian Journal of Statistics*, vol. 21, no. 1, pp. 39–43, 1993

[3] P. Flach, “Machine Learning: The Art and Science of Algorithms that Make Sense of Data,” Cambridge University Press, p. 356, 2012

[4] P. Flach, “Machine Learning: The Art and Science of Algorithms that Make Sense of Data,” Cambridge University Press, p. 350, 2012