Assignment 2

Arlind Iseni
Email: aris20@student.bth.se

Alexander Jamal
Email: aris20@student.bth.se

1. Introduction

The spam detection feature of e-mail systems are crucial. Three algorithms were used in this study to compare how well they classified emails as spam or not. To compare the algorithms, we will be conducting two statistical tests, namely Friedman and Nemenyi test.

The classifiers chosen were Support Vector Machine (SVM), AdaBoost and Random Forest. Support Vector Machine (SVM) objective is to find a hyperplane in an N-dimensional space that distinctly classifies the data points [1]. AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance [2]. Random forest is a commonly used machine learning algorithm, which combines the output of multiple decision trees to reach a single result [3].

Our data set consist of aggregated statistical summaries of e-mails in a big cohort.

2. Method

2.1. Data exploration

Upon examination we find a right-skewed distribution for all but the target feature. The target column has a 60/40-procent ratio of ham and spam making it a fairly balanced data set. Furthermore, we identified 391 duplicate instances.

2.2. Data transformation

Due to support vector machines being very sensitive to outliers [4] we discretized our data set so that we'd get more comparable performance results between the algorithms. Discretization was performed with the KBinsDiscretizer method from Sklearn [5] with 7 bins and kmeans binning technique. Note however that the discretized data set was only used for training the SVM classifier. All other classifiers were trained on non-transformed data.

2.3. Computing the Friedman and Nemenyi test

The Friedman test is a non-parametric statistical test used to determine whether there are significant differences between treatments being compared in a study. The Friedman test is a rank-based test, meaning it ranks the treatments being compared and the computes a test statistic based on the ranks. If the test statistic passes a certain threshold, referred to as the critical value (determined by the number of treatments being compared and the significance level), then the null hypothesis H_o , that there are no significant differences between the treatments is rejected.

Friedman test [6]

$$(1.1)\ Mean\ rank : \bar{R} = \frac{k+1}{2};$$

$$(1.2)\ Sum\ of\ squared\ differences : n\sum_{j}(R_j - \bar{R})^2;;$$

$$(1.3)\ Sum\ of\ squared\ differences : \frac{1}{n(k-1)}\sum_{ij}(R_{ij} - \bar{R})^2;$$

$$(1.4)\ Friedman\ score\ -\ ratio\ between\ 1.2\ and\ 1.3 :$$

The Nemenyi test is a post-hoc statistical test that is conducted after completition of a study in order to compare treatments in a group. The Nemenyi test allows us to identify which treatments are significantly different from each other. With the Nemenyi test we compare the rank means of each treatment to identify which pairs of treatments that display a significant difference from each other. We conclude whether the difference between treatments is significant or not by seeing if the difference between means is greater than the critical difference.

Nemenyi test

$$(2.1)\ Critical\ difference : CD = q_\alpha\sqrt{\frac{k(k+1)}{6n}};\ [6]$$

$$(2.2)\ \sum_{i}\sum_{j}|\bar{R}_i - \bar{R}_j| > CD.\ [7]$$

2.4. Algorithms

This study has choosen the following 3 classification models to compare:

Support vector machine: fits a hyperplane in N-dimensions with maximal margin seperating instances by their class. Instances exist in geometric space with their coordinates being the features. For scenarios where no linear function can seperate the data points a non-linear function will be applied so that data points will be re-oriented in a higher dimension allowing for a hyperplane to seperate the instances.

AdaBoost: a meta-algorithm that wraps over a set of weaker classifiers are training on samples of the training data. For each iteration the trained model will be tested on the other samples (random) of the training split and for those data points, if any, that have a high error associated with them will be weighted more in the next training iteration so that the model can train on it's weaker areas.

Random forest: an ensamble technique which uses a set of decision trees and intelligent pruning algorithms for removing splits with low information-yield and a voting system so that the all predictions from the entire set of decision trees are aggregated into a final prediction.

These algorithms show great performance, are very popular [8] and don't require much-to-any hyper parameter tuning. These reasons were the motivation behind our decision.

2.5. Code and algorithm implementation

The study was performed in a virtual environment on jupyter-lab with python 3.11. Our classifier models and performance measures came from the sklearn library and the data was processed and read via the pandas package. For more complex filtering and mathmetical operations NumPy and itertools was used and lastly for measuring the training time we borrowed the time method from the standard library time.

2.6. Training and evaluation

We used stratified k-fold cross validation as our training and testing procedure.The paradigm is as following: we split our data into k-folds and use k-1 of these folds to train our data and the remaining one to test. We repeat this procedure k-1 more times without replacement. To ensure we get a fair representation of our target with each fold the folds preserve the percentage of samples for each class [6]. With this technique we get a more robust measure of performance by allowing our model to train and test on different parts of the data set. However this methodology is more computationally expensive [9].

We measure model performance with the following three metrics:

F-measure: also known as F1 score, is a performance metric used to evaluate a classification model. It is computed as the harmonic mean between precision and recall. F-measure ranges between 0 to 1, with higher values meaning better performance. It is a balanced metric that considers both precision and recall of the model.

$F1 = 2 * \frac{precision*recall}{precision+recall}, F1 \in [0, 1]$

Precision: is the fraction of true positive (TP) predictions made by the model among all positive predictions.

$precision = \frac{TP}{TP+FP}, precision \in [0, 1]$

Recall: is the fraction of true positive predictions made by the classifier among all actual positive predictions.

$recall = \frac{TP}{TP+FN}, recall \in [0, 1]$

Accuracy: the percentage of predictions that are correct.

$accuracy = \frac{TP+TN}{TP+FP+TN+FN}, accuracy \in [0, 1]$

Computational time: time taken to train algorithm.

3. Results

The algorithms performance, group statistic and rank for each fold is presented in the below tables. Note that these are rounded values. For complete values check our python notebook.

	SVM	AdaBoost	Random forest
1	0.9350 (3)	0.93450 (2)	0.9588 (1)
2	0.9326 (3)	0.9413 (2)	0.9565 (1)
3	0.9283 (3)	0.9370 (2)	0.9522 (1)
4	0.9609 (2)	0.9630 (1)	0.9522 (3)
5	0.9304 (3)	0.9348 (2)	0.9587 (1)
6	0.9435 (3)	0.9457 (2)	0.9500 (1)
7	0.9457 (2)	0.9391 (3)	0.9609 (1)
8	0.9457 (3)	0.94783 (2)	0.9652 (1)
9	0.9391 (2)	0.9261 (3)	0.9457 (1)
10	0.9217 (3)	0.9304 (2)	0.9522 (1)
Average	0.9383	0.9400	0.9552
Std	0.0112	0.0104	0.0058
Average rank	2.7	2.2	1.2

TABLE 1. ACCURACY

	SVM	AdaBoost	Random forest
1	0.9153 (3)	0.9180 (2)	0.9359 (1)
2	0.9132 (3)	0.9264 (2)	0.9448 (1)
3	0.9081 (3)	0.9192 (2)	0.9392 (1)
4	0.9508 (2)	0.9544 (1)	0.9479 (3)
5	0.9070 (3)	0.9148 (2)	0.9337 (1)
6	0.926 (3)	0.9319 (2)	0.9359 (1)
7	0.9315 (2)	0.9218 (3)	0.9392 (1)
8	0.9292 (3)	0.9322 (2)	0.9524 (1)
9	0.9222 (2)	0.9056 (3)	0.9438 (1)
10	0.8960 (3)	0.9080 (2)	0.9307 (1)
Average	0.9199	0.9232	0.9404
Std	0.0155	0.0141	0.007
Average rank	2.7	2.1	1.2

TABLE 2. F1-SCORE

	SVM	AdaBoost	Random forest
1	0.6522 (2)	0.4799 (1)	0.8543 (3)
2	0.4660 (1)	0.4678 (2)	1.0299 (3)
3	0.4923 (2)	0.4803 (1)	0.9217 (3)
4	0.4581 (1)	0.4777 (2)	0.9284 (3)
5	0.4696 (2)	0.4502 (1)	0.9095 (3)
6	0.4982 (1)	0.4997 (2)	0.8108 (3)
7	0.4859 (2)	0.4617 (1)	0.8169 (3)
8	0.4772 (1)	0.4879 (2)	0.8161 (3)
9	0.5296 (2)	0.4937 (1)	0.8134 (3)
10	0.4527 (1)	0.5061 (2)	0.8488 (3)
Average	0.4982	0.4805	0.8750
Std	0.0586	0.01730	0.0716
Average rank	1.5	1.5	3.0

TABLE 3. Training time

The computed Friedman statistic for each table is approximately: 15.4 for accuracy, 11.4 for F1-score and 20.0 for time, which compared to the critical value, being 7.8 for k = 3 and n = 10 at the α = 0.05 [6], the Friedman statistic of all experiments is larger, resulting in all of them rejecting the null-hypotheses. To figure out which algorithms differed significantly from eachother for each respective table, we conducted a Nemenyi test. We calculated the critical distance with formula (2.1): resulting in $\approx$ 1.149. Next we computed equation (2.2) for each algorithm and table. In the accuracy experiment we find SVM and Random forest displaying a significant difference. Further, we looked into the F1-table, which gave us the same result as in accuracy table. Finally, we took a look at the training time table, where AdaBoost and Random Forest showed a difference greater than the critical difference.

4. Conclusion

The Friedman and Nemenyi test gave us a reliable and robust method for comparing performance between different machine learning algorithms. We see from the experiment results that the Random forest algorithm had on average a better F1-score and accuracy, however requiring more training time. On the flipside we see the AdaBoost algorithm requiring the least amount of training time. The statistical tests give us a valuable metric for decision making however context is needed for decision making. If one can afford having a more computationally expensive algorithm then, accorind to our tests, The random forest classifier serves this scenario best. If computational resources are scarce then an algorithm such as AdaBoost might lend you the best results with respect to cost.

References

[1] “Support Vector Machine Algorithm,” GeeksforGeeks, Jan. 20, 2021. https://www.geeksforgeeks.org/support-vector-machine-algorithm/

[2] Wikipedia Contributors, “AdaBoost,” *Wikipedia*, Nov. 01, 2019. https://en.wikipedia.org/wiki/AdaBoost

[3] IBM Cloud Education, “What is Random Forest?,” *www.ibm.com*, Dec. 07, 2020. https://www.ibm.com/cloud/learn/random-forest

[4] T. Kanamori, S. Fujiwara, and A. Takeda, “Breakdown Point of Robust Support Vector Machine,” arXiv:1409.0934 [cs, stat], Sep. 2014, Accessed: Dec. 20, 2022. [Online]. Available: https://arxiv.org/abs/1409.0934

[5] “sklearn.preprocessing.KBinsDiscretizer,” scikit-learn. https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html (accessed Dec. 20, 2022).

[6] P. A. Flach, “Chapter 12,” in Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012

[7] “Nemenyi Test after KW Test — Real Statistics Using Excel,” www.real-statistics.com. https://www.realstatistics.com/one-way-analysis-of-variance-anova/kruskal-wallis-test/nemenyi-test-after-kw/ (accessed Dec. 20, 2022).

[8] S. Tavasoli, “Top 10 Machine Learning Algorithms You Need to Know in 2020,” Simplilearn.com, Nov. 09, 2016. https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article

[9] ”What is Cross-validation?,” Outerbounds. https://outerbounds.com/docs/what-is-cross-val/ (accessed Dec. 20, 2022).