Assignment 2

Sebastian Bengtsson
DVAMI20h
BTH
Karlskrona, Sweden
sebe20@student.bth.se

Marius Stokkedal
DVAMI20h
BTH
Karlskrona, Sweden
matk20@student.bth.se

I. Introduction

Comparing the computational and predictive performance of SVM, KNN and Decision Tree.

II. Information about the used classifiers

A. *Support Vector Machine*

Support Vector Machine (SVM) tries to find the best separating line (or hyperplane) between the datapoints of different classes. It does this by finding the highest margin by the datapoints and the separator i.e., the support vectors. If it can't find a good separator it can add another dimension so the data can be linearly separable. It can then classify new datapoints by seeing on which side the new datapoint ends up.

There are a few parameters that were used, first is the C-value which was set to 0.05. The C-value is a sort of penalty of the error, i.e., the cost of misclassification. Higher C value means potentially higher accuracy but also higher risk of overfitting. The kernel was set to linear which means that as mentioned the data can be transformed into a higher dimension until it can be linearly separable. For all this to work better the data was also standardized.

B. *K-nearest neighbors*

K-nearest neighbors (KNN) is a supervised algorithm. KNN tries to classify datapoints by calculating the distance from a new datapoint to every datapoint in the training set and then classify the new datapoint as whatever the majority of the K nearest neighbors are.

As a user you need to decide the K-value and for this report it was set to 3 as it gave the highest accuracy, compared to other K-values that were tried.

C. *Decision Tree*

Decision Tree (DT) classifies datapoints by letting a new datapoint traverse through the tree structure. At every node there are criteria that decides what node the datapoint will continue traversing towards; when it reaches a leaf-node it will get the classification.

To build the tree the entropy is calculated of the dataset. Entropy is a measure of the impurity of a set of examples. The higher the entropy, the more impure the set is. Based on this you can calculate the information gain for every attribute. Information gain is the change in entropy when going from one state to another. Based on this the element with the highest information gain is selected as the top node. You repeat this process and build the entire tree.

III. A brief description of the Friedman and Nemenyi tests along with the formulas [1]

A. *Friedman test*

The data from the results must be treated like matrix with n rows and k columns. The data in each row will be ranked in relation to each other based on their value; if multiple instances have the same value, they will be given mean value of all the ranks that otherwise would have been assigned to them. The data in the matrix will be replaced with the result of the ranking operation.

After the ranking is done each row's average rank (R_j) is calculated as well as the total average rank ($\bar{R}$). The Friedman statistic is calculated by the ratio between the spread between the average ranks for the algorithms and the spread of the ranks overall. These quantities are calculated with the following equations:

$$n\sum_{j=1}^{n}(R_j - \bar{R})^2 \quad (1),$$

$$\frac{1}{n(k-1)}\sum_{i=1}^{k}(\sum_{j=1}^{n}(R_{ij} - \bar{R})^2) \quad (2).$$

B. *Nemenyi test*

The Nemenyi test is carried out after the Friedman test to calculate the critical distance. Here the algorithms are compared pair wise. The absolute value of the difference between two algorithm's average rank ($\bar{R}$) is compared to critical difference. In the Nemenyi test the critical difference (CD) is calculated by the following equation:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \quad (3).$$

q_α depends on the significance level α and k; in this case, with $\alpha = 0.05$ and $k = 3$, it is $q_\alpha = 2.343$.

The comparison is therefore:

$$|\bar{R}_a - \bar{R}_b| > q_\alpha \sqrt{\frac{k(k+1)}{6n}} \quad (4).$$

If the result of the comparison is true, the result exceeds the critical difference and the algorithm which performed better can be said to perform significantly better than the one with worse performance.

IV. RESULTS OF THE COMPARISON STATING WHETHER THE ALGORITHMS PERFORM SIGNIFICANTLY DIFFERENT OR NOT FROM EACH OTHER FOR EACH PERFORMANCE MEASURE

Measure: Time

Fold	KNN	SVM	DT
1	0.0233	0.3978	0.0963
2	0.0205	0.3802	0.1113
3	0.0202	0.3726	0.1027
4	0.0203	0.3880	0.1024
5	0.0213	0.4124	0.0960
6	0.0217	0.4107	0.0954
7	0.0268	0.4963	0.0964
8	0.0211	0.3850	0.0983
9	0.0227	0.3723	0.0983
10	0.0212	0.3684	0.1032
Avg.	**0.0219**	**0.3984**	**0.1000**
Stdev.	**0.0019**	**0.0358**	**0.0047**

The average rank for all algorithms ($\bar{R}$) was 2, $n\sum_{j=1}^{n}(R_j-\bar{R})^2 = 20$ and $\frac{1}{n(k-1)}\sum_{i=1}^{k}(\sum_{j=1}^{n}(R_{ij}-\bar{R})^2) = 1$. Therefor the Friedman statistic is 20.

Measure: Accuracy

Fold	KNN	SVM	DT
1	0.8503	0.9436	0.9393
2	0.8239	0.9261	0.9109
3	0.8065	0.9370	0.9391
4	0.7826	0.9391	0.9283
5	0.8196	0.9087	0.9065
6	0.8261	0.9000	0.9261
7	0.7696	0.9000	0.8891
8	0.8109	0.9413	0.9304
9	0.8087	0.9283	0.9130
10	0.8435	0.9326	0.9130
Avg.	**0.8142**	**0.9257**	**0.9196**
Stdev.	**0.0235**	**0.0159**	**0.0150**

The average rank for all algorithms ($\bar{R}$) was 2, $n\sum_{j=1}^{n}(R_j-\bar{R})^2 = 16.8$ and $\frac{1}{n(k-1)}\sum_{i=1}^{k}(\sum_{j=1}^{n}(R_{ij}-\bar{R})^2) = 1$. Therefor the Friedman statistic is 16.8.

Measure: F-measure

Fold	KNN	SVM	DT
1	0.8099	0.9274	0.9086
2	0.7781	0.9029	0.8763
3	0.7627	0.9192	0.9218
4	0.7326	0.9231	0.9180
5	0.7688	0.8793	0.8969
6	0.7765	0.8671	0.9106
7	0.7104	0.8686	0.8698
8	0.7549	0.9256	0.8967
9	0.7471	0.9091	0.8883
10	0.7978	0.9141	0.8962
Avg.	**0.7639**	**0.9036**	**0.8983**
Stdev.	**0.0280**	**0.0223**	**0.0161**

The average rank for all algorithms ($\bar{R}$) was 2, $n\sum_{j=1}^{n}(R_j-\bar{R})^2 = 15.2$ and $\frac{1}{n(k-1)}\sum_{i=1}^{k}(\sum_{j=1}^{n}(R_{ij}-\bar{R})^2) = 1$. Therefor the Friedman statistic is 15.2.

The null hypothesis in this case was "all algorithms perform equally". The critical value for k=3 and n=10 at the α=0.05 level is 7.8, which the Friedman statistic exceeds in all three measures. The null hypothesis can therefore be discarded, and further pairwise comparison is needed to evaluate which algorithms' performance differ.

V. CRITICAL DIFFERENENCE EVALUATION

The critical difference was compared to the algorithms difference for all three measurements were calculated.

Measurement	Algorithm 1	Algorithm 2	Is the difference in performance significant?
Time	KNN	SVM	**Yes**
Time	KNN	DT	No
Time	SVM	DT	No
Accuracy	KNN	SVM	**Yes**
Accuracy	KNN	DT	**Yes**
Accuracy	SVM	DT	No
F-measure	KNN	SVM	**Yes**
F-measure	KNN	DT	**Yes**
F-measure	SVM	DT	No

As seen in the table above, KNN can be assumed to perform significantly different from the others regarding both F-measure and Accuracy, with KNN performing slightly worse. Regarding Time KNN was only performing significantly different from SVM, with SVM being the slowest of the algorithms and KNN being the fastest.

VI. References

[1] P. Flach, Machine Learning - The Art and Science of Algorithms that Make Sense of Data, Cambridge University, 2012.