Machine Learning - Exercise 1: Classification

Bianca Apostolescu, Lu Chen, Matthias Glinzner

Dataset: Breast cancer

Preprocessing

For preprocessing the data set, two different scaling methods were used: Standardization and Normalization. Because the data set contains no missing values and only one categorical value (the one we're testing for), no further data modification is necessary.

To detect outliers, four different methods were employed: Isolation Forest, Minimum Covariance Determinant, Local Outlier Factor and One-Class SVM.

The difference in quality these methods make will be discussed in a later section.

Classification

K-Nearest Neighbours

The first classification method we used is *K-Nearest Neighbours.* This method's performance is solid, especially after applying the right combination of preprocessing methods; its cross-validation accuracy sits at 0.958.

When compared to the default settings, this classifier's performance for the chosen dataset can be improved by switching the weight function to *distance*. Further, experimenting with the number of neighbours led to the value 17 as optimal.

Random Forest

The second method, *Random Forest*, performs worst of the three with a cross-validation accuracy of only 0.947. However, it is most unaffected by switching scaling methods. In fact, it performs almost optimally without any preprocessing at all.

We found this method to work best with default parameters.

Multilayer Perceptron

The last method is *Multilayer Perceptron.* It outperforms the other two methods in all configurations; the cross-validation accuracy is 0.979.

In terms of parameter-tweaking, setting the solver to *lbfgs* yields the best results since the data set is relatively small. Experimenting with different settings shows further that results can be improved even more by increasing the tolerance for optimization to 0.01 and switching the activation function to *tanh*.

Performance

To measure performance we used accuracy scores and confusion matrices for the holdout method as well as accuracy for cross-validation. When used without any preprocessing or parameter-tweaking, the *Random Forest* method clearly outperforms the other two. This picture changes once optimizations take place: MLP then yields the most promising results.

The tables show further that *Random Forest*, unlike the other two methods, profits most from normalization, together with outlier detection via *Minimum Covariance Determinant.*

K-NN's results are relatively homogenous within the same scaler class; using standardization, *Local Outlier Factor* performs slightly better.

Unfortunately *MLP* doesn't converge fast enough when using normalization. For standardization though, *Isolation Forest* is the best choice.

Not surprisingly, each method yields slightly better results when validated via the holdout method; the exception being *K-NN.*

The confusion matrices show that save for *K-NN* (normalized), false negatives are 0. And even the combination mentioned above has a recall value of 0.97. In terms of false positives it is also *K-NN* that performs worst.

	Accuracy score (holdout)	Accuracy score (cross-validation)
K-NN	0.912	0.905
Random Forest	0.965	0.947
MLP	0.912	0.912

Table 1: Baseline performances (no preprocessing, default parameters).

35	0
5	17

(a) K-NN

35	0
2	20

(b) Random Forest

35	0
5	17

(c) MLP

Table 2: Confusion matrices for baseline performance.

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9473684211	0.95789
Min. Covariance Det.	0.9473684211	0.95789
Local Outlier Factor	0.9649122807	0.95789
One-Class SVM	0.9473684211	0.95789

Table 3: K-NN performance (Standardization, improved parameters).

35	0
4	18

(a) Isolation Forest

35	0
3	19

(b) Min. Covariance Det.

35	0
2	20

(c) Local Outlier Factor

35	0
3	19

(d) One-Class SVM

Table 4: Confusion matrices K-NN performance (Standardization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.8947368421	0.90877
Min. Covariance Det.	0.8947368421	0.90877
Local Outlier Factor	0.8947368421	0.90877
One-Class SVM	0.8947368421	0.90877

Table 5: K-NN performance (Normalization, improved parameters).

34	1
5	17

(a) Isolation Forest

34	1
5	17

(b) Min. Covariance Det.

34	1
5	17

(c) Local Outlier Factor

34	1
5	17

(d) One-Class SVM

Table 6: Confusion matrices K-NN performance (Normalization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9649122807	0.92982
Min. Covariance Det.	0.9473684211	0.93333
Local Outlier Factor	0.9649122807	0.9193
One-Class SVM	0.9649122807	0.92982

Table 7: Random Forest performance (Standardization, improved parameters).

35	0
2	20

(a) Isolation Forest

35	0
2	20

(b) Min. Covariance Det.

35	0
2	20

(c) Local Outlier Factor

35	0
2	20

(d) One-Class SVM

Table 8: Confusion matrices Random Forest performance (Standardization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	0.9649122807	0.94737
Min. Covariance Det.	0.9824561404	0.93684
Local Outlier Factor	0.9649122807	0.94035
One-Class SVM	0.9473684211	0.94386

Table 9: Random Forest performance (Normalization, improved parameters).

35	0
1	21

(a) Isolation Forest

35	0
1	21

(b) Min. Covariance Det.

35	0
3	19

(c) Local Outlier Factor

35	0
3	19

(d) One-Class SVM

Table 10: Confusion matrices Random Forest performance (Normalization, improved parameters).

	Accuracy score (holdout)	Accuracy score (cross-validation)
Isolation Forest	1	0.97895
Min. Covariance Det.	0.9824561404	0.97895
Local Outlier Factor	0.9824561404	0.97895
One-Class SVM	0.9824561404	0.97895

Table 11: MLP performance (Standardization, improved parameters).

35	0
0	22

(a) Isolation Forest

35	0
1	21

(b) Min. Covariance Det.

35	0
1	21

(c) Local Outlier Factor

35	0
1	21

(d) One-Class SVM

Table 12: Confusion matrices MLP performance (Standardization, improved parameters)