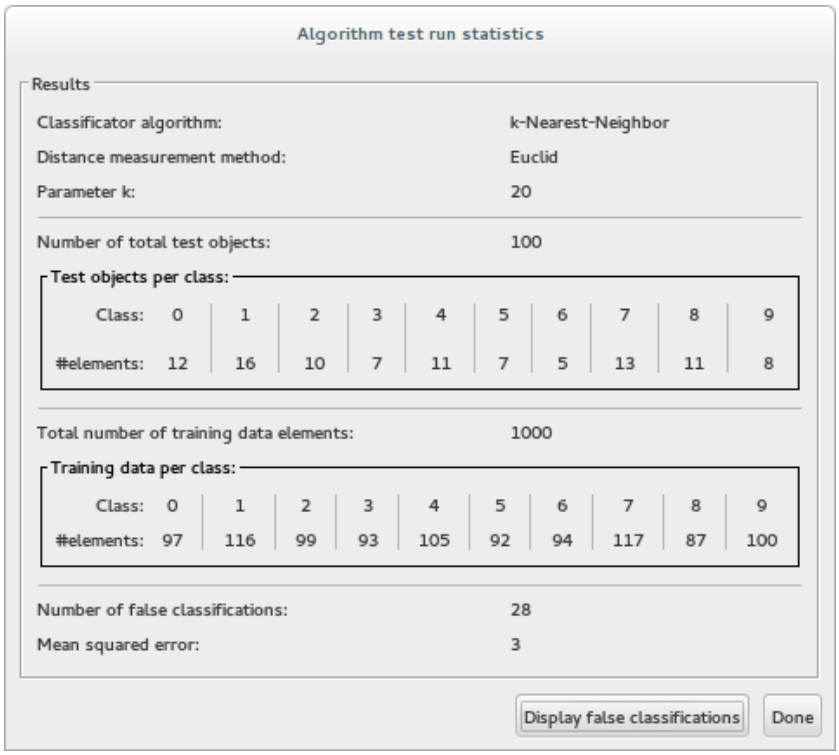
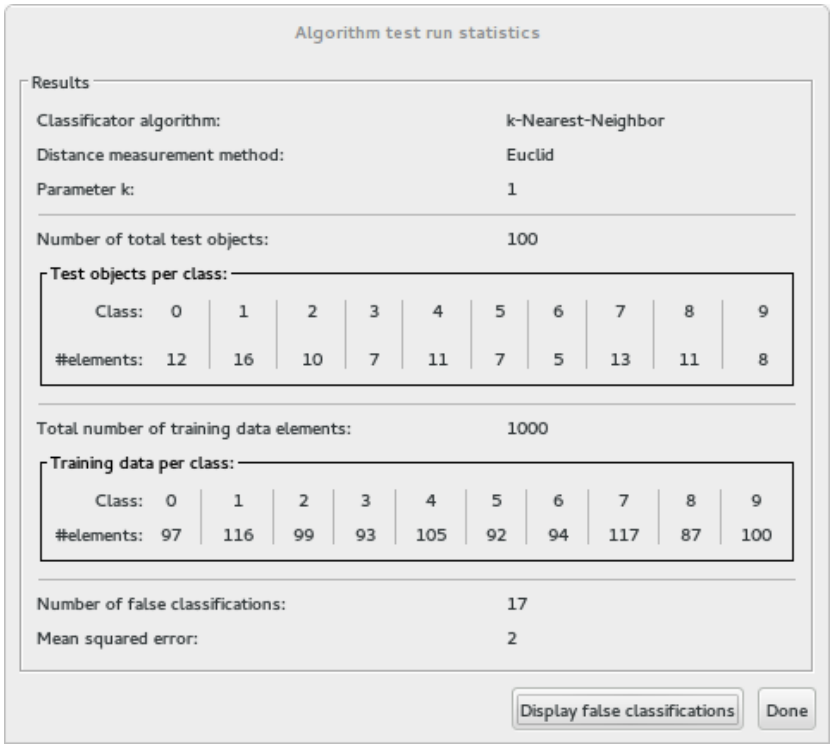
Statistik

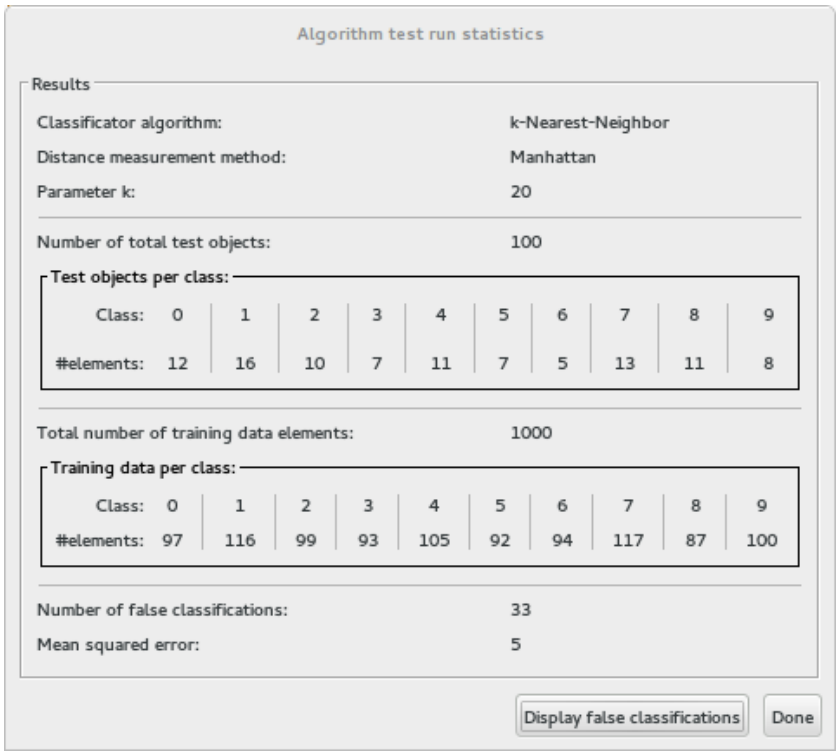
In den folgenden 6 Durchläufen wurden die ersten 1000 Elemente aus der MNIST Datenbank als Trainingsdaten verwendet. Die darauf folgenden 100 Elemente waren zu klassifizieren.

In den ersten vier Durchläufen haben wir den KNN Algorithmus mit unterschiedlichem k und den beiden Distanzfunktionen getestet.
Es ist festzustellen, dass KNN mit einem geringerem k besser funktioniert. Es wurden nicht nur mehr Elemente besser erkannt, es ist auch die mittlere quadratische Abweichung zur eigentlichen Zahl geringer. Es ist zu vermuten, dass ein zu hohes k dazu führt, dass bei der Klassifizierung einer Zahl mehr Datenpunkte zur Betrachtung gezogen werden, die zu den Punkthäufungen anderer Zahlen gehören. Dadurch kommen mehr falsche Klassifizierungen zu Stande.

Auf den folgenden Seiten sind Screenshots direkt aus dem Programm zu sehen.


Algorithm test run statistics
Results
Classificator algorithm: k-Nearest-Neighbor
Distance measurement method: Euclid
Parameter k: 20
Number of total test objects: 100
Test objects per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 12 16 10 7 11 7 5 13 11 8
Total number of training data elements: 1000
Training data per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 97 116 99 93 105 92 94 117 87 100
Number of false classifications: 28
Mean squared error: 3
Display false classifications
Done


Algorithm test run statistics
Results
Classificator algorithm: k-Nearest-Neighbor
Distance measurement method: Euclid
Parameter k: 1
Number of total test objects: 100
Test objects per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 12 16 10 7 11 7 5 13 11 8
Total number of training data elements: 1000
Training data per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 97 116 99 93 105 92 94 117 87 100
Number of false classifications: 17
Mean squared error: 2
Display false classifications
Done


Algorithm test run statistics
Results
Classificator algorithm: k-Nearest-Neighbor
Distance measurement method: Manhattan
Parameter k: 20
Number of total test objects: 100
Test objects per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 12 16 10 7 11 7 5 13 11 8
Total number of training data elements: 1000
Training data per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 97 116 99 93 105 92 94 117 87 100
Number of false classifications: 33
Mean squared error: 5
Display false classifications
Done


Algorithm test run statistics
Results
Classificator algorithm: k-Nearest-Neighbor
Distance measurement method: Manhattan
Parameter k: 1
Number of total test objects: 100
Test objects per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 12 16 10 7 11 7 5 13 11 8
Total number of training data elements: 1000
Training data per class:
Class: 0 1 2 3 4 5 6 7 8 9
#elements: 97 116 99 93 105 92 94 117 87 100
Number of false classifications: 18
Mean squared error: 2
Display false classifications
Done

Der K-Mean Algorithmus basiert natürlich auf der Klassifizierung der Cluster durch einen Menschen, der Zahlen in den Schwerpunkten eines Cluster zu erkennen hat.
Es würde sich nicht erweisen hier ein geringeres k als 20 zu wählen, da es schnell passiert, dass eine Zahl als Cluster nicht mit abgedeckt wird.
Die beiden Distanzfunktionen verhalten sich ähnlich, wie auch der mittlere quadratische Abstand zur eigentlichen Zahl.
Als Abbruchkriterium musste der Wert 1 für die mittlere Schwerpunktverschiebung bei der Neuberechnung der Schwerpunkte unterschritten werden. Ein weiteres Abbruchkriterium war die maximale Anzahl von 20 Iterationsschritten.