AI course - Lab 4

Fredrik Mårtensson

December 2019

1 Part 1

The training and test set during part one have been divided as: Training: 1600 Testing: 400

1.1 Task A

KNN is a simple algorithm for finding neighbouring neighbours for classification. By measuring the distance between a training and test set it is possible to determine the neighbouring neighbours. The method is not very effective since with every "request" for classification, the whole model must be repeated. The classification model ended up with a accuracy of 83%, this number might differ since we randomly pick out a test set of 20% on the total set.

1.2 Task B

Model	Cross validation
Nearest Neighbors	72.71%
SVC	93.01%
Decision Tree	82.27%
Random Forest	87.22%

Table 1: Performance of each algorithm

1.3 Task C

The parameters that were changes was the k value that specifies the number of neighbours that should be used to classify a prediction. In these tests k value went between 1-15. The metrics used is euclidean, manhattan and chebyshev.

By running all combinations the top five of the predictions were as following:

Metric	k	Cross validation
manhattan	5	79.0%
manhattan	3	78.25%
manhattan	7	78.0%
euclidean	3	77.75%
manhattan	9	77.75%

Table 2: Performance of each algorithm

It is noticeable that manhattan had the best score which we can see on the cross validation.

1.4 Task D (Bonus)

Both models in Task B and Task C performed well, however by utilising SVC it was possible to reach 93.01% compared to the KNN that reached 79% at most. This means that comparing and using different algorithms could prove better use than just changing the metrics. However The metrics did matter but as it could improve the result it could also make it worse.

The point that were discovered is the following:

$$[[0.23261, 1, 20.051, 0.38085, 85.232, 5.5067, 1742.6, 35.594, 31.772]]$$

The expected prediction is 1. In the table below two of four algorithms guessed the correct answer.

Model	Prediction
Nearest Neighbors	1.
SVC	1.
Decision Tree	2.
manhattan	2.

Table 3: Performance of each algorithm

1.5 Task E

The regression model is quite similar to classification, in code at least. The meaning of regression is to set a number instead of a class. This however is not possible for all cases depending on what type of data that is used. Instead of counting the most common when doing the model we take the sum divided by the length of the data. As expected we get an accuracy of 0% using this model. This is caused by minimal changes. One example is the following: guess 32.565, correct 32.133. There isn't a huge change in number but enough to make the answer and prediction incorrect. Therefor the algorithm land on a low accuracy.

1.6 Task F

As expected from the previous results this data is not well adapted for regression.

Model	Cross validation
Nearest Neighbors	28.24%
Linear Regression	73.15%
Random Forest	56.46%

Table 4: Performance of each algorithm

1.7 Task G

The parameters that were changes was the k value that specifies the number of neighbours that should be used to classify a prediction. In these tests k value went between 1-15. The metrics used is euclidean, manhattan and chebyshev.

Metric	k	Cross validation
manhattan	13	35.92%
manhattan	9	35.78%
manhattan	10	35.73%
manhattan	12	35.56%
manhattan	14	35.23%

Table 5: Performance of each algorithm

1.8 Task H (Bonus)

As given before in Task D it is possible to see that Linear Regression in Task F performed best and that the different metrics almost had no effect on the predictions. As expected before the metrics does not help the results while changing method of algorithm improves the result. The same point from Task D were used:

$$[[0.23261, 1, 20.051, 0.38085, 85.232, 5.5067, 1742.6, 35.594, 31.772]]$$

The expected value is however changed to: 35.594. The result in the table below shows that none of the models were able to guess the right answer but were quite close. Even thought the didn't guess the correct answer it could be considered an good result since the model would most likely overfit if it predicted the exact answer.

Model	Prediction
Nearest Neighbors	35.1844
Linear	35.69753719
Random Forest	34.3974268

Table 6: Performance of each algorithm

2 Part 2

The training and test set during part one have been divided as: Training: 160 Testing: 40

2.1 Task A

The additional parameters added is Average raised and Raise Ratio.

2.2 Task B

Model	Cross validation
Nearest Neighbors	78.5%
SVC	87.83%
Random Forest	89.57%

Table 7: Performance of Agent 1

Model	Cross validation
Nearest Neighbors	86.33%
SVC	65.55%
Random Forest	96.07%

Table 8: Performance of Agent 2

2.3 Task C

The k value were changed between 1-15 and the metrics used was: euclidean, manhattan and chebyshev. In the table below we can see that the highest accuracy was on predicting Agent 1 actions.

Agent	k	Metric	Cross validation
p1	1	manhattan	87.17%
p1	2	manhattan	87.17%
p1	8	manhattan	87.17%
p1	9	manhattan	84.67%
p1	7	manhattan	83.83%

Table 9: Performance of Agents