# Homework 2

Your Name: \_\_\_\_\_\_\_\_\_\_\_\_TsuCheng Lu\_\_\_\_\_\_\_\_\_\_\_

Student ID:\_\_\_\_\_\_\_A20525040\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. (50 points) Classification: We are going to make a decision about whether an animal is useful (P) or useless (N) in our experiments. We measure their age in days, whether fat or not, and the size of their soles of the feet. [Manually solve this problem without Python]**

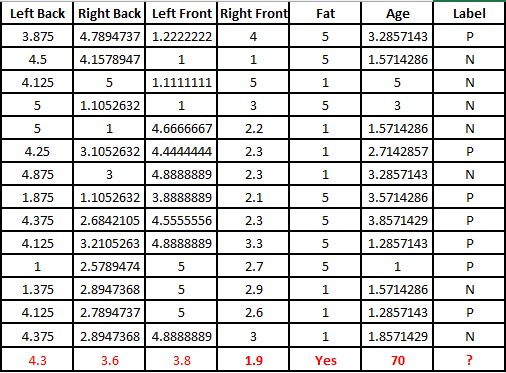
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Left Back** | **Right Back** | **Left Front** | **Right Front** | **Fat** | **Age** | **Label** |
| 4.1 | 4.8 | 1.6 | 3.2 | Yes | 100 | P |
| 4.6 | 4.2 | 1.4 | 0.2 | Yes | 40 | N |
| 4.3 | 5.0 | 1.5 | 4.2 | No | 160 | N |
| 5 | 1.3 | 1.4 | 2.2 | Yes | 90 | N |
| 5 | 1.2 | 4.7 | 1.4 | No | 40 | N |
| 4.4 | 3.2 | 4.5 | 1.5 | No | 80 | P |
| 4.9 | 3.1 | 4.9 | 1.5 | No | 100 | N |
| 2.5 | 1.3 | 4 | 1.3 | Yes | 110 | P |
| 4.5 | 2.8 | 4.6 | 1.5 | Yes | 120 | P |
| 4.3 | 3.3 | 4.9 | 2.5 | Yes | 30 | P |
| 1.8 | 2.7 | 5.0 | 1.9 | Yes | 20 | P |
| 2.1 | 3 | 5.0 | 2.1 | No | 40 | N |
| 4.3 | 2.9 | 5.0 | 1.8 | No | 30 | P |
| 4.5 | 3 | 4.9 | 2.2 | No | 50 | N |
| **4.3** | **3.6** | **3.8** | **1.9** | **Yes** | **70** | **?** |

a). [10 points] Do we need normalization and discretization (data type transformation) to use KNN classifier? Why (use your own text/description)?

Yes, normalization and discretization can be valuable tools for processing data when using K-Nearest Neighbors (KNN). Because all of the data value are too big and some of values are not even integer. For this reason, we have to change Fat value, yes change to 5, and no change to 1. Also, the soles of the feet and age will calculate to the max and min normalization.

b). [10 points] If your answer is Yes in part 1), please apply normalization (to new scale [1,5]) and discretization. Give the process of preprocessing and the table of final data

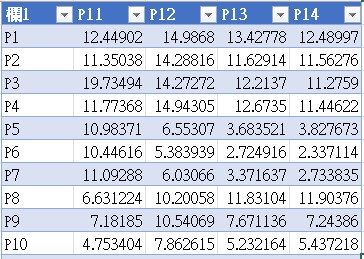
First of all, changing the fat column (Yes:5 and No:1) and calculating the max min normalization of the left back, right back, left front, right front, and age.



c). [30 points] Use first 10 rows as training, the next 4 rows as testing set. Apply KNN Classifier to the new data table in part b). In other words, build your KNN classifier by the following requirements based on the knowledge in the table, and then predict which class/label the object (in red) belongs to:

After preprocessing, we calculate the distance by using Manhattan distance

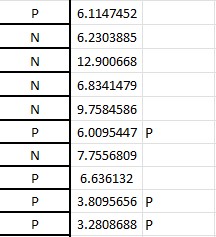
* Distance measures: Manhattan distance



Then, we can know the K values is P or N, and find the best model is K=3. After that we can start to calculate the distance of the unseen data. Finally, we can find the lowest 3 values label are all P, so the unseen label will be the P.

* K = 1, 3, 5

Find the best K value by examining accuracy on the test set. Finally apply the best setting and predict the label for the unseen data.





**2. (50 points) Python practice for KNN classification**

**Use the following data, and run KNN to find the best parameters and performance**

* Use Malware detection data set (malware\_Binary.csv) to run KNN by using 75% as training, 25% as testing
* Read the description and ReadMe document in the malware detection data

Note:

* You need to change different/multiple parameters to find the best KNN model.
* Use accuracy, precision, recall, AUC as metrics
* You can find data sets from “slide & data” on blackboard system

Submission

* The ipynb and saved html files
* A comparison of different parameters and AUC values

As a result, K = 25 will be the best K value because the accuracy is most which is closer to the 1.

