# Homework 2

Your Name: \_Vikas Sanil\_\_\_\_\_\_\_\_\_\_\_

Student ID:\_\_A20511552\_\_\_\_\_\_\_\_\_\_\_\_\_

**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 | P |
| 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 | N |
| 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** | **1.5** | **1.8** | **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, we need normalization of “Age” features since they have higher value differences and will dominate the KNN classifier result if not normalized. Discretization is required for the nominal feature “Fat” since we need to find distance between the features.

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

Used Min-Max normalization with scale[1,5] for ‘Age’ feature and performed discretization of feature ‘Fat’ to “No” as 1 and “Yes” as 5 since we are using scale[1,5], which would work well with the rest of the data.

| **Left Back** | **Right Back** | **Left Front** | **Right Front** | **Fat** | **norm(Age)** | **Label** |
| --- | --- | --- | --- | --- | --- | --- |
| 4.1 | 4.8 | 1.6 | 3.2 | 5 | 3.3 | P |
| 4.6 | 4.2 | 1.4 | 0.2 | 5 | 1.6 | N |
| 4.3 | 5 | 1.5 | 4.2 | 1 | 5 | N |
| 5 | 1.3 | 1.4 | 2.2 | 5 | 3 | P |
| 5 | 1.2 | 4.7 | 1.4 | 1 | 1.6 | N |
| 4.4 | 3.2 | 4.5 | 1.5 | 1 | 2.7 | P |
| 4.9 | 3.1 | 4.9 | 1.5 | 1 | 3.3 | N |
| 2.5 | 1.3 | 4 | 1.3 | 5 | 3.6 | P |
| 4.5 | 2.8 | 4.6 | 1.5 | 5 | 3.9 | P |
| 4.3 | 3.3 | 4.9 | 2.5 | 5 | 1.3 | N |
| 1.8 | 2.7 | 5 | 1.9 | 5 | 1 | P |
| 2.1 | 3 | 5 | 2.1 | 1 | 1.6 | N |
| 4.3 | 2.9 | 5 | 1.8 | 1 | 1.3 | P |
| 4.5 | 3 | 4.9 | 2.2 | 1 | 1.9 | N |
| **4.3** | **3.6** | **1.5** | **1.8** | **5** | 2.4 | **?** |

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:

* Distance measures: Manhattan distance

|  | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | p10 | p11 | p12 | p13 | p14 | p15 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| p2 | 6 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| p3 | 7.2 | 12.6 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |
| p4 | 5.9 | 6.7 | 12.5 | 0 |  |  |  |  |  |  |  |  |  |  |  |
| p5 | 15.1 | 11.9 | 13.9 | 9.6 | 0 |  |  |  |  |  |  |  |  |  |  |
| p6 | 11.1 | 10.7 | 9.9 | 10.6 | 4 | 0 |  |  |  |  |  |  |  |  |  |
| p7 | 11.5 | 11.9 | 10.3 | 10.4 | 4 | 1.6 | 0 |  |  |  |  |  |  |  |  |
| p8 | 9.7 | 10.7 | 16.3 | 6.6 | 9.4 | 9.4 | 9.6 | 0 |  |  |  |  |  |  |  |
| p9 | 7.7 | 8.3 | 13.3 | 6.8 | 8.6 | 5.8 | 5.6 | 4.6 | 0 |  |  |  |  |  |  |
| p10 | 7.7 | 7.3 | 14.5 | 8.2 | 8.4 | 7 | 7.8 | 8.2 | 4.6 | 0 |  |  |  |  |  |
| p11 | 11.4 | 10.2 | 18.6 | 10.5 | 10.1 | 9.7 | 10.3 | 6.3 | 6.5 | 4.1 | 0 |  |  |  |  |
| p12 | 14 | 13.2 | 13.2 | 13.7 | 5.7 | 4.7 | 5.3 | 9.9 | 9.9 | 7.3 | 5.4 | 0 |  |  |  |
| p13 | 12.9 | 11.1 | 11.7 | 12 | 3.4 | 2.6 | 3.2 | 11.2 | 7.6 | 5.2 | 7.1 | 2.9 | 0 |  |  |
| p14 | 11.9 | 11.1 | 10.7 | 10.8 | 3.6 | 2.2 | 2.6 | 11.2 | 7.2 | 5.4 | 8.3 | 2.9 | 1.4 | 0 |  |
| p15 | 3.8 | 3.4 | 10.4 | 4.1 | 11.5 | 8.1 | 9.7 | 8.3 | 5.9 | 5.5 | 9.3 | 11.4 | 9.3 | 9.1 | 0 |

* K = 1, 3, 5

When K = 1, then p15` KNN =(p2, p1, p4, p10, p9, p6, p8, p14,p11, p13, p7, p3, p12, p5.)

When K = 3, then p15` KNN =(p2, p1, p4, p10, p9, p6, p8, p14, p11, p13, p7, p3, p12, p5.) and

When K = 5, then p15` KNN =(p10, p4, p9, p1, p2, p6, p8, p14, p11, p13, p7, p3, p12, p5)

Find the best K value by examining F1 score on the test set. Finally apply the best setting and predict the label for the unseen data.

From the above sets the nearest matching parameters are p4 and p1. Hence the best suited label for p15 is “P”

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

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

* Use Loans\_20K.csv data by using 10-fold cross validation
* Use Loans\_200K.csv data by using 75% as training, 25% as testing

Note:

* You need to change different/multiple parameters to find the best KNN model.
* You can find data sets from “slide & data” on blackboard system

Submission

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