**Logo

Description automatically generated**

**San Francisco Bay University**

**CS483 - Fundamentals of Artificial Intelligence**

**Homework Assignment #3**

**Due day: 7/9/2022**

**Instruction:**

1. **Push the source code to Github**

**<https://github.com/SharonCao0920/AI/blob/main/CS483_Lecture_HW_3.ipynb>**

1. **Overdue homework submission could not be accepted.**
2. **Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**
3. Confusion matrix is the table to present the performance of an algorithm for the classification. Assuming that the example of 3 by 3 confusion matrix comes from the outputs of 3 clusters classification as follows, please find the total accuracy, and each cluster’s precision, recall and F1-score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Predicted Values | | |
| Cat | Dog | Bird |
| Actual Values | Cat | 20 | 1 | 1 |
| Dog | 1 | 19 | 2 |
| Bird | 2 | 0 | 28 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  |  |  | |  |  |
|  |  | | | Predicted Values | | | |  |  |
|  |  | | | Cat | Dog | Bird | |  |  |
|  | Actual Values | Cat | | 20 | 1 | 1 | | 22 |  |
|  |  | Dog | | 1 | 19 | 2 | | 22 |  |
|  |  | Bird | | 2 | 0 | 28 | | 30 |  |
|  |  |  | | 23 | 20 | 31 | | 74 |  |
|  |  |  | |  |  |  | |  |  |
|  |  | |  | | | |  | | |
|  | **Accuracy: Total Correct /Actual** | | | | | | | | |
|  | ( 20 + 19 + 28) / 74 = | | 0.905405405 | | | |  | | |
|  |  | |  | | | |  | | |
|  | **Precision = Current Predicted / Total Predicted** | | | | | | | | |
|  | Cat: | | 20 /23 = | | | | 0.869565217 | | |
|  | Dog: | | 19 / 20 = | | | | 0.95 | | |
|  | Bird: | | 28 /31 = | | | | 0.903225806 | | |
|  |  | |  | | | |  | | |
|  | **Recall = Correct / Predicted** | | | | | | | | |
|  | Cat: | | 20 /22 = | | | | 0.909090909 | | |
|  | Dog: | | 19 / 22 = | | | | 0.863636364 | | |
|  | Bird: | | 28 /30 = | | | | 0.933333333 | | |
|  |  | |  | | | |  | | |
|  | **F1-score = ( 2\*Precision\*Recall ) / ( Precision + Recall )** | | | | | | | | |
|  | Cat: | | (2\*0.87\*0.91)/(0.87+0.91)= | | | | 0.888888889 | | |
|  | Dog: | | (2\*0.95\*0.87)/(0.95+0.87)= | | | | 0.904761905 | | |
|  | Bird: | | (2\*0.90\*0.93)/(0.90+0.93)= | | | | 0.918032787 | | |
|  |  | |  | | | |  | | |

1. Design KNN classifier based on the following small dataset

- Preprocess the dataset first before any processing by substituting *M* & *F* with *0* and *1* respectively

- Randomly separate the dataset to training set (*70%*) and validation set (*30%*) by sample’s **ID** generated either from Python program or Excel unduplicated random function

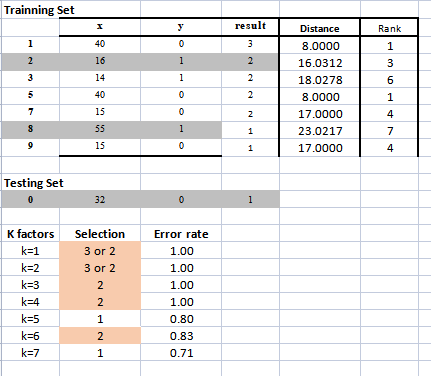
- Calculate error rate for validation set from *K=1* to *K=7* either in Python program or Excel

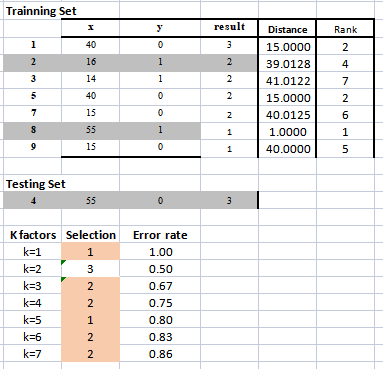
- Select an appropriate *K*’s value and predict what class the new data in **red** color belongs to

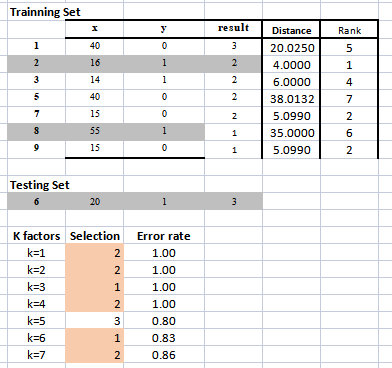
- Finally write Python program by calling functions from ***scikit-learn*** to verify your design based on hand calculation results

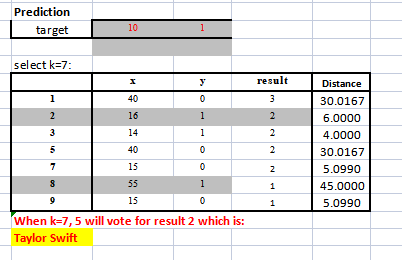
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Age** | **Gender** | **Fan** |
| **0** | Bill | 32 | M | Rolling Stones |
| **1** | Henry | 40 | M | Neither |
| **2** | Mary | 16 | F | Taylor Swift |
| **3** | Tiffany | 14 | F | Taylor Swift |
| **4** | Michael | 55 | M | Neither |
| **5** | Carlos | 40 | M | Taylor Swift |
| **6** | Ashely | 20 | F | Neither |
| **7** | Robert | 15 | M | Taylor Swift |
| **8** | Sally | 55 | F | Rolling Stones |
| **9** | John | 15 | M | Rolling Stones |
| **10** | Michelle | 10 | F | ? |











1. K-Means algorithm is one of popular methods in unsupervised learning. Please plot elbow curve of total WCSS (within cluster sum of square) vs *K* from *1* to *5* either created by hand or Python program and select a proper *K*’s value based on your observation as the final number of clusters in your design. And then write Python program to verify your by-hand calculation results

|  |  |  |  |
| --- | --- | --- | --- |
| **Objects** | **X** | **Y** | **Z** |
| **OB-1** | 1 | 4 | 1 |
| **OB-2** | 1 | 2 | 2 |
| **OB-3** | 1 | 4 | 2 |
| **OB-4** | 2 | 1 | 2 |
| **OB-5** | 1 | 1 | 1 |
| **OB-6** | 2 | 4 | 2 |
| **OB-7** | 1 | 1 | 2 |
| **OB-8** | 2 | 1 | 1 |