**Confusion Matrix For binary classification**

A 2X2 Confusion matrix is shown below for the image recognition having a Dog image or Not Dog image.

|  |  | **Actual** |  |
|---|---|---|---|
|  |  | **Dog** | **Not Dog** |
| **Predicted** | **Dog** | True Positive (TP) | False Positive (FP) |
|  | **Not Dog** | False Negative (FN) | True Negative (TN) |

- **True Positive (TP):** It is the total counts having both predicted and actual values are Dog.
- **True Negative (TN):** It is the total counts having both predicted and actual values are Not Dog.
- **False Positive (FP):** It is the total counts having prediction is Dog while actually Not Dog.
- **False Negative (FN):** It is the total counts having prediction is Not Dog while actually, it is Dog.

**Example for binary classification problems**

| **Index** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Actual** | Dog | Dog | Dog | Not Dog | Dog | Not Dog | Dog | Dog | Not Dog | Not Dog |
| **Predicted** | Dog | Not Dog | Dog | Not Dog | Dog | Dog | Dog | Dog | Not Dog | Not Dog |
| **Result** | TP | FN | TP | TN | TP | FP | TP | TP | TN | TN |

- Actual Dog Counts = 6
- Actual Not Dog Counts = 4
- True Positive Counts = 5
- False Positive Counts = 1
- True Negative Counts = 3
- False Negative Counts = 1

|  | **Actual** |
|---|---|

| | | Dog | Not Dog |
|---|---|---|---|
| **Predicted** | **Dog** | True Positive<br>(TP =5) | False Positive<br>(FP=1) |
| | **Not Dog** | False Negative<br>(FN =1) | True Negative<br>(TN=3) |

**Metrics based on Confusion Matrix Data**

**1. Accuracy**

Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances.

*Accuracy=TP+TN+FP+FNTP+TN*

For the above case:

Accuracy = (5+3)/(5+3+1+1) = 8/10 = 0.8

**2. Precision**

Precision is a measure of how accurate a model's positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.

Precision=*TP+FPTP*

For the above case:

Precision = 5/(5+1) =5/6 = 0.8333

**3. Recall**

Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

Recall=*TP+FNTP*

For the above case:

Recall = 5/(5+1) =5/6 = 0.8333

**4. F1-Score**

F1-score is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall,

F1-Score=2·

F1-Score=*Precision+Recall*2·*Precision·Recall*

For the above case:

F1-Score: = (2* 0.8333* 0.8333)/( 0.8333+ 0.8333) = 0.8333

**5. Specificity:**

Specificity is another important metric in the evaluation of classification models, particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the True Negative Rate.

Specificity=$TN+FPTN$
Specificity=3/(1+3)=3/4=0.75

### 6. Type 1 and Type 2 error

### Type 1 error

Type 1 error occurs when the model predicts a positive instance, but it is actually negative. Precision is affected by false positives, as it is the ratio of true positives to the sum of true positives and false positives.

Type 1 Error=$TN+FPFP$

### Type 2 error

Type 2 error occurs when the model fails to predict a positive instance. Recall is directly affected by false negatives, as it is the ratio of true positives to the sum of true positives and false negatives.

In the context of medical testing, a Type 2 Error, often known as a false negative, occurs when a diagnostic test fails to detect the presence of a disease in a patient who genuinely has it. The consequences of such an error are significant, as it may result in a delayed diagnosis and subsequent treatment.

Type 2 Error=$TP+FNFN$

Precision emphasizes minimizing false positives, while recall focuses on minimizing false negatives.

**Confusion Matrix For Multi-class Classification**

Let’s consider there are three classes. A 3X3 Confusion matrix is shown below for the image having three classes.

Here, TP= True Positive , FP= False Positive , FN= False Negative.

| **Index** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Actual** | Cat | Dog | Horse | Cat | Dog | Cat | Dog | Horse | Horse | Cat |
| **Precticted** | Cat | Dog | Dog | Cat | Dog | Cat | Dog | Horse | Horse | Dog |
| **Result** | TP | TP | FN | TP | TP | TP | TP | TP | TP | FN |

A 3X3 Confusion matrix is shown below for three classes.

|  |  | **Actual** |  |  |
|---|---|---|---|---|
|  |  | **Cat** | **Dog** | **Horse** |
| **Predicted** | Cat | TP | FP | FP |
|  | Dog | **FN** | **TP** | **FP** |

|  |  |  |  |  |
|---|---|---|---|---|
|  | Horse | FN | FN | TP |

- Class-wise Summary:

**For Cat:**

- **True Positives (TP):** 3
    - Index 1: True Positive (Cat actual, Cat predicted)
    - Index 4: True Positive (Cat actual, Cat predicted)
    - Index 6: True Positive (Cat actual, Cat predicted)
- **False Negatives (FN):** 1
    - Index 10: False Negative (Cat actual, Dog predicted)

**For Dog:**

- **True Positives (TP):** 5
    - Index 2: True Positive (Dog actual, Dog predicted)
    - Index 5: True Positive (Dog actual, Dog predicted)
    - Index 7: True Positive (Dog actual, Dog predicted)
    - Index 10: True Positive (Cat actual, Dog predicted)
    - Index 3: False Negative (Horse actual, Dog predicted)

**For Horse:**

- **True Positives (TP):** 3
    - Index 8: True Positive (Horse actual, Horse predicted)
    - Index 9: True Positive (Horse actual, Horse predicted)
    - Index 3: False Negative (Horse actual, Dog predicted)

Then, the confusion matrix will be:

|  |  | **Actual** |  |  |
|---|---|---|---|---|
|  |  | **Cat** | **Dog** | **Horse** |
| **Predicted** | Cat | TP(3) | FP(1) | FP(0) |
|  | Dog | **FN(0)** | **TP(5)** | **FP(1)** |
|  | Horse | FN(1) | FN(1) | TP(3) |

### K-means Clustering Numerical Example with Solution

You are given 15 points in the Cartesian coordinate system as follows.

| Point | Coordinates |
|---|---|

| | |
|---|---|
| A1 | (2,10) |
| A2 | (2,6) |
| A3 | (11,11) |
| A4 | (6,9) |
| A5 | (6,4) |
| A6 | (1,2) |
| A7 | (5,10) |
| A8 | (4,9) |
| A9 | (10,12) |
| A10 | (7,5) |
| A11 | (9,11) |
| A12 | (4,6) |
| A13 | (3,10) |
| A14 | (3,8) |
| A15 | (6,11) |

We are also given the information that we need to make 3 clusters.

It means we are given K=3.We will solve this numerical on k-means clustering using the approach discussed below.

First, we will randomly choose 3 centroids from the given data. Let us consider A2 (2,6), A7 (5,10), and A15 (6,11) as the centroids of the initial clusters. Hence, we will consider that

- Centroid 1=(2,6) is associated with cluster 1.
- Centroid 2=(5,10) is associated with cluster 2.
- Centroid 3=(6,11) is associated with cluster 3.

Now we will find the euclidean distance between each point and the centroids. Based on the minimum distance of each point from the centroids, we will assign the points to a cluster. I have tabulated the distance of the given points from the clusters in the following table

| Point | Distance from Centroid 1 (2,6) | Distance from Centroid 2 (5,10) | Distance from Centroid 3 (6,11) | Assigned Cluster |
|---|---|---|---|---|
| A1 | 4 | 3 | 4.123106 | Cluster 2 |

| | | | | |
|---|---|---|---|---|
| (2,10) | | | | |
| A2<br>(2,6) | 0 | 5 | 6.403124 | Cluster 1 |
| A3<br>(11,11) | 10.29563 | 6.082763 | 5 | Cluster 3 |
| A4<br>(6,9) | 5 | 1.414214 | 2 | Cluster 2 |
| A5<br>(6,4) | 4.472136 | 6.082763 | 7 | Cluster 1 |
| A6<br>(1,2) | 4.123106 | 8.944272 | 10.29563 | Cluster 1 |
| A7<br>(5,10) | 5 | 0 | 1.414214 | Cluster 2 |
| A8<br>(4,9) | 3.605551 | 1.414214 | 2.828427 | Cluster 2 |
| A9<br>(10,12) | 10 | 5.385165 | 4.123106 | Cluster 3 |
| A10<br>(7,5) | 5.09902 | 5.385165 | 6.082763 | Cluster 1 |
| A11<br>(9,11) | 8.602325 | 4.123106 | 3 | Cluster 3 |
| A12<br>(4,6) | 2 | 4.123106 | 5.385165 | Cluster 1 |
| A13<br>(3,10) | 4.123106 | 2 | 3.162278 | Cluster 2 |
| A14<br>(3,8) | 2.236068 | 2.828427 | 4.242641 | Cluster 1 |
| A15<br>(6,11) | 6.403124 | 1.414214 | 0 | Cluster 3 |

Results from 1st iteration of K means clustering
At this point, we have completed the first iteration of the k-means clustering algorithm and assigned each point into a cluster.

In the above table, you can observe that the point that is closest to the centroid of a given cluster is assigned to the cluster.

we will calculate the new centroid for each cluster.

- In cluster 1, we have 6 points i.e. A2 (2,6), A5 (6,4), A6 (1,2), A10 (7,5), A12 (4,6), A14 (3,8). To calculate the new centroid for cluster 1, we will find the mean of the x and y coordinates of each point in the cluster. Hence, the new centroid for cluster 1 is (3.833, 5.167).
- In cluster 2, we have 5 points i.e. A1 (2,10), A4 (6,9), A7 (5,10) , A8 (4,9), and A13 (3,10). Hence, the new centroid for cluster 2 is (4, 9.6)
- In cluster 3, we have 4 points i.e. A3 (11,11), A9 (10,12), A11 (9,11), and A15 (6,11). Hence, the new centroid for cluster 3 is (9, 11.25).

Now that we have calculated new centroids for each cluster, we will calculate the distance of each data point from the new centroids. Then, we will assign the points to clusters based on their distance from the centroids. The results for this process have been given in the following table.

| Point | Distance from Centroid 1 (3.833, 5.167) | Distance from centroid 2 (4, 9.6) | Distance from centroid 3 (9, 11.25) | Assigned Cluster |
|---|---|---|---|---|
| A1 (2,10) | 5.169 | 2.040 | 7.111 | Cluster 2 |
| A2 (2,6) | 2.013 | 4.118 | 8.750 | Cluster 1 |
| A3 (11,11) | 9.241 | 7.139 | 2.016 | Cluster 3 |
| A4 (6,9) | 4.403 | 2.088 | 3.750 | Cluster 2 |
| A5 (6,4) | 2.461 | 5.946 | 7.846 | Cluster 1 |
| A6 (1,2) | 4.249 | 8.171 | 12.230 | Cluster 1 |
| A7 (5,10) | 4.972 | 1.077 | 4.191 | Cluster 2 |
| A8 (4,9) | 3.837 | 0.600 | 5.483 | Cluster 2 |
| A9 (10,12) | 9.204 | 6.462 | 1.250 | Cluster 3 |
| A10 (7,5) | 3.171 | 5.492 | 6.562 | Cluster 1 |
| A11 (9,11) | 7.792 | 5.192 | 0.250 | Cluster 3 |
| A12 | 0.850 | 3.600 | 7.250 | Cluster 1 |

| | | | | |
|---|---|---|---|---|
| (4,6) | | | | |
| A13 (3,10) | 4.904 | 1.077 | 6.129 | Cluster 2 |
| A14 (3,8) | 2.953 | 1.887 | 6.824 | Cluster 2 |
| A15 (6,11) | 6.223 | 2.441 | 3.010 | Cluster 2 |

Results from 2nd iteration of K means clustering

Now, we have completed the second iteration of the k-means clustering algorithm and assigned each point into an updated cluster. In the above table, you can observe that the point closest to the new centroid of a given cluster is assigned to the cluster.

we will calculate the new centroid for each cluster for the third iteration.

- In cluster 1, we have 5 points i.e. A2 (2,6), A5 (6,4), A6 (1,2), A10 (7,5), and A12 (4,6). To calculate the new centroid for cluster 1, we will find the mean of the x and y coordinates of each point in the cluster. Hence, the new centroid for cluster 1 is (4, 4.6).
- In cluster 2, we have 7 points i.e. A1 (2,10), A4 (6,9), A7 (5,10) , A8 (4,9), A13 (3,10), A14 (3,8), and A15 (6,11). Hence, the new centroid for cluster 2 is (4.143, 9.571)
- In cluster 3, we have 3 points i.e. A3 (11,11), A9 (10,12), and A11 (9,11). Hence, the new centroid for cluster 3 is (10, 11.333).

we have calculated new centroids for each cluster. Now, we will calculate the distance of each data point from the new centroids. Then, we will assign the points to clusters based on their distance from the centroids. The results for this process have been given in the following table.

| Point | Distance from Centroid 1 (4, 4.6) | Distance from centroid 2 (4.143, 9.571) | Distance from centroid 3 (10, 11.333) | Assigned Cluster |
|---|---|---|---|---|
| A1 (2,10) | 5.758 | 2.186 | 8.110 | Cluster 2 |
| A2 (2,6) | 2.441 | 4.165 | 9.615 | Cluster 1 |
| A3 (11,11) | 9.485 | 7.004 | 1.054 | Cluster 3 |
| A4 (6,9) | 4.833 | 1.943 | 4.631 | Cluster 2 |
| A5 (6,4) | 2.088 | 5.872 | 8.353 | Cluster 1 |

| | | | | |
|---|---|---|---|---|
| A6 (1,2) | 3.970 | 8.197 | 12.966 | Cluster 1 |
| A7 (5,10) | 5.492 | 0.958 | 5.175 | Cluster 2 |
| A8 (4,9) | 4.400 | 0.589 | 6.438 | Cluster 2 |
| A9 (10,12) | 9.527 | 6.341 | 0.667 | Cluster 3 |
| A10 (7,5) | 3.027 | 5.390 | 7.008 | Cluster 1 |
| A11 (9,11) | 8.122 | 5.063 | 1.054 | Cluster 3 |
| A12 (4,6) | 1.400 | 3.574 | 8.028 | Cluster 1 |
| A13 (3,10) | 5.492 | 1.221 | 7.126 | Cluster 2 |
| A14 (3,8) | 3.544 | 1.943 | 7.753 | Cluster 2 |
| A15 (6,11) | 6.705 | 2.343 | 4.014 | Cluster 2 |

Results from 3rd iteration of K means clustering

Now, we have completed the third iteration of the k-means clustering algorithm and assigned each point into an updated cluster. In the above table, you can observe that the point that is closest to the new centroid of a given cluster is assigned to the cluster.

we will calculate the new centroid for each cluster for the third iteration.

- In cluster 1, we have 5 points i.e. A2 (2,6), A5 (6,4), A6 (1,2), A10 (7,5), and A12 (4,6). To calculate the new centroid for cluster 1, we will find the mean of the x and y coordinates of each point in the cluster. Hence, the new centroid for cluster 1 is (4, 4.6).
- In cluster 2, we have 7 points i.e. A1 (2,10), A4 (6,9), A7 (5,10) , A8 (4,9), A13 (3,10), A14 (3,8), and A15 (6,11). Hence, the new centroid for cluster 2 is (4.143, 9.571)
- In cluster 3, we have 3 points i.e. A3 (11,11), A9 (10,12), and A11 (9,11). Hence, the new centroid for cluster 3 is (10, 11.333).

no point has changed its cluster compared to the previous iteration. Due to this, the centroid also remains constant. Therefore, we will say that the clusters have been stabilized. Hence, the clusters obtained after the third iteration are the final clusters made from the given dataset.