**Practical No. 3: Manual Content**

| | | |
|---|---|---|
| | **Guru Gobind Singh Foundation**<br>**Guru Gobind Singh College of Engineering and Research Center, Nashik** | |

**Experiment No: 03**

**Title of Experiment:** Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.

Perform following steps:

1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points of improvement and implement the same.
5. Print the accuracy score and confusion matrix (5 points).

| | | | |
|---|---|---|---|
| **Student Name:** | | | |
| **Class:** | **BE (Computer)** | | |
| **Div:** | **A** | **Batch:** | **BECO** |
| **Roll No.:** | | | |
| **Date of Attendance (Performance):** | | | |
| **Date of Evaluation:** | | | |
| **Marks (Grade) Attainment of CO Marks out of 10** | | | |
| **CO Mapped** | CO6: Design a neural network for solving engineering problems. | | |
| **Signature of Subject Teacher** | | | |

| A | P | W | T | Total |
|---|---|---|---|---|
| | | | | |

**TITLE:** Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months.

**Dataset Description:** The case study is from an open-source dataset from Kaggle. The dataset contains 10,000 sample points with 14 distinct features such as CustomerId, CreditScore, Geography, Gender, Age, Tenure, Balance, etc.

Perform following steps:

1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points of improvement and implement the same.
5. Print the accuracy score and confusion matrix (5 points).

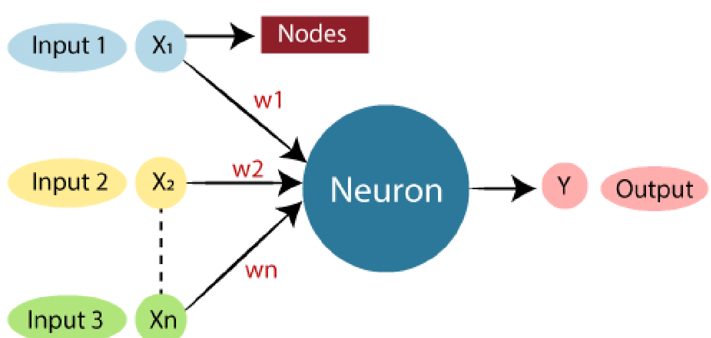
**AIM:** Aim of this practical is to design the neural networks classifier, distinguish the dataset features, create training-testing data and normalize it, estimate the accuracy and confusion matrix.

**OBJECTIVES:** Based on above main aim following are the objectives

1. To understand the concept of neural-network
2. To implement neural-network based classification models.

## Artificial Neural Network

The term "Artificial Neural Network" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.




Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

An Artificial Neural Network in the field of Artificial intelligence where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

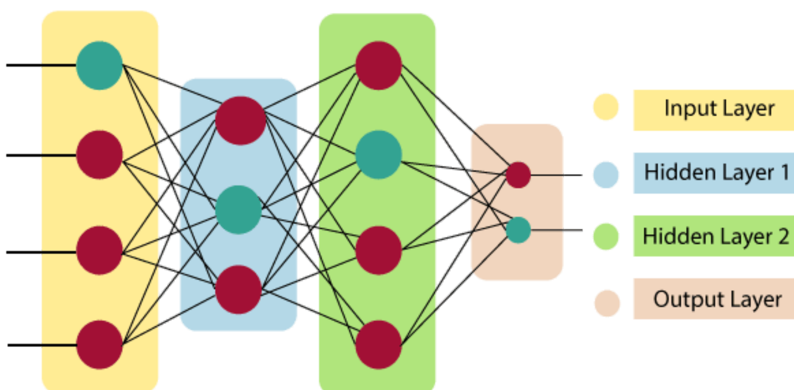
There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

**The architecture of an artificial neural network:**

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:




**Input Layer:**
As the name suggests, it accepts inputs in several different formats provided by the programmer.
**Hidden Layer:**
The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.
**Output Layer:**
The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b$$

It determines the weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

## What is Normalization in Machine Learning?

- Normalization is a scaling technique in Machine Learning applied during data preparation to change the values of numeric columns in the dataset to use a common scale. It is not necessary for all datasets in a model. It is required only when features of machine learning models have different ranges.
- Mathematically, we can calculate normalization with the below formula:

**Xn = (X - Xminimum) / ( Xmaximum - Xminimum)**

Xn = Value of Normalization

Xmaximum = Maximum value of a feature

Xminimum = Minimum value of a feature

## Normalization techniques in Machine Learning

Although there are so many feature normalization techniques in Machine Learning, few of them are most frequently used. These are as follows:

- **Min-Max Scaling:** This technique is also referred to as scaling. As we have already discussed above, the Min-Max scaling method helps the dataset to shift and rescale the values of their attributes, so they end up ranging between 0 and 1.
- **Standardization scaling:** Standardization scaling is also known as Z-score normalization, in which values are centered around the mean with a unit standard deviation, which means the attribute becomes zero and the resultant distribution has a unit standard deviation. Mathematically, we can calculate the standardization by subtracting the feature value from the mean and dividing it by standard deviation. Hence, standardization can be expressed as follows:

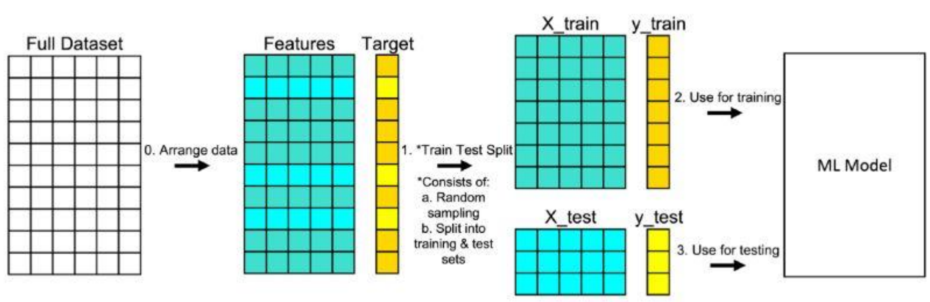
$$X' = \frac{X - \mu}{\sigma}$$

Here, μ represents the mean of feature value, and σ represents the standard deviation of feature values.

- However, unlike Min-Max scaling technique, feature values are not restricted to a specific range in the standardization technique.

## What Is the Train Test Split Procedure?

Train test split is a model validation procedure that allows you to simulate how a model would perform on new/unseen data. Here is how the procedure works:



**1. Arrange The Data**
Make sure your data is arranged into a format acceptable for train test split. In scikit-learn, this consists of separating your full data set into “Features” and “Target.”

**2. Split The Data**
Split the data set into two pieces — a training set and a testing set. This consists of random sampling without replacement about 75 percent of the rows (you can vary this) and putting them into your training set. The remaining 25 percent is put into your test set. Note that the colors in “Features” and “Target” indicate where their data will go (“X_train,” “X_test,” “y_train,” “y_test”) for a particular train test split.

**3. Train The Model**
Train the model on the training set. This is “X_train” and “y_train” in the image.

**4. Test The Model**
Test the model on the testing set (“X_test” and “y_test” in the image) and evaluate the performance.

## Confusion Matrix

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. Some features of Confusion matrix are given below:

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.
- The matrix is divided into two dimensions, that are predicted values and actual values along with the total number of predictions.
- Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
- It looks like the below table:

| n = total predictions | Actual: No | Actual: Yes |
|---|---|---|
| Predicted: No | True Negative | False Positive |
| Predicted: Yes | False Negative | True Positive |

The above table has the following cases:

- True Negative: Model has given prediction No, and the real or actual value was also No.
- True Positive: The model has predicted yes, and the actual value was also true.
- False Negative: The model has predicted no, but the actual value was Yes, it is also called Type-II error.
- False Positive: The model has predicted Yes, but the actual value was No. It is also called a Type-I error.

**Accuracy Score:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN}$$

**Conclusion:** Thus we implemented a neural network-based classifier, compute the accuracy score and confusion matrix.