**Experiment No. 04**

**Aim**: Implement Artificial Neural Network (ANN) for Multiclass Classification using Keras Library

**Objectives**:

1. Understand the concept of Artificial Neural Network for multiclass classification.
2. Learn to preprocess and prepare data for ANN analysis in a multiclass setting.
3. Implement ANN using the Keras library for multiclass classification.
4. Evaluate the performance of the ANN model in handling multiple classes.
5. Gain practical experience in predicting multiclass outcomes using the trained model.

**Theory**:

**Artificial Neural Network (ANN):**

Artificial Neural Network is a versatile machine learning algorithm capable of handling various tasks, including multiclass classification. It consists of interconnected nodes organized in layers.

**Preprocessing Data:**

Data preprocessing for multiclass classification involves tasks such as handling missing values, scaling, and encoding categorical variables. Proper preprocessing ensures the ANN can effectively learn patterns from the data.

**Implementing ANN:**

Keras, a high-level neural networks API, simplifies the implementation of ANN for multiclass classification. It provides tools for building, training, and evaluating complex neural network architectures.

**Model Architecture:**

Designing the architecture of an ANN for multiclass classification includes choosing the number of layers, nodes in each layer, and activation functions. Keras allows for easy customization to suit the specific problem.

**Model Evaluation:**

Performance metrics such as accuracy, precision, recall, and F1-score are essential for assessing how well the ANN can classify multiple classes. These metrics aid in understanding the overall effectiveness of the model.

**Implementation Steps:**

1. Import necessary libraries (NumPy, pandas, and Keras).
2. Load and preprocess the dataset suitable for multiclass classification.
3. Build the neural network model with appropriate adjustments for multiclass scenarios.
4. Compile the model with a suitable loss function and optimizer.
5. Train the model on the training data.
6. Evaluate the model on the testing data and analyze multiclass performance metrics.
7. Fine-tune the model if necessary based on the evaluation results.

**Conclusion**:

Implementing Artificial Neural Network using the Keras library for multiclass classification expands the capabilities of machine learning models. This experiment provides hands-on experience in designing and evaluating neural networks for scenarios involving multiple classes. The insights gained can be applied to a wide range of multiclass classification problems.