Practical no: 3

Name: Saloni Vishwakarma
Batch-Roll No: C1-13

Aim: To implement a Machine Learning Classification model using Logistic regression and K-nearest neighbor (KNN) algorithms

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class.

Logistic Regression:

- It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities.It maps any real value into another value within a range of 0 and 1. o The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit.In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

K-nearest neighbor (KNN) algorithms

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

Distance Metrics Used in KNN Algorithm

- Euclidean Distance-This metric helps us calculate the net displacement done between the two states of an object.
- Manhattan Distance-This metric is calculated by summing the absolute difference between the coordinates of the points in n-dimensions.
- Minkowski Distance- distance/ similarity measurement between two points in the normed vector space (N dimensional real space) and is a generalization of the Euclidean distance and the Manhattan distance.

Accuracy, precision, recall, and F1-score are common metrics used to evaluate the performance of classification models, and they can be calculated based on the information in a confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification algorithm. It is particularly useful when dealing with imbalanced datasets.
Calculating these metrics using a confusion matrix:
True Positives (TP): The number of correctly predicted positive (class 1) instances.
True Negatives (TN): The number of correctly predicted negative (class 0) instances.
False Positives (FP): The number of instances that were incorrectly predicted as positive when they were actually negative. (Type I error)
False Negatives (FN): The number of instances that were incorrectly predicted as negative when they were actually positive. (Type II error)
Now, using these terms, we can calculate the metrics:
Accuracy: Accuracy measures the overall correctness of the classification model.
Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision: Precision measures the accuracy of positive predictions. It tells you how many of the predicted positive instances were actually positive.
Precision = TP / (TP + FP)
Recall (Sensitivity or True Positive Rate): Recall measures the ability of the model to identify all relevant instances. It tells you how many of the actual positive instances were correctly predicted.
Recall = TP / (TP + FN)
F1-Score: The F1-Score is the harmonic mean of precision and recall. It provides a balance between precision and recall, especially when you need to account for imbalanced datasets.
F1-Score = 2 * (Precision * Recall) / (Precision + Recall)
These metrics provide a comprehensive understanding of a model's performance:
Accuracy is a measure of overall correctness.
Precision emphasizes the ability to make positive predictions correctly.
Recall focuses on the model's ability to find all relevant instances.
F1-Score balances precision and recall when there is an uneven class distribution or when both false positives and false negatives are costly.

These metrics are valuable for evaluating a classifier's performance, and the choice of which one to emphasize depends on the specific problem and the trade-offs between precision and recall in that context.