K-Nearest Neighbours

**Objective:**

The objective of this assignment is to implement and evaluate the K-Nearest Neighbours algorithm for classification using the given datasets

**Dataset:**

Need to Classify the animal type

**Tasks:**

1. Analyse the data using the visualizations

2. Preprocess the data by handling missing values & Outliers, if any.

3. Split the dataset into training and testing sets (80% training, 20% testing).

4. Implement the K-Nearest Neighbours algorithm using a machine learning library like scikit-learn On training dataset

5. Choose an appropriate distance metric and value for K.

6. Evaluate the classifier's performance on the testing set using accuracy, precision, recall, and F1-score metrics.

7. Visualize the decision boundaries of the classifier.

**Interview Questions:**

1. What are the key hyperparameters in KNN?

Answer: The key hyperparameters in K-Nearest Neighbors (KNN) are:

1. Number of Neighbors (k):
   1. This is the number of nearest neighbors to consider for making the prediction. It is the most important hyperparameter in KNN. Choosing the right value of k is crucial for the algorithm's performance. A small value of k (e.g., k=1) can lead to a model that is too complex (overfitting), while a large value of k can result in a model that is too simple (underfitting).
2. Distance Metric:
   1. This determines how the distance between points is calculated. Common distance metrics include:
      1. Euclidean Distance: The straight-line distance between two points.
      2. Manhattan Distance: The sum of the absolute differences of their coordinates.
      3. Chebyshev Distance: The maximum absolute difference between the coordinates.
      4. Minkowski Distance: A generalization of both Euclidean and Manhattan distances.
      5. Hamming Distance: Used for categorical variables; it counts the number of positions at which the corresponding elements are different.
3. Weighting Function:
   1. This determines how the contributions of the neighbors are weighted. Common options include:
      1. Uniform Weight: All neighbors contribute equally.
      2. Distance Weight: Closer neighbors contribute more than farther ones, often using the inverse of the distance.
4. Algorithm for Finding Nearest Neighbors:
   1. This can affect the efficiency of the model, especially for large datasets:
      1. Brute Force: A straightforward approach that calculates the distance to each point in the dataset.
      2. k-d Tree: A tree-based approach that can speed up the search for neighbors.
5. Ball Tree: Another tree-based method suitable for high-dimensional data.
6. What distance metrics can be used in KNN?

Answer: In KNN, various distance metrics can be used to measure the similarity or dissimilarity between data points. The choice of distance metric can significantly affect the performance of the KNN algorithm. Here are some commonly used distance metrics:

1. Euclidean Distance:

(,) =

* The most common distance metric, representing the straight-line distance between two points in Euclidean space.

1. **Manhattan Distance** (L1 distance):

(,) =

* Also known as the L1 norm or city block distance, it is the sum of the absolute differences of their coordinates.

1. Chebyshev Distance:

(,) =

* The maximum absolute difference between any pair of coordinates.

1. **Minkowski Distance**:

(,) =

* A generalization of both Euclidean and Manhattan distances. When p=2p=2p=2, it is equivalent to Euclidean distance, and when p=1p=1p=1, it is equivalent to Manhattan distance.

1. Hamming Distance:

* Used for categorical variables, it counts the number of positions at which the corresponding elements are different.

(,) =

where is the indicator function that returns 1 if and 0 otherwise.