task_9.2

Machine Learning (WiSe 2025/2026)

Author: Suvansh Shukla
Matriculation No. 256245

Assignment 9 Task 2

Part A

Though KNN uses may use majority vote for labeling, there may be scenarios where the majority class may be misleading. One such scenario is when there is class imbalance in the dataset, leading to one class to be more numerous than others. This means that when trying to label the data point in question, even though it may be surrounded by instances from the more numerous class, it may actually belong to one of the less numerous (minority) classes. This would ultimately cause the instance to be mislabelled.

Part B

The following are some alternatives to majority voting for classification in KNN:

1. Using weighted nearest neighbour classification: here we assign each instance a weight depending on its proximity to the instance we are trying to classify. The weight assigned to each point is inversely proportional to its distance from the instance we are trying to classify, i.e. the nearer a point is the greater its weight.
2. Time based emphasis: here each neighbour is given weights depending on how recently it was added to the dataset, so we give more recent instances greater weights than older instances when trying to classify an instance.

Part C

We can modify a KNN classifier that was using majority voting to classify categorical values to a regressor by using weights. In this scenario we would calculate the distances of the k-nearest neighbours to the instance we are trying to classify then we would take the average of all the distances. In case we use weights we may take a weighted average and then this would be the predicted value of the instance to be used in regression.

Part D

It is important to normalize attribute values when using the KNN algorithm to avoid attributes with large scales to dominate the distance calculations. This basically means that if an attribute can take on very large linear values, then if we calculate its distance using Euclidean distance, its square will no doubt have a larger effect on the final distance. Such dominance will cause the KNN algorithm to behave as if the other attributes have no significance in determining the similarity thus having no influence in the final classification. To avoid this we need to normalize attributes with large scales so that KNN can appropriately account for other attributes as well. Basically, KNN without normalization is unit-sensitive, not importance-sensitive.

Part E

KNN can be easily used to impute or determine missing values in a data set by:

- calculate distances between instance with missing values and the rest of the dataset using available attributes
- select the k-nearest neighbours that have values for the missing attribute
- use majority voting in case of categorical values or weighted mean for numerical values to predict the missing value of the attribute
- weights can be assigned based on proximity of the neighbour to the instance in question, i.e. closer means greater weight, farther means lower weight