Hanz Adrian S. Tagabucba
DATA103 S18

NOTEBOOK QUESTIONS

- **Question 1:**
    a. Target 0 is easily distinguishable among the three targets as its distribution does not overlap with the other targets in the dataset.
    b. The petal length and petal width are good distinguishing factors of the target's type of species.
    c. Target 1 and Target 2 should not be differentiated by its sepal width and sepal length due to frequent overlap distribution

- **Question 2:**

The test_size code determines the portion of data from the dataset used for testing. In the notebook, 0.2 was specified as the test size, meaning that 20% of the dataset would be utilized for testing and the remaining portion of data will be used for the training set. As for the random state, its function is similar to a random seed, which determines the number of shuffles the data has to go through before splitting the dataset to ensure accuracy and avoid bias.

- **Question 3**

The reason behind not needing to call a validation set, specifically for this dataset, is due to the simplicity and small size of the Iris Dataset. Hence, allowing the defined test set to serve as the validation set of the model. In addition, creating a validation set, despite the relatively small size of the dataset, would further reduce the amount of data that may be used for the training set.

- **Question 4**

As straightforward as it is, using a training set with unbalanced values may provide unreliable and inconsistent results, which may seriously impact the validity and reliability of the entire model.

- **Question 5**

```
 #A define function that indicates the parameters that will be used in the whole function - x, point, and p. Its goal is to return a list of distances in X.
def minkowski(X, point, p:int=2) -> list:

#This line calculates d between x and point using the minkowski distance.
```

```
abs(x-point) - It first calculates the abs value between any value in X and point.
**p - Next, raises the results to the corresponding power value.
.sum - Sums all of the results. Which then,
1/p - takes the corresponding p-root of the sum result.

   d = lambda x: (((abs(x-point))**p).sum())**(1/p)

   distances = list(map(d, X)) #Creates a list of all calculated distances from point to all values in X.

   return distances #Returns the list of distances calculated through the function
```

- **Question 6**

Based on previous statistical classes, a high correlation indicates a significant relationship between the variables or features in a model. A correlation value less than 0 implies a negative relationship, while a value larger than 0 implies a positive relationship between the model's variables. If we were to base it from the notebook, the correlation matrix of wine quality presents that fixed acidity and citric acid are positively correlated with each other with a correlational value of 0.67.

- **Question 7**

Based on the graph, both the train scores and test scores show a trend wherein the accuracy of the train score decreases over time as the k value increases. In contrast, the accuracy of the test score increases while the k value increases before it stabilizes. This occurs because the model gets less overfit and more general as the k value increases.

As for the optimal k of the model, based on the graph, the optimal k is between the values of 8 to 10 where it maintains a stabilized trend, providing balance and accuracy to the whole model.

300 WORD DISCUSSION

The K-Nearest Neighbors (KNN) is one of the simplest methods and models of machine learning, hence being called by some as a “lazy learner algorithm”. This algorithm is commonly used in a variety of supervised machine learning, classification, regression, and recommender system scenarios. Moreover, It is mainly used to classify a point based on its distance from multiple points of the dataset and predict the class type it belongs to. Other than its simple and intuitive nature, one of the algorithm’s advantages is its flexibility to adapt and adjust to newly added data. Another advantage would be its ability to perform a variety of distance metrics such as Hamming, Minkowski, euclidean, and Manhattan. Lastly, it does not require multiple hyperparameters as it only has a few default hyperparameters to consider which are K and the distance points. However, despite its advantages, performing KNN also comes with several disadvantages and pitfalls that are important to consider especially when it comes to the size and features of data that could greatly impact the efficiency of the model. A notorious disadvantage of the KNN algorithm would be its sensitivity to handle high dimensional and heavy amounts of data, called the curse of dimensionality. This becomes a problem because as the data and dimension increases the more sparse the data would become, making it difficult to classify especially for an algorithm that heavily relies on the proximity of data points. Moreover, another downside of KNN would be the expensive cost of computing, especially with larger datasets. In order to address these issues, several alternative methods could be implemented such as performing KD Trees and Ball Trees, choosing the appropriate number for k, and performing the right distance metric, in which all of these methods can influence the overall efficiency and validity of the model.