1) AI Prediction for k-Nearest Neighbours when k = 1 in order of appearance, the first 25 entries expect “Iris-setosa”, the next 25 expect “Iris-versicolor” and the last 25 expect “Iris-virginica”:

Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor

Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”

Accuracy: 68/75 = 90.67%

2) AI Prediction for k-Nearest Neighbours when k = 3 in order of appearance, the first 25 entries expect “Iris-setosa”, the next 25 expect “Iris-versicolor” and the last 25 expect “Iris-virginica”:

Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica Expected: “Iris-versicolor”
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor

Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-versicolor Expected: “Iris-virginica”
Iris-versicolor Expected: “Iris-virginica”
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica

Accuracy = 72/75 =96%

As seen from the two outputs of the AI’s predictions with a ‘k’ value of 1 and a ‘k’ value of 3, it is evident that having a ‘k’ value of 3 provides better accuracy for the predictions. From this example it provided about a 5.3% increase in the accuracy of the AI’s prediction.

3) Some advantages of using the k-Nearest Neighbour search technique is that it is very robust to noisy data and anomalies won't have big weights on the final prediction. This technique is good with large training datasets as more neighbours reduce uncertainties of the final prediction. Some disadvantages of this search technique includes determining a value for 'k', this causes complications as it may be hard to determine which 'k' will provide the most accurate result for the dataset. It is also very expensive to run this algorithm as we require iterating through all nodes in our training set for each node in our test set.

4) In order to apply the k-fold cross validation technique for this problem we would first split up our dataset into 'k' equal subsets, we would then assign the first subset to be our test set and use the other k-1 subsets as our training set. We would then iterate through and our second subset would then become our training set and the other remaining k-1 subsets to be our test set. After we completed k iterations we can average our results/accuracy to get our true accuracy for the AI.

5) I would use the k-means clustering method for the problem if there were no class labels. Firstly, since we know there are 3 types of flowers we are trying to identify (domain knowledge) we are going to set k to 3. We would set the values of 'k' randomly from the dataset. I would then iterate through all instances in the dataset and find the euclidean distance to each of the different k's and assign it to the cluster with the smallest distance. After iterating through the dataset and all instances are assigned to a cluster I would then replace the centroid to the centre of the cluster by finding the mean of all values assigned to that cluster. Repeat until the centroid is stable (convergence)