Pakpoom khamchuai
CPE 261754 Advanced Pattern Recognition, Term 1, Academic year 2023/2566
September 21, 2023

Computer Assignment 2.1: k-Nearest Neighbors

Link to assignment

1. K-Nearest Neighbors Algorithm

The k-nearest neighbors algorithm, often referred to as KNN or k-NN, is a non-parametric, supervised learning classifier. KNN performs well on the non-parametric data. Although the labels of the training data points are required to give predictions, as KNN uses the proximity of data points to classify or predict the grouping of a specific data point. Let us define data vector $\vec{x}$, number of data points N, and number of class C, KNN algorithm has steps as follows:

1) Given a value of k where $1 < k < N$, compute distances from a (test) data point vector $\vec{x}$ to other data points (train data points).
2) Select k train data points that has the nearest distance to $\vec{x}$. From these points, perform a majority voting on their class/label, then assign $\vec{x}$ to the class with the majority.

2. Algorithm

2.1. KNN algorithm. The KNN classifier is defined as a class `KNNclassifier` consisted of methods `__init__`, `__call__`, `compute__nearest_neighbors`, and `euclidean_distance`. When initialized, the class `KNNclassifier` initializes k. Then, when it is called with input train dataset and test data vector $\vec{x}$, the class calls method `compute_k_nearest_neighbors` to compute the Euclidean distance and returns k nearest neighbors. After that, it returns the class with maximum number of members in the set of k nearest neighbors as the predicted class for $\vec{x}$. The algorithm is presented below.

Algorithm 1 K-Nearest Neighbors Classifier

function KNNClassifier(k_{num})
    Initialize k to k_{num}
    return a function that takes a test sample and train data as input
function Call(test_sample, train_data)
    $k_{\text{nearest\_neighbors}} \leftarrow \text{compute\_k\_nearest\_neighbors}(\text{test\_sample}, \text{train\_data})$
    $class\_labels \leftarrow$ [sample[-1] for sample in $k_{\text{nearest\_neighbors}}$]
    $predicted\_label \leftarrow \text{int}(\max(\text{set}(class\_labels), \text{key} = \text{class\_labels.count}))$
    return $predicted\_label$
function EuclideanDistance(x_1, x_2)
    return $\sqrt{\sum (x_1 - x_2)^2}$
function ComputeKNearestNeighbors(test_sample, train_data)
    Initialize an empty list *distances*
    for train_sample_features in train_data **do**
        $distance \leftarrow \text{EuclideanDistance}(\text{test\_sample}, \text{train\_sample\_features})$
        Append (train_sample_features, distance) to *distances*
    Sort *distances* by the second element in each tuple
    $k_{\text{nearest\_neighbors}} \leftarrow$ [sample[0] for sample in *distances*[: k]]
    return $k_{\text{nearest\_neighbors}}$

2.2. 10-fold cross validation. To evaluate the classifier's performance, we perform 10% cross validation. The dataset is partitioned into 10 folds. For each fold, one fold is used to evaluate the classification performance with the true label, while the remaining 9 folds are used to compute the nearest neighbors. The accuracy of the classifier is presented as a percentage of the correctly classified samples.

3. Results

This section explains the datasets used to test the KNN classifier, then presents and discusses the results.

3.1. Datasets.

In this work we use 4 datasets to test the implemented classifier, which are TWOCLASS, IRIS, CROSS, and ELLIPSE datasets. The details of each dataset are shown in Table 1.

Table 1. Details of the datasets

Dataset	No. data	Feature	Class
TWOCLASS	200	4	2
IRIS	150	4	3
CROSS	200	2	2
ELLIPSE	100	2	2

3.2. Classification results.

In Table 2 we show the classification performance as a percentage of correctly classified data points. We compare the results from KNN using k from 2-5 and results from the Maximum Likelihood classifier (MLC) implemented in the previous assignment.

For TWOCLASS dataset, the KNN has an equal performance as the MLC for k =3 and 5 at 97.5 %, and about 0.5-1.0 % worse than MLC for others. For IRIS dataset, MLC is overall better than KNN for all compared k values with 97.93 % performance, while KNN's results are 94.67 and 96.67 %.

For CROSS dataset MLC achieved only 66.72 % where KNN could classify up to 93.0-95.0 % correctly. A similar outcome has also occurred with the ELLIPSE dataset that MLC's accuracy is 73.88 % and KNN's accuracy ranges from 76.0 - 88.0 %.

Table 2. Average classification accuracy (percent correctly classified). Bold face highlights the best performance.

Dataset	MLC	k = 2	k = 3	k = 4	k = 5
TWOCLASS	**97.5**	96.5	**97.5**	97.0	**97.5**
IRIS	**97.93**	94.67	96.67	96.67	96.67
CROSS	66.72	93.0	94.5	94.0	**95.0**
ELLIPSE	73.88	76.0	**88.0**	79.0	86.0

4. Analysis of experimental results

4.1. MLC.

The classification accuracy of MLC for the TWOCLASS and IRIS datasets is high thanks to the parametric form and the linearly separability of the features 3 and 4. This can be seen in subfigures 3 and 4 in the Figures 1 and 2.

In contrast, since the data from CROSS and ELLIPSE datasets is of a non-parametric form, and is not linearly-separable, so MLC could not perform well with these datasets. Another Bayes classifier named the mixture model is more preferable than MLC for the non-parametric data.

4.2. KNN.

Unlike MLC, the KNN classifier, which based its classification algorithm of a data point on the majority vote on class of the given nearest k points, performs up to 95% and 88.0% on CROSS and ELLIPSE datasets. However, in Table 2 we can see that the accuracy on ELLIPSE dataset is more varied with the changing values of k than other datasets. This could be caused by some data points that are located closer to the data points from other class than points from its own class (shown in Figure 6). This makes the majority vote results of the class for these data points giving out the incorrect classifications depending on the number of points used (or k).

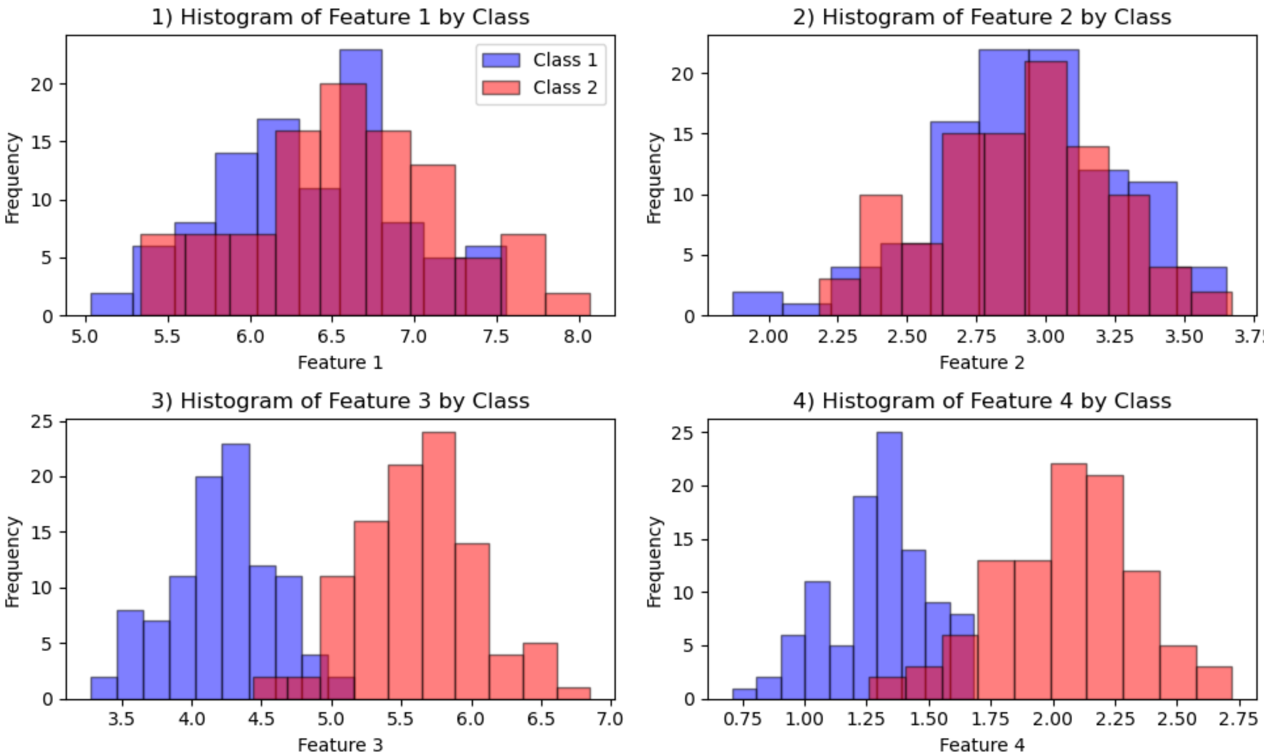


FIGURE 1. Histograms for each feature of TWOCLASS dataset.

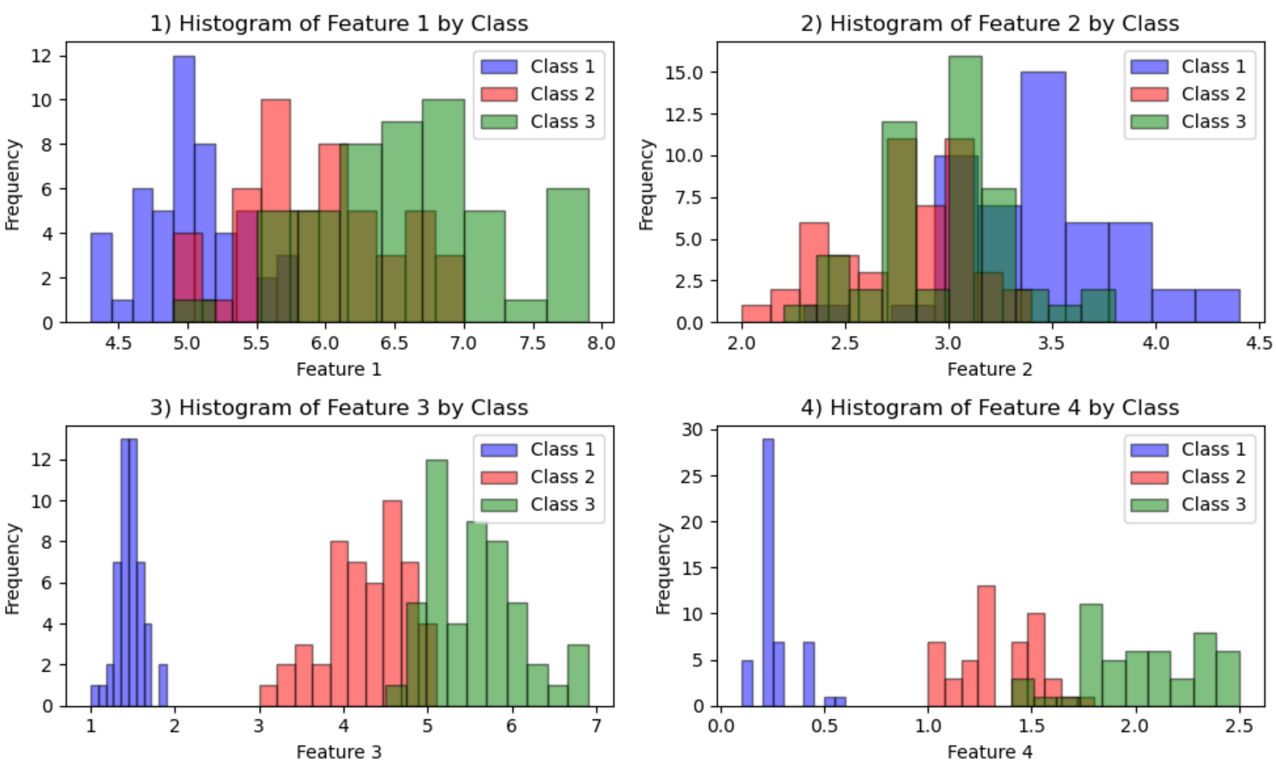


FIGURE 2. Histograms for each feature of IRIS dataset.

5. CODES

In following the codes and output are listed. The KNN classifier was implemented using Python on a Jupyter notebook. The external library `Numpy` was used despite restricted mainly to handle vector and matrix operations, and mathematical function such as a square root.

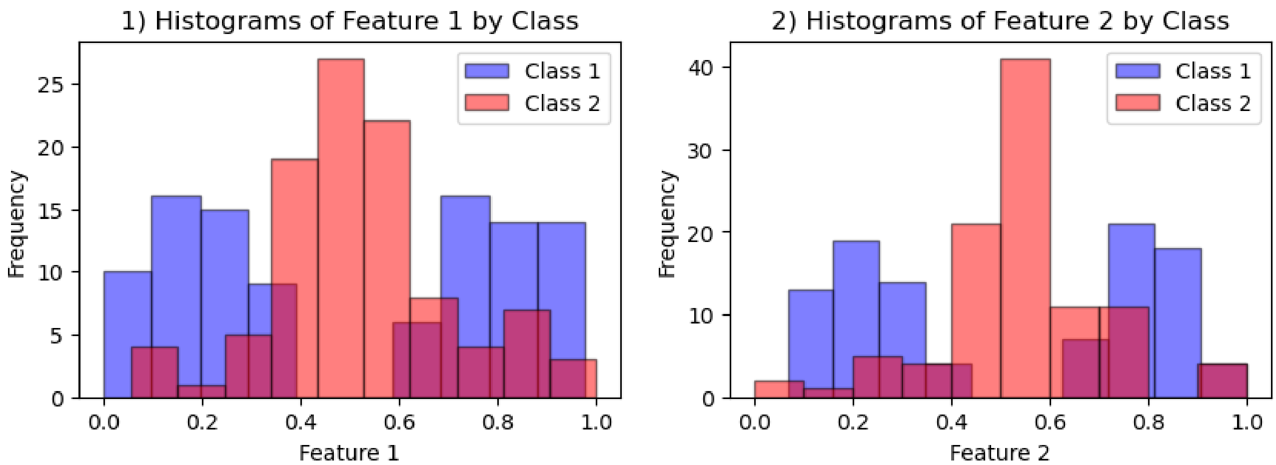


FIGURE 3. Histograms for each feature of CROSS dataset.

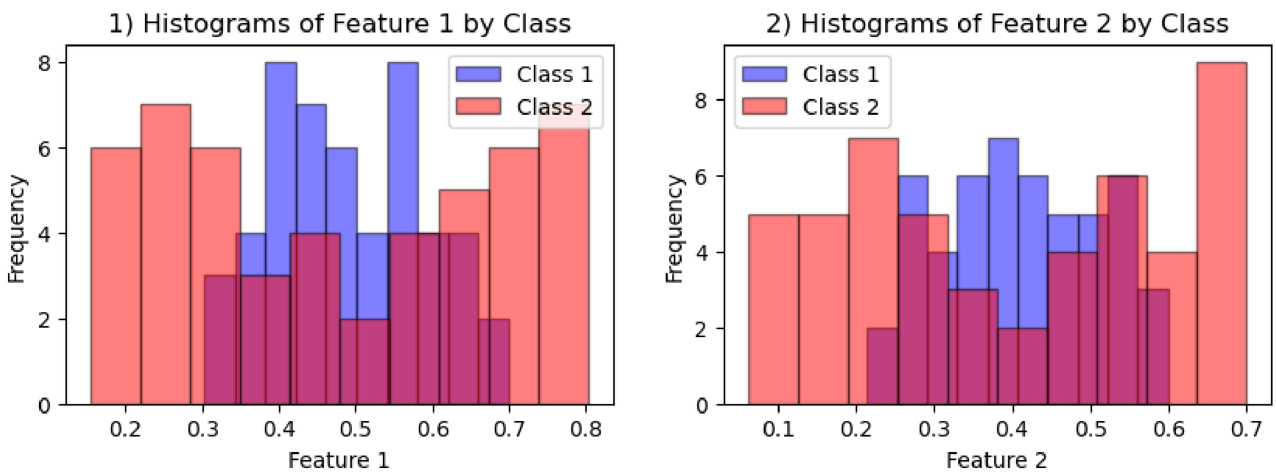


FIGURE 4. Histograms for each feature of ELLIPSE dataset.

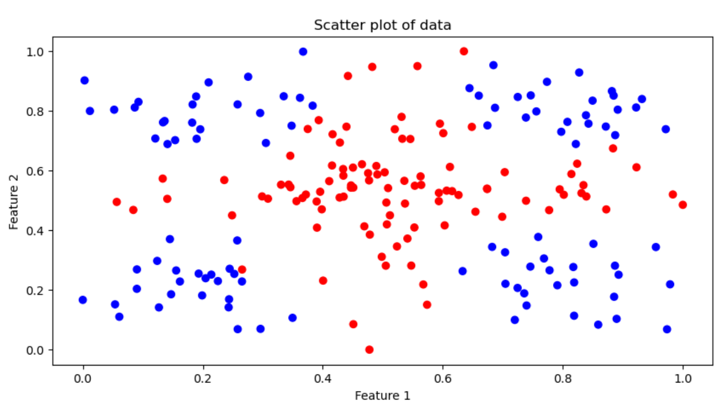


FIGURE 5. CROSS dataset

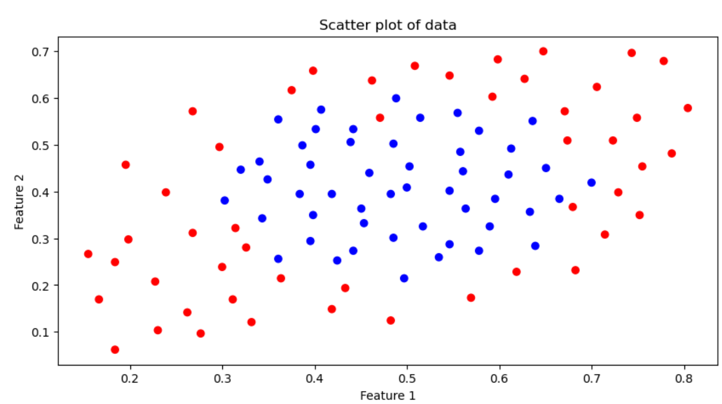


FIGURE 6. ELLIPSE dataset

The brief steps of the code are:

1) Read and process data
2) Call the Cross Validation function (`for` loop)
   (a) Initialize the KNN classifier class, input k and training data
   (b) Input the test data
   (c) Call the MLC function

3) Evaluate the predicted labels and display results

The following Jupyter notebook code list contains both codes and outputs. The KNN classifier was used to classify all 4 datasets using $k = 2, 3, 4, 5, 6, 7, 8, 9, 10$. The output also shows the predicted labels and accuracy for each test sample for each fold.

HW-2-1-clean

September 21, 2023

1 Computer Assignment 2 Problem 1

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import os
```

2 Maximum Likelihood

From previous homework, used for comparison

[2]:

```
# Step 1: Estimate Parameters (Mean vectors and Covariance matrices)
def estimate_parameters(data):
    num_features = len(data[0]) - 1  # Exclude the last column (class label)
    num_classes = int(max(data, key=lambda x: x[-1])[-1])  # Assuming class␣
 ↪labels are 1-indexed

    mean_vectors = {i: np.zeros(num_features) for i in range(1, num_classes +␣
 ↪1)}
    covariance_matrices = {i: np.zeros((num_features, num_features)) for i in␣
 ↪range(1, num_classes + 1)}
    class_counts = {i: 0 for i in range(1, num_classes + 1)}

    # Calculate the sum of feature values for each class
    for row in data:
        class_label = int(row[-1])
        class_counts[class_label] += 1
        for i in range(num_features):
            mean_vectors[class_label][i] += row[i]

    # Calculate the mean vectors
    for class_label in mean_vectors:
        mean_vectors[class_label] /= class_counts[class_label]

    # Calculate the covariance matrices
    for row in data:
        class_label = int(row[-1])
        x_minus_mean = row[:-1] - mean_vectors[class_label]
```

```python
        x_minus_mean = x_minus_mean.reshape((-1, 1))  # Convert to column vector
        covariance_matrices[class_label] += np.dot(x_minus_mean, x_minus_mean.T)

    for class_label in covariance_matrices:
        covariance_matrices[class_label] /= (class_counts[class_label] - 1)

    return mean_vectors, covariance_matrices

# Step 2: Minimum Risk Bayes Decision Theoretic Classifier
def multivariate_normal_pdf(x, mean, covariance_matrix):
    # Calculate the multivariate normal probability density function (PDF) for a given test sample 'x'
    # with the given mean and covariance matrix.
    k = len(x)
    coefficient = 1.0 / ((2 * np.pi) ** (k / 2) * np.linalg.det(covariance_matrix))

    # Calculate (x - mean)
    x_minus_mean = x - mean

    # Calculate the inverse of the covariance matrix
    inv_covariance = np.linalg.inv(covariance_matrix)

    # Calculate the Mahalanobis distance squared
    mahalanobis_dist_sq = np.dot(x_minus_mean, np.dot(inv_covariance, x_minus_mean))

    # Calculate the exponent
    exponent = -0.5 * mahalanobis_dist_sq

    return coefficient * np.exp(exponent)

def minimum_risk_classifier(test_sample, mean_vectors, covariance_matrices, prior_probabilities):
    num_classes = len(mean_vectors)
    risks = [0] * num_classes

    for class_label in range(1, num_classes + 1):
        mean_vector = np.array(mean_vectors[class_label])
        covariance_matrix = np.array(covariance_matrices[class_label])

        # Calculate the multivariate normal PDF for the current class
        pdf = multivariate_normal_pdf(test_sample, mean_vector, covariance_matrix)

        # Calculate the risk for the current class, which is the negative log-PDF plus the log-prior probability.
```

```
        risks[class_label - 1] = -np.log(pdf) + np.
↪log(prior_probabilities[class_label])

    # Choose the class with the minimum risk as the predicted class label.
    predicted_label = np.argmin(risks) + 1
    return predicted_label
```

3 k-NN classifier

[3]:
```
class KNNClassifier:
    # k-Nearest Neighbors Classifier
    # first, initial the class with k value
    # then, call the class with test sample and train data
    def __init__(self, k_num=3):
        self.k = k_num

    def __call__(self, test_sample, train_data):
        k_nearest_neighbors = self.compute_k_nearest_neighbors(test_sample,␣
↪train_data) # Get the k nearest neighbors
        class_labels = [sample[-1] for sample in k_nearest_neighbors] # Get the␣
↪class labels of the k nearest neighbors
        predicted_label = int(max(set(class_labels), key=class_labels.count)) #␣
↪Majority voting
        return predicted_label

    def euclidean_distance(self, x1, x2):
        # Calculate the Euclidean distance between two vectors
        return np.sqrt(np.sum((x1 - x2) ** 2))

    def compute_k_nearest_neighbors(self, test_sample, train_data):
        # Compute the k nearest neighbors of the given test sample
        distances = []

        for train_sample_features in train_data:
            test_sample_ = test_sample[:-1] # Exclude the last column (class␣
↪label)
            train_sample_features_ = train_sample_features[:-1] # Exclude the␣
↪last column (class label)
            distance = self.euclidean_distance(test_sample_,␣
↪train_sample_features_) # Calculate the Euclidean distance
            distances.append((train_sample_features, distance)) # Add the␣
↪distance along with the sample features to the list

        distances.sort(key=lambda x: x[1]) # Sort the list by the distances
        k_nearest_neighbors = [sample[0] for sample in distances[:self.k]]  #␣
↪Get the k nearest neighbors
```

```
        return k_nearest_neighbors
```

4 10-fold cross validation

[4]:
```
# perform 10-fold cross validation on maximum likelihood classifier and KNN␣
↪classifier
def cross_validation(data, fold=10, k=3, name='Twoclass', plot=False):
    print(f"Performing {fold}-fold cross validation with k = {k}")
    # shuffle data before cross validation
    np.random.shuffle(data)
    fold_size = len(data) // fold
    accuracy_scores_knn = []
    accuracy_scores_mlc = []

    for i in range(fold):
        # split data into training and test data
        start = i * fold_size
        end = (i + 1) * fold_size
        data_test_fold = data[start:end]
        data_train_fold = np.concatenate([data[:start], data[end:]])
        y_test_fold = data_test_fold[:, -1]

        ############################################ k-NN classifier␣
↪##################################################
        print(f"\nFold {i+1}")
        knn = KNNClassifier(k_num=k) # Initialize the k-NN classifier with k = 3
        y_pred = [knn(x, data_train_fold) for x in data_test_fold] # Predict␣
↪the class labels of the test data
        # display results for each test sample
        print(f"Predicted labels: {y_pred}")
        accuracy = np.sum(y_pred == y_test_fold) / len(y_test_fold) # Calculate␣
↪the accuracy
        # Display results for the current fold
        print(f"Number of misclassified samples: {np.sum(y_pred !=␣
↪y_test_fold)}")

        accuracy_scores_knn.append(accuracy) # Store the accuracy for the␣
↪current fold
        print(f"k-NN accuracy: {100*accuracy:.2f} %")

        ###################################### Maximum likelihood classifier␣
↪############################################
        # calculate prior probabilities from data_train_fold
        y_train_fold = data_train_fold[:, -1]
        prior_probabilities = {i: np.sum(y_train_fold == i) / len(y_train_fold)␣
↪for i in np.unique(y_train_fold)}
```

```
        # print(f"Prior probabilities: {prior_probabilities}")
        mean_vectors, covariance_matrices = estimate_parameters(data_train_fold)

        mlc_predicted_labels = []  # List to store the predicted labels for the current fold
        for sample in data_test_fold:
            test_sample = sample[:-1]
            # true_label = int(sample[-1])
            predicted_label = minimum_risk_classifier(test_sample, mean_vectors, covariance_matrices, prior_probabilities)
            mlc_predicted_labels.append(predicted_label)  # Store the predicted label for the current sample

        mlc_accuracy = 1 - (np.sum(mlc_predicted_labels != y_test_fold) / len(y_test_fold))

        # Display results for the current fold
        print(f"MLC accuracy: {100*mlc_accuracy:.2f} %")
        accuracy_scores_mlc.append(mlc_accuracy)
        ################################# end of Maximum likelihood classifier ###########################################

        if not plot:
            continue
        else:
            # plot the results of each fold, comparing the two classifiers
            # using 2 subplots
            # x axis: feature 3
            # y axis: feature 4
            # title: fold i
            # subplot 1: k-NN classifier
            # subplot 2: MLC classifier
            plt.figure(figsize=(10, 5))
            plt.suptitle(f"Fold {i+1}")
            plt.subplot(1, 2, 1)
            plt.title(f"k-NN classifier (k = {k}): {accuracy:.2f}")
            plt.xlabel("Feature 3")
            plt.ylabel("Feature 4")
            plt.scatter(data_train_fold[:, 2], data_train_fold[:, 3], c=data_train_fold[:, -1])
            # test data as star with increasing size
            plt.scatter(data_test_fold[:, 2], data_test_fold[:, 3], s=150, marker="*", c=y_pred)
            plt.subplot(1, 2, 2)
            plt.title(f"MLC classifier: {mlc_accuracy:.2f}")
            plt.xlabel("Feature 3")
```

```
            plt.ylabel("Feature 4")
            plt.scatter(data_train_fold[:, 2], data_train_fold[:, 3],␣
 ↪c=data_train_fold[:, -1])
            plt.scatter(data_test_fold[:, 2], data_test_fold[:, 3], s=150,␣
 ↪marker="*", c=mlc_predicted_labels) # PiYG
            plt.legend(("Training", "Test"))
            os.makedirs(f"HW1-{name}", exist_ok=True)
            plt.savefig(f"HW1-{name}/fold_{i+1}.png")
            # plt.show()
            plt.close()

    avg_accuracy_knn = np.mean(accuracy_scores_knn)
    avg_accuracy_mlc = np.mean(accuracy_scores_mlc)
    print(f"\nk-NN average accuracy: {100*avg_accuracy_knn:.2f} %")
    print(f"MLC average accuracy: {100*avg_accuracy_mlc:.2f} %\n")
    return avg_accuracy_knn, avg_accuracy_mlc
```

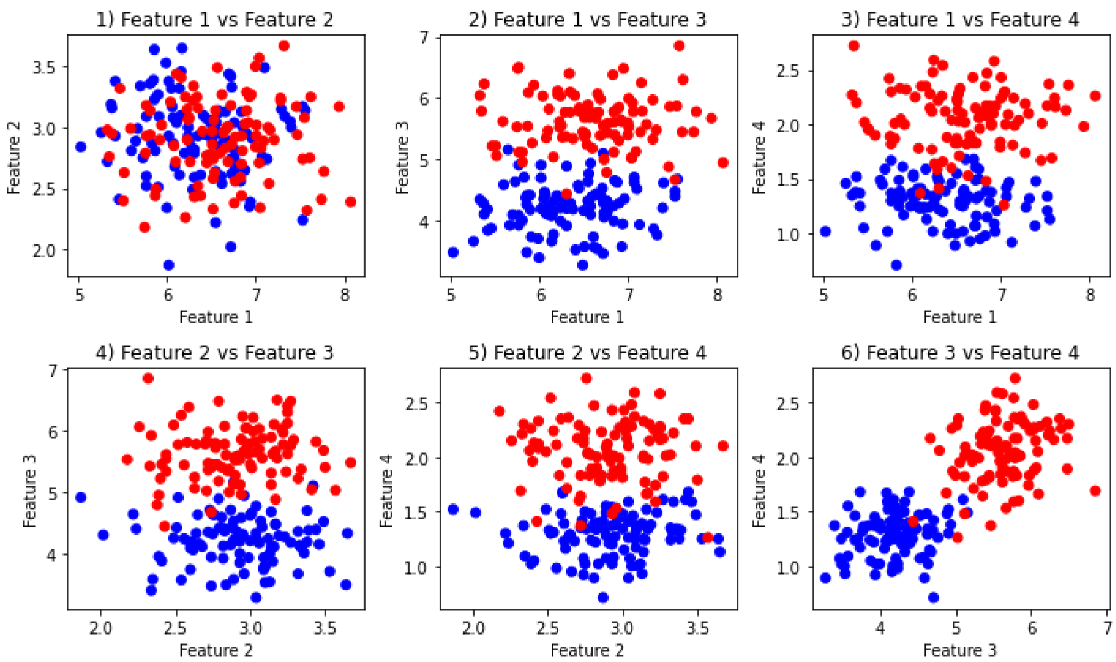
5 Twoclass dataset

[5]:
```
file_path = "TWOCLASS.dat"

try:
    with open(file_path, "r") as file:
        content = file.read()
        data = content.split()
        # print(data)
except FileNotFoundError:
    print("File not found!")

# drop the first 6 elements from list 'data'
data = data[6:]

data_processed = []

for i in range(0, len(data), 5):
    data_processed.append([float(data[i]), float(data[i+1]), float(data[i+2]), \
                           float(data[i+3]), int(data[i+4])])

# print length of data
print("Length of data: ", len(data_processed))
```

```
Length of data:  200
```

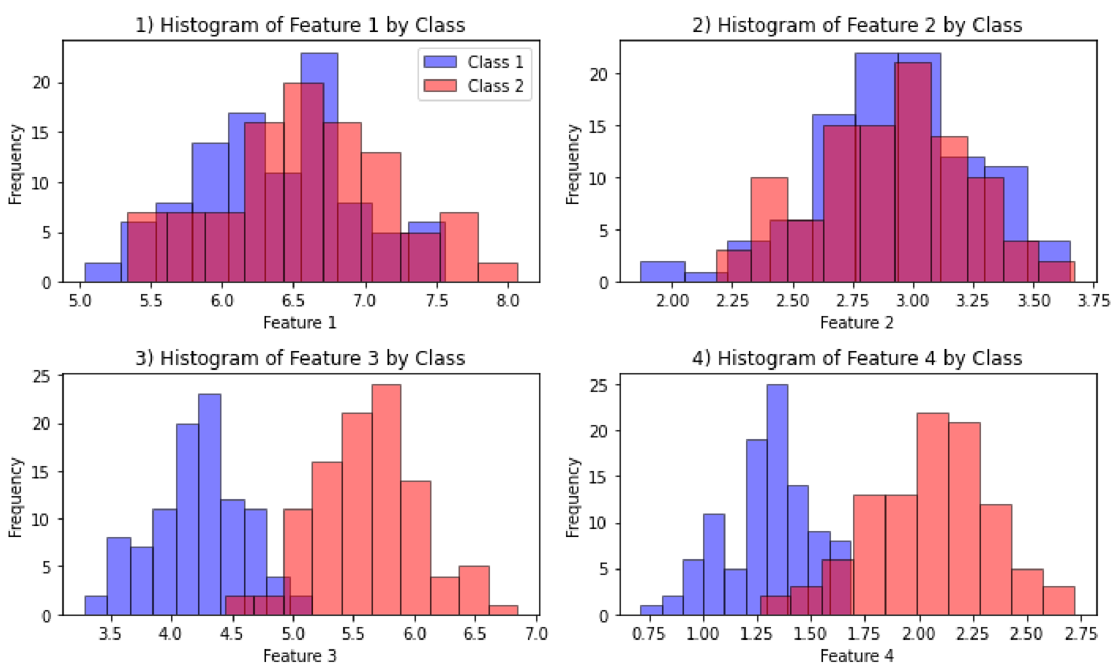
[6]:
```
# make subplots of scatter plot of data with class labels for all pairs of␣
 ↪features
# 1 vs 2, 1 vs 3, 1 vs 4, 2 vs 3, 2 vs 4, 3 vs 4
```

```
# 6 subplots in total
# use 2 for loops to plot

# plot the data
plt.figure(figsize=(10, 6))
# plt.suptitle("Scatter plot of data with class labels")
count = 1
for i in range(0, 4):
    for j in range(i+1, 4):
        plt.subplot(2, 3, count)
        plt.title(f"{count}) Feature {i+1} vs Feature {j+1}")
        plt.xlabel(f"Feature {i+1}")
        plt.ylabel(f"Feature {j+1}")
        plt.scatter([sample[i] for sample in data_processed], [sample[j] for␣
↪sample in data_processed], \
                        c=[sample[-1] for sample in data_processed], cmap="bwr")

        count += 1

plt.tight_layout()
```



[7]:
```
# Separate data by class
class1_data = [sample[:-1] for sample in data_processed if sample[-1] == 1]
class2_data = [sample[:-1] for sample in data_processed if sample[-1] == 2]
```

```python
# Features (replace these labels with your actual feature names)
feature_labels = ['Feature 1', 'Feature 2', 'Feature 3', 'Feature 4']

# Create a subplot of 2x2 graphs, each with a size of 10x10
fig, axs = plt.subplots(2, 2, figsize=(10, 6))

# Plot histograms for each feature
for i in range(len(feature_labels)):
    ax = axs[i // 2, i % 2]   # Get the appropriate subplot
    ax.hist([x[i] for x in class1_data], bins=10, alpha=0.5, color='blue', label='Class 1', edgecolor='black')
    ax.hist([x[i] for x in class2_data], bins=10, alpha=0.5, color='red', label='Class 2', edgecolor='black')
    ax.set_xlabel(feature_labels[i])
    ax.set_ylabel('Frequency')
    ax.set_title(f'{i+1}) Histogram of {feature_labels[i]} by Class')
    if i == 0:
        ax.legend()

    # ax.grid(True)

plt.tight_layout()
plt.show()
```



5.1 Results

```
[8]: # cross_validation(np.array(data_processed), k = 3, plot = True, name =␣
      ↪'Twoclass')
```

```
[9]: # run cross validation from k = 2 to 10
     # collect the average accuracy for each k
     k_list = [2, 3, 4, 5, 6, 7, 8, 9, 10]
     avg_accuracy_knn_list = []
     avg_accuracy_mlc_list = []
     for k in k_list:
         avg_accuracy_knn, avg_accuracy_mlc = cross_validation(np.
      ↪array(data_processed), fold=10, k=k, name='Twoclass', plot=False)
         print("==================================================")
         avg_accuracy_knn_list.append(avg_accuracy_knn)
         avg_accuracy_mlc_list.append(avg_accuracy_mlc)

     print(f'\nk: {k_list}')
     print(f'Average accuracy for k-NN classifier: {100*np.
      ↪array(avg_accuracy_knn_list)} %')
     print(f'Average accuracy for MLC classifier: {100*np.
      ↪mean(avg_accuracy_mlc_list)} %\n')
```

```
Performing 10-fold cross validation with k = 2

Fold 1
Predicted labels: [1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 2
Predicted labels: [1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 4
Predicted labels: [2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 5
Predicted labels: [2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 90.00 %

k-NN average accuracy: 97.00 %
MLC average accuracy: 97.50 %

==================================================
Performing 10-fold cross validation with k = 3

Fold 1
Predicted labels: [2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 2
Predicted labels: [1, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 85.00 %
MLC accuracy: 90.00 %

Fold 8
Predicted labels: [1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 90.00 %
```

```
Fold 10
Predicted labels: [1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.50 %
MLC average accuracy: 97.50 %

=================================================
Performing 10-fold cross validation with k = 4

Fold 1
Predicted labels: [1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 2
Predicted labels: [1, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 3
Predicted labels: [1, 2, 1, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 5
Predicted labels: [2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 7
Predicted labels: [2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 8
Predicted labels: [1, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.00 %
MLC average accuracy: 98.00 %

=================================================
Performing 10-fold cross validation with k = 5

Fold 1
Predicted labels: [1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 2
Predicted labels: [1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 3
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 4
Predicted labels: [1, 1, 2, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 6
Predicted labels: [2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [1, 1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.00 %
MLC average accuracy: 98.00 %

=================================================
Performing 10-fold cross validation with k = 6
```

```
Fold 1
Predicted labels: [2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 90.00 %

Fold 2
Predicted labels: [2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 4
Predicted labels: [1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 8
Predicted labels: [2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 9
Predicted labels: [2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.50 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 7

Fold 1
Predicted labels: [2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 4
Predicted labels: [1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %
```

```
Fold 6
Predicted labels: [1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [2, 1, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 9
Predicted labels: [1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 10
Predicted labels: [2, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

k-NN average accuracy: 97.00 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 8

Fold 1
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %
```

```
Fold 3
Predicted labels: [1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 5
Predicted labels: [1, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 95.00 %

Fold 6
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [1, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %
```

```
k-NN average accuracy: 97.50 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 9

Fold 1
Predicted labels: [2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [2, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 90.00 %
MLC accuracy: 95.00 %

Fold 4
Predicted labels: [1, 2, 1, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 5
Predicted labels: [1, 2, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %
```

```
Fold 8
Predicted labels: [1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 85.00 %
MLC accuracy: 95.00 %

Fold 10
Predicted labels: [1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 96.50 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 10

Fold 1
Predicted labels: [1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 3
Predicted labels: [1, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 4
Predicted labels: [1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %
```

```
Fold 5
Predicted labels: [1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 6
Predicted labels: [2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 95.00 %

Fold 7
Predicted labels: [1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 95.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.00 %
MLC average accuracy: 97.50 %

=================================================

k: [2, 3, 4, 5, 6, 7, 8, 9, 10]
Average accuracy for k-NN classifier: [97.  97.5 97.  97.  97.5 97.  97.5 96.5
97. ] %
Average accuracy for MLC classifier: 97.83333333333333 %
```

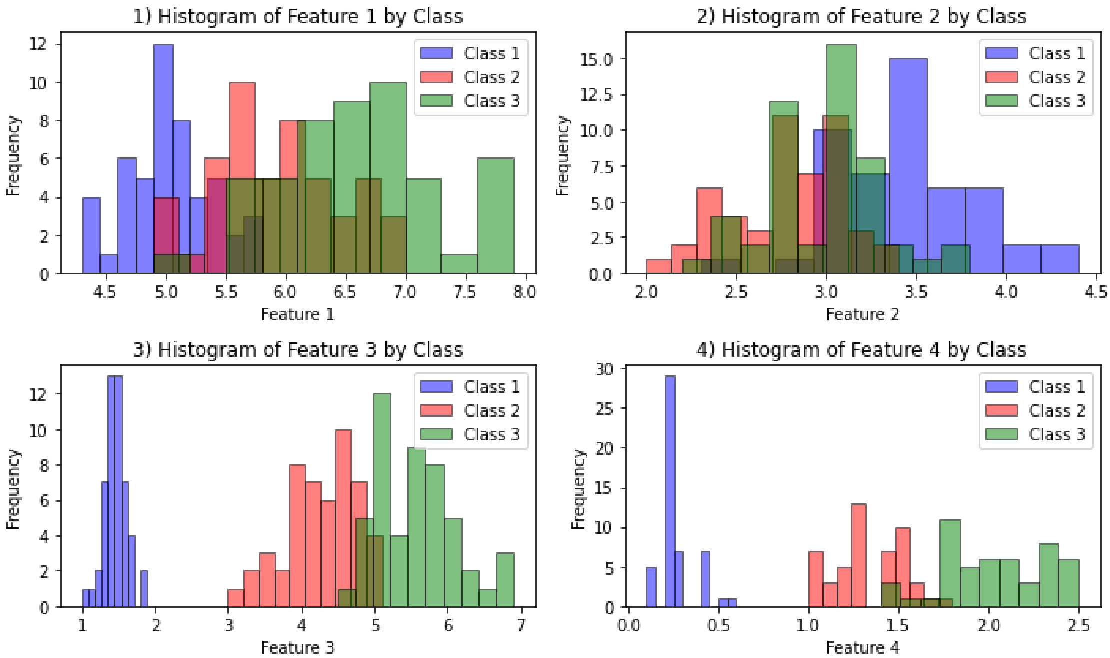
6 IRIS dataset

[10]:
```python
file_path = "iris.pat"

try:
    with open(file_path, "r") as file:
        content = file.read()
        data = content.split()
        # print(data)
except FileNotFoundError:
    print("File not found!")

header = data[:5]
data = data[5:]

data_processed = []

for i in range(0, len(data), 5):
    data_processed.append([float(data[i]), float(data[i+1]), float(data[i+2]), \
                           float(data[i+3]), int(data[i+4])])

# print length of data
print("Length of data: ", len(data_processed))
```

```
Length of data:  150
```

[11]:
```python
# Separate data by class
class1_data = [sample[:-1] for sample in data_processed if sample[-1] == 1]
class2_data = [sample[:-1] for sample in data_processed if sample[-1] == 2]
class3_data = [sample[:-1] for sample in data_processed if sample[-1] == 3]

# Features (replace these labels with your actual feature names)
feature_labels = ['Feature 1', 'Feature 2', 'Feature 3', 'Feature 4']

# Create a subplot of 2x2 graphs, each with a size of 10x10
fig, axs = plt.subplots(2, 2, figsize=(10, 6))

# Plot histograms for each feature
for i in range(len(feature_labels)):
    ax = axs[i // 2, i % 2]  # Get the appropriate subplot
    ax.hist([x[i] for x in class1_data], bins=10, alpha=0.5, color='blue',␣
 ↪label='Class 1', edgecolor='black')
    ax.hist([x[i] for x in class2_data], bins=10, alpha=0.5, color='red',␣
 ↪label='Class 2', edgecolor='black')
    ax.hist([x[i] for x in class3_data], bins=10, alpha=0.5, color='green',␣
 ↪label='Class 3', edgecolor='black')
    ax.set_xlabel(feature_labels[i])
    ax.set_ylabel('Frequency')
```

```
    ax.set_title(f'{i+1}) Histogram of {feature_labels[i]} by Class')
    ax.legend()
    # ax.grid(True)

plt.tight_layout()
plt.show()
```



6.1 Results

[12]:
```
# cross_validation(np.array(data_processed), k = 3, name='IRIS', plot = True)
```

[13]:
```
# run cross validation from k = 2 to 10
# collect the average accuracy for each k
k_list = [2, 3, 4, 5, 6, 7, 8, 9, 10]
avg_accuracy_knn_list = []
avg_accuracy_mlc_list = []
for k in k_list:
    avg_accuracy_knn, avg_accuracy_mlc = cross_validation(np.
 ↪array(data_processed), \
                                                        fold=10, k=k,␣
 ↪name='IRIS', plot=False)
    print("==================================================")
    avg_accuracy_knn_list.append(avg_accuracy_knn)
    avg_accuracy_mlc_list.append(avg_accuracy_mlc)
```

```python
print(f'\nk: {k_list}')
print(f'Average accuracy for k-NN classifier: {100*np.
 ↪array(avg_accuracy_knn_list)} %')
print(f'Average accuracy for MLC classifier: {100*np.
 ↪mean(avg_accuracy_mlc_list)} %\n')
```

```
Performing 10-fold cross validation with k = 2

Fold 1
Predicted labels: [2, 1, 1, 3, 1, 2, 2, 2, 1, 1, 3, 2, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [3, 1, 3, 1, 1, 1, 2, 3, 1, 3, 2, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 3, 2, 1, 3, 1, 3, 1, 2, 1, 2, 2, 3, 2, 3]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 93.33 %

Fold 4
Predicted labels: [1, 1, 2, 3, 3, 3, 2, 1, 3, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 3, 1, 3, 2, 1, 3, 3, 1, 3, 2, 2, 2, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [2, 3, 3, 3, 1, 2, 2, 1, 2, 2, 3, 1, 1, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 7
Predicted labels: [1, 1, 2, 2, 2, 2, 3, 1, 3, 3, 1, 1, 1, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [1, 2, 2, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [1, 1, 3, 2, 2, 1, 3, 2, 2, 3, 2, 3, 1, 3, 1]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 93.33 %

Fold 10
Predicted labels: [2, 2, 3, 3, 2, 3, 3, 3, 2, 1, 2, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

k-NN average accuracy: 94.67 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 3

Fold 1
Predicted labels: [1, 1, 3, 2, 1, 2, 1, 1, 3, 2, 1, 1, 3, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 2
Predicted labels: [1, 1, 3, 2, 3, 2, 3, 3, 3, 2, 3, 3, 2, 2, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 3, 1, 3, 2, 2, 1, 2, 3, 3, 2, 2, 3, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [3, 1, 2, 1, 2, 1, 1, 2, 3, 3, 1, 3, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
```

```
MLC accuracy: 93.33 %

Fold 5
Predicted labels: [2, 3, 2, 3, 2, 2, 1, 3, 1, 1, 2, 3, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 6
Predicted labels: [3, 3, 3, 2, 3, 3, 2, 1, 2, 3, 3, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [3, 3, 2, 3, 2, 2, 3, 1, 1, 2, 3, 1, 3, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [3, 1, 1, 3, 1, 1, 1, 3, 3, 2, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [3, 1, 2, 3, 1, 2, 3, 2, 1, 1, 2, 1, 2, 3, 2]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 3, 2, 2, 3, 1, 3, 2, 3, 2, 3, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 96.00 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 4

Fold 1
Predicted labels: [3, 1, 2, 3, 3, 3, 1, 1, 2, 1, 1, 3, 1, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [3, 1, 2, 2, 2, 2, 2, 3, 2, 3, 1, 3, 2, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 2, 1, 1, 1, 3, 2, 2, 1, 3, 1, 2, 3, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 3, 3, 1, 3, 2, 2, 3, 1, 1, 3, 3, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 3, 2, 2, 1, 2, 2, 3, 1, 2, 1, 1, 3, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [2, 3, 2, 2, 2, 1, 2, 3, 1, 1, 3, 1, 3, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [3, 2, 2, 1, 2, 2, 3, 2, 2, 3, 2, 1, 3, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [3, 1, 3, 1, 3, 2, 1, 2, 1, 3, 2, 1, 3, 3, 2]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 86.67 %

Fold 9
Predicted labels: [3, 1, 2, 2, 1, 1, 1, 3, 2, 1, 2, 1, 1, 3, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [3, 3, 2, 3, 3, 3, 2, 2, 3, 3, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 86.67 %

k-NN average accuracy: 96.67 %
MLC average accuracy: 97.33 %

==================================================
Performing 10-fold cross validation with k = 5

Fold 1
Predicted labels: [3, 1, 2, 2, 3, 1, 2, 3, 3, 1, 3, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [3, 1, 3, 2, 3, 1, 2, 3, 3, 1, 1, 3, 3, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 1, 3, 3, 3, 2, 2, 3, 1, 3, 2, 2, 3, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 3, 3, 2, 2, 1, 1, 2, 3, 3, 2, 2, 3, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 93.33 %

Fold 5
Predicted labels: [1, 2, 1, 3, 1, 1, 2, 2, 3, 3, 2, 1, 1, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 6
Predicted labels: [2, 3, 3, 1, 1, 3, 1, 2, 1, 1, 2, 1, 2, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 93.33 %

Fold 7
Predicted labels: [1, 3, 2, 3, 1, 2, 3, 3, 1, 3, 1, 2, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 3, 2, 3, 2]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 93.33 %

Fold 9
Predicted labels: [2, 3, 1, 3, 2, 2, 1, 3, 3, 1, 3, 3, 3, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [3, 1, 1, 1, 1, 2, 3, 1, 3, 2, 1, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 96.67 %
MLC average accuracy: 97.33 %

==================================================
Performing 10-fold cross validation with k = 6

Fold 1
Predicted labels: [2, 3, 2, 1, 2, 3, 3, 2, 2, 2, 1, 3, 1, 3, 2]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 86.67 %

Fold 2
Predicted labels: [3, 2, 3, 2, 3, 1, 2, 1, 2, 3, 1, 1, 1, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 3, 1, 1, 3, 3, 3, 1, 1, 3, 2, 3, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 3, 1, 2, 1, 2, 1, 1, 3, 2, 3, 2, 3, 3, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [1, 3, 1, 1, 3, 1, 2, 3, 3, 1, 3, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [1, 2, 3, 3, 1, 2, 3, 1, 1, 3, 2, 2, 3, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 3, 2, 3, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 3, 2, 1, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 9
Predicted labels: [1, 3, 1, 2, 1, 3, 1, 1, 3, 2, 1, 2, 1, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 2, 1, 2, 3, 1, 2, 3, 2, 1, 3, 1, 2, 3, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

k-NN average accuracy: 96.67 %
MLC average accuracy: 98.00 %

==================================================
```

```
Performing 10-fold cross validation with k = 7

Fold 1
Predicted labels: [3, 2, 3, 1, 3, 3, 1, 1, 2, 2, 2, 2, 3, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [2, 2, 2, 2, 1, 2, 3, 1, 1, 3, 1, 1, 3, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 1, 3, 1, 2, 3, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [1, 3, 3, 3, 2, 1, 3, 1, 1, 2, 1, 1, 1, 2, 3]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 86.67 %

Fold 5
Predicted labels: [3, 3, 2, 3, 2, 3, 3, 3, 1, 3, 3, 1, 2, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [3, 3, 3, 1, 2, 3, 3, 1, 2, 2, 2, 3, 2, 1, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [2, 3, 1, 3, 2, 2, 1, 2, 1, 2, 3, 3, 2, 3, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [3, 1, 1, 1, 3, 1, 3, 2, 2, 3, 2, 2, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 93.33 %

Fold 9
Predicted labels: [1, 1, 1, 2, 3, 1, 1, 2, 3, 2, 1, 3, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 1, 3, 3, 3, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 97.33 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 8

Fold 1
Predicted labels: [2, 3, 3, 2, 2, 1, 3, 3, 1, 2, 3, 1, 2, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 2
Predicted labels: [1, 3, 3, 2, 3, 3, 3, 2, 3, 2, 3, 1, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 3, 2, 3, 1, 2, 3, 2, 1, 2, 3, 1, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [2, 2, 1, 3, 1, 3, 2, 1, 1, 1, 1, 3, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [3, 3, 2, 3, 1, 1, 3, 3, 1, 1, 1, 1, 3, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 93.33 %

Fold 7
Predicted labels: [3, 1, 1, 2, 3, 1, 1, 3, 2, 2, 3, 1, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 8
Predicted labels: [3, 1, 2, 1, 1, 2, 1, 3, 3, 1, 1, 2, 2, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 9
Predicted labels: [2, 1, 2, 2, 3, 1, 3, 2, 2, 3, 2, 2, 2, 3, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 10
Predicted labels: [2, 3, 1, 1, 3, 1, 3, 1, 1, 1, 3, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 96.00 %
MLC average accuracy: 98.00 %

==================================================
Performing 10-fold cross validation with k = 9

Fold 1
Predicted labels: [3, 3, 2, 1, 1, 3, 1, 3, 3, 2, 3, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [3, 2, 1, 2, 1, 3, 3, 1, 2, 1, 1, 2, 3, 3, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 93.33 %

Fold 3
Predicted labels: [2, 3, 1, 1, 2, 3, 1, 3, 1, 1, 1, 1, 3, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [2, 2, 2, 3, 1, 2, 2, 3, 3, 1, 2, 2, 1, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 5
Predicted labels: [3, 2, 1, 3, 2, 3, 2, 1, 3, 1, 3, 2, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [3, 1, 3, 2, 3, 2, 2, 2, 2, 3, 2, 2, 1, 3, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [3, 3, 1, 2, 3, 2, 1, 1, 2, 3, 1, 1, 3, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 93.33 %

Fold 8
Predicted labels: [1, 3, 2, 1, 1, 1, 3, 1, 3, 2, 2, 1, 3, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 86.67 %
MLC accuracy: 93.33 %

Fold 9
Predicted labels: [1, 3, 2, 3, 2, 3, 2, 2, 2, 3, 3, 2, 3, 1, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 1, 3, 3, 1, 3, 1, 1, 3, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
```

```
MLC accuracy: 100.00 %

k-NN average accuracy: 97.33 %
MLC average accuracy: 98.00 %

=================================================
Performing 10-fold cross validation with k = 10

Fold 1
Predicted labels: [2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 2, 2, 3, 1, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 2
Predicted labels: [1, 3, 3, 3, 3, 3, 3, 1, 2, 2, 3, 3, 1, 3, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 3
Predicted labels: [2, 1, 2, 1, 2, 1, 2, 1, 3, 1, 1, 3, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 4
Predicted labels: [3, 3, 2, 3, 2, 2, 3, 2, 1, 3, 2, 2, 3, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 93.33 %

Fold 5
Predicted labels: [1, 1, 2, 1, 3, 1, 1, 3, 2, 3, 1, 1, 3, 2, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 6
Predicted labels: [1, 3, 3, 1, 1, 1, 2, 1, 2, 1, 1, 3, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

Fold 7
Predicted labels: [1, 1, 1, 2, 3, 2, 3, 2, 3, 1, 2, 1, 3, 2, 3]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
```

```
MLC accuracy: 93.33 %

Fold 8
Predicted labels: [3, 1, 2, 2, 2, 2, 3, 3, 1, 2, 3, 3, 1, 3, 3]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 93.33 %

Fold 9
Predicted labels: [2, 2, 2, 1, 1, 1, 3, 2, 2, 3, 1, 2, 3, 3, 2]
Number of misclassified samples: 1
k-NN accuracy: 93.33 %
MLC accuracy: 100.00 %

Fold 10
Predicted labels: [2, 3, 2, 1, 1, 3, 1, 3, 1, 2, 1, 1, 3, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 100.00 %
MLC accuracy: 100.00 %

k-NN average accuracy: 98.00 %
MLC average accuracy: 98.00 %

==================================================

k: [2, 3, 4, 5, 6, 7, 8, 9, 10]
Average accuracy for k-NN classifier: [94.66666667 96.         96.66666667
96.66666667 96.66666667 97.33333333
 96.         97.33333333 98.        ] %
Average accuracy for MLC classifier: 97.85185185185186 %
```

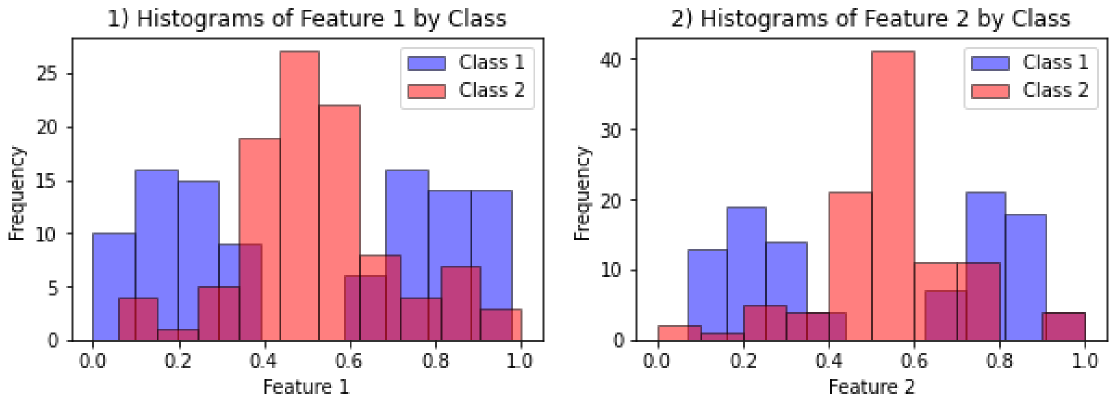
7 CROSS dataset

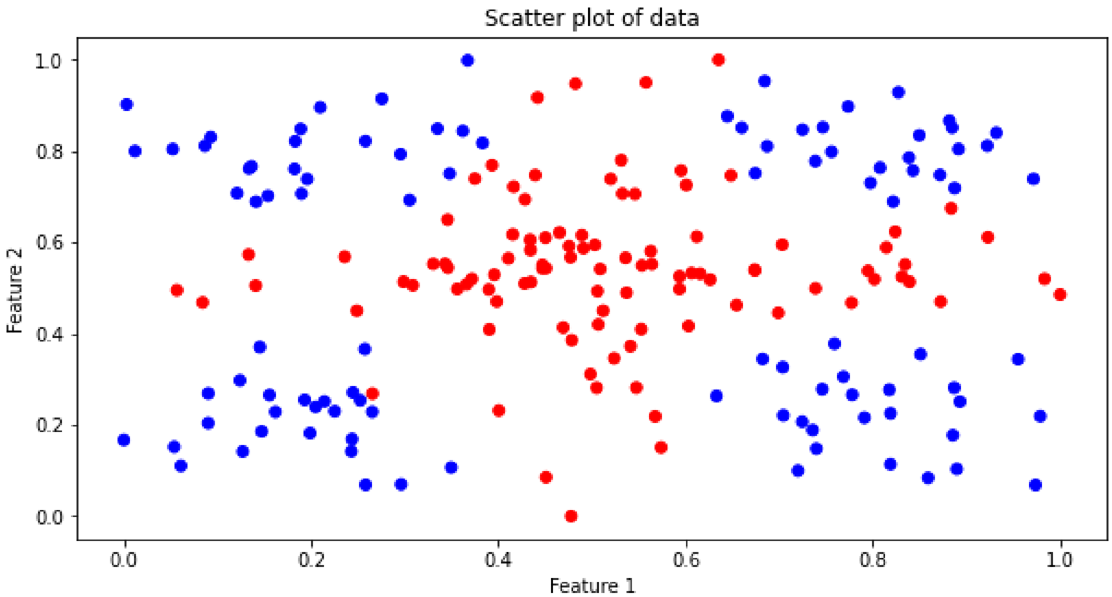
[14]:
```
file_path = "cross.pat"

try:
    with open(file_path, "r") as file:
        content = file.read()
        data = content.split()
        # print(data)
except FileNotFoundError:
    print("File not found!")

data = np.array(data).reshape(-1, 5).T
print(data.shape)
f1 = data[1].astype(float)
```

```
f2 = data[2].astype(float)
class_labels = data[4].astype(int) + 1 # Convert class labels from 0-indexed to␣
↪1-indexed
data_processed = np.array([f1, f2, class_labels]).T
```

(5, 200)

[15]:
```
# Separate data by class
class1_data = [sample[:-1] for sample in data_processed if sample[-1] == 1]
class2_data = [sample[:-1] for sample in data_processed if sample[-1] == 2]

# Features (replace these labels with your actual feature names)
feature_labels = ['Feature 1', 'Feature 2']

# Create a subplot of 2x2 graphs, each with a size of 10x10
fig, axs = plt.subplots(1, 2, figsize=(10, 3))

# Plot histograms for each feature
for i in range(len(feature_labels)):
    ax = axs[i]
    ax.hist([x[i] for x in class1_data], bins=10, alpha=0.5, color='blue',␣
↪label='Class 1', edgecolor='black')
    ax.hist([x[i] for x in class2_data], bins=10, alpha=0.5, color='red',␣
↪label='Class 2', edgecolor='black')
    ax.set_xlabel(feature_labels[i])
    ax.set_ylabel('Frequency')
    ax.set_title(f'{i+1}) Histograms of {feature_labels[i]} by Class')
    ax.legend()
    # ax.grid(True)

# plot scatter plot of data
plt.figure(figsize=(10, 5))
plt.title("Scatter plot of data")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.scatter(f1, f2, c=class_labels, cmap='bwr')
plt.show()
```





[16]:
```
# define the function again because the structure of data is different (only␣
 ↪plotting is changed)
# perform 10-fold cross validation on maximum likelihood classifier and KNN␣
 ↪classifier
def cross_validation(data, fold=10, k=3, name='CROSS', plot=False):
    print(f"Performing {fold}-fold cross validation with k = {k}")
    # shuffle data before cross validation
    np.random.shuffle(data)
    fold_size = len(data) // fold
    accuracy_scores_knn = []
    accuracy_scores_mlc = []
```

```
    for i in range(fold):
        # split data into training and test data
        start = i * fold_size
        end = (i + 1) * fold_size
        data_test_fold = data[start:end]
        data_train_fold = np.concatenate([data[:start], data[end:]])
        y_test_fold = data_test_fold[:, -1]

        ############################################# k-NN classifier␣
↪####################################################
        print(f"\nFold {i+1}")
        knn = KNNClassifier(k_num=k) # Initialize the k-NN classifier with k = 3
        y_pred = [knn(x, data_train_fold) for x in data_test_fold] # Predict␣
↪the class labels of the test data
        # display results for each test sample
        print(f"Predicted labels: {y_pred}")
        accuracy = np.sum(y_pred == y_test_fold) / len(y_test_fold) # Calculate␣
↪the accuracy
        # Display results for the current fold
        print(f"Number of misclassified samples: {np.sum(y_pred !=␣
↪y_test_fold)}")

        accuracy_scores_knn.append(accuracy) # Store the accuracy for the␣
↪current fold
        print(f"k-NN accuracy: {accuracy:.2f}")

        ###################################### Maximum likelihood classifier␣
↪#############################################
        # calculate prior probabilities from data_train_fold
        y_train_fold = data_train_fold[:, -1]
        prior_probabilities = {i: np.sum(y_train_fold == i) / len(y_train_fold)␣
↪for i in np.unique(y_train_fold)}
        # print(f"Prior probabilities: {prior_probabilities}")
        mean_vectors, covariance_matrices = estimate_parameters(data_train_fold)

        mlc_predicted_labels = []  # List to store the predicted labels for the␣
↪current fold
        for sample in data_test_fold:
            test_sample = sample[:-1]
            # true_label = int(sample[-1])
            predicted_label = minimum_risk_classifier(test_sample,␣
↪mean_vectors, covariance_matrices, prior_probabilities)
            mlc_predicted_labels.append(predicted_label)  # Store the predicted␣
↪label for the current sample
```

```
        mlc_accuracy = 1 - (np.sum(mlc_predicted_labels != y_test_fold) /␣
↪len(y_test_fold))

        # Display results for the current fold
        print(f"MLC accuracy: {mlc_accuracy:.2f}")
        accuracy_scores_mlc.append(mlc_accuracy)
        ################################# end of Maximum likelihood classifier␣
↪###########################################

        if not plot:
            continue
        else:
            # plot the results of each fold, comparing the two classifiers
            # using 2 subplots
            # x axis: feature 1
            # y axis: feature 2
            # training data: class 1: red, class 2: blue
            # test data: class 1: orange, class 2: green
            # title: fold i
            # subplot 1: k-NN classifier
            # subplot 2: MLC classifier
            plt.figure(figsize=(10, 5))
            plt.suptitle(f"Fold {i+1}")
            plt.subplot(1, 2, 1)
            plt.title(f"k-NN classifier (k = {k}): {accuracy:.2f}")
            plt.xlabel("Feature 1")
            plt.ylabel("Feature 2")
            plt.scatter(data_train_fold[:, 0], data_train_fold[:, 1],␣
↪c=data_train_fold[:, -1])
            # test data as star with increasing size
            plt.scatter(data_test_fold[:, 0], data_test_fold[:, 1], s=150,␣
↪marker="*", c=y_pred)
            plt.subplot(1, 2, 2)
            plt.title(f"MLC classifier: {mlc_accuracy:.2f}")
            plt.xlabel("Feature 1")
            plt.ylabel("Feature 2")
            plt.scatter(data_train_fold[:, 0], data_train_fold[:, 1],␣
↪c=data_train_fold[:, -1])
            plt.scatter(data_test_fold[:, 0], data_test_fold[:, 1], s=150,␣
↪marker="*", c=mlc_predicted_labels) # PiYG
            plt.legend(("Training", "Test"))
            os.makedirs(f"HW1-{name}", exist_ok=True)
            plt.savefig(f"HW1-{name}/fold_{i+1}.png")
            # plt.show()
            plt.close()
```

```
    avg_accuracy_knn = np.mean(accuracy_scores_knn)
    avg_accuracy_mlc = np.mean(accuracy_scores_mlc)
    print(f"\nk-NN average accuracy: {100*avg_accuracy_knn:.2f} %")
    print(f"MLC average accuracy: {100*avg_accuracy_mlc:.2f} %\n")
    return avg_accuracy_knn, avg_accuracy_mlc
```

7.1 Results

[17]:
```
# cross_validation(np.array(data_processed), k = 3, name='CROSS', plot = True)
```

[18]:
```
# run cross validation from k = 2 to 10
# collect the average accuracy for each k
k_list = [2, 3, 4, 5, 6, 7, 8, 9, 10]
avg_accuracy_knn_list = []
avg_accuracy_mlc_list = []
for k in k_list:
    avg_accuracy_knn, avg_accuracy_mlc = cross_validation(np.
 ↪array(data_processed), fold=10, k=k, name='CROSS', plot=False)
    print("==================================================")
    avg_accuracy_knn_list.append(avg_accuracy_knn)
    avg_accuracy_mlc_list.append(avg_accuracy_mlc)

print(f'\nk: {k_list}')
print(f'Average accuracy for k-NN classifier: {100*np.
 ↪array(avg_accuracy_knn_list)} %')
print(f'Average accuracy for MLC classifier: {100*np.
 ↪mean(avg_accuracy_mlc_list)} %\n')
```

```
Performing 10-fold cross validation with k = 2

Fold 1
Predicted labels: [2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.80

Fold 2
Predicted labels: [2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 3
Predicted labels: [2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70
```

```
Fold 4
Predicted labels: [1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.65

Fold 5
Predicted labels: [2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.60

Fold 6
Predicted labels: [1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 7
Predicted labels: [2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.50

Fold 8
Predicted labels: [1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 9
Predicted labels: [1, 1, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 10
Predicted labels: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.50

k-NN average accuracy: 93.50 %
MLC average accuracy: 65.00 %

=================================================
Performing 10-fold cross validation with k = 3
```

```
Fold 1
Predicted labels: [1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.80

Fold 2
Predicted labels: [1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 3
Predicted labels: [1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 4
Predicted labels: [1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.60

Fold 5
Predicted labels: [1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 6
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.75

Fold 7
Predicted labels: [1, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2]
Number of misclassified samples: 4
k-NN accuracy: 0.80
MLC accuracy: 0.65

Fold 8
Predicted labels: [1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.45
```

```
Fold 9
Predicted labels: [1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.55

Fold 10
Predicted labels: [1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.80

k-NN average accuracy: 93.50 %
MLC average accuracy: 67.00 %

=================================================
Performing 10-fold cross validation with k = 4

Fold 1
Predicted labels: [2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 2
Predicted labels: [1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.75

Fold 3
Predicted labels: [2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.60

Fold 4
Predicted labels: [1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 5
Predicted labels: [2, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.75
```

```
Fold 6
Predicted labels: [1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.65

Fold 7
Predicted labels: [2, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.60

Fold 8
Predicted labels: [1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.75

Fold 9
Predicted labels: [2, 1, 1, 1, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 10
Predicted labels: [2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

k-NN average accuracy: 94.50 %
MLC average accuracy: 68.00 %

==================================================
Performing 10-fold cross validation with k = 5

Fold 1
Predicted labels: [2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 2
Predicted labels: [2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.60
```

```
Fold 3
Predicted labels: [2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.75

Fold 4
Predicted labels: [2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.50

Fold 5
Predicted labels: [1, 1, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.80

Fold 6
Predicted labels: [2, 1, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 7
Predicted labels: [2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 8
Predicted labels: [2, 1, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.85

Fold 9
Predicted labels: [2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.55

Fold 10
Predicted labels: [1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70
```

```
k-NN average accuracy: 96.00 %
MLC average accuracy: 67.50 %

==================================================
Performing 10-fold cross validation with k = 6

Fold 1
Predicted labels: [1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.75

Fold 2
Predicted labels: [1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.75

Fold 3
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.70

Fold 4
Predicted labels: [2, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 5
Predicted labels: [1, 2, 1, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.60

Fold 6
Predicted labels: [1, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 7
Predicted labels: [1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.50
```

```
Fold 8
Predicted labels: [2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.60

Fold 9
Predicted labels: [2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.80

Fold 10
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.85

k-NN average accuracy: 95.00 %
MLC average accuracy: 68.50 %

==================================================
Performing 10-fold cross validation with k = 7

Fold 1
Predicted labels: [2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.55

Fold 2
Predicted labels: [1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.60

Fold 3
Predicted labels: [2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 4
Predicted labels: [2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.70
```

```
Fold 5
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.60

Fold 6
Predicted labels: [1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.75

Fold 7
Predicted labels: [1, 1, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.65

Fold 8
Predicted labels: [1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 9
Predicted labels: [2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 10
Predicted labels: [1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

k-NN average accuracy: 95.50 %
MLC average accuracy: 66.00 %

==================================================
Performing 10-fold cross validation with k = 8

Fold 1
Predicted labels: [2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.75
```

```
Fold 2
Predicted labels: [1, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 3
Predicted labels: [1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.50

Fold 4
Predicted labels: [2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 5
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 6
Predicted labels: [1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 4
k-NN accuracy: 0.80
MLC accuracy: 0.75

Fold 7
Predicted labels: [1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.55

Fold 8
Predicted labels: [1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.60

Fold 9
Predicted labels: [2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.80
```

```
Fold 10
Predicted labels: [1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

k-NN average accuracy: 94.00 %
MLC average accuracy: 66.50 %

=================================================
Performing 10-fold cross validation with k = 9

Fold 1
Predicted labels: [1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 2
Predicted labels: [1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 3
Predicted labels: [1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.70

Fold 4
Predicted labels: [1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.75

Fold 5
Predicted labels: [1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 6
Predicted labels: [1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65
```

```
Fold 7
Predicted labels: [2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 8
Predicted labels: [2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.50

Fold 9
Predicted labels: [1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.65

Fold 10
Predicted labels: [1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.75

k-NN average accuracy: 93.50 %
MLC average accuracy: 67.50 %

==================================================
Performing 10-fold cross validation with k = 10

Fold 1
Predicted labels: [1, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.70

Fold 2
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.65

Fold 3
Predicted labels: [1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.60
```

```
Fold 4
Predicted labels: [2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.70

Fold 5
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.60

Fold 6
Predicted labels: [2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.70

Fold 7
Predicted labels: [1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.95
MLC accuracy: 0.75

Fold 8
Predicted labels: [2, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 4
k-NN accuracy: 0.80
MLC accuracy: 0.60

Fold 9
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.85
MLC accuracy: 0.85

Fold 10
Predicted labels: [2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.90
MLC accuracy: 0.75

k-NN average accuracy: 91.50 %
MLC average accuracy: 69.00 %

==================================================
```

```
k: [2, 3, 4, 5, 6, 7, 8, 9, 10]
Average accuracy for k-NN classifier: [93.5 93.5 94.5 96.  95.  95.5 94.  93.5
91.5] %
Average accuracy for MLC classifier: 67.2222222222222 %
```

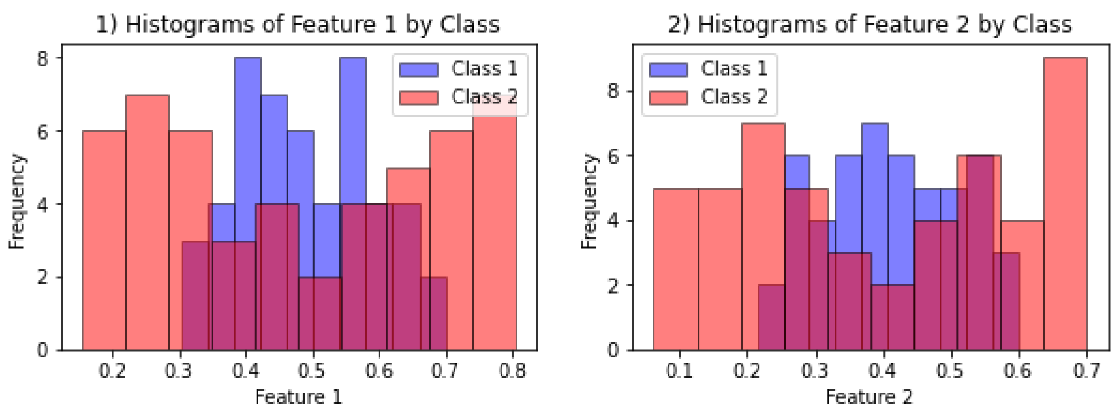
8 ELLIPSE dataset

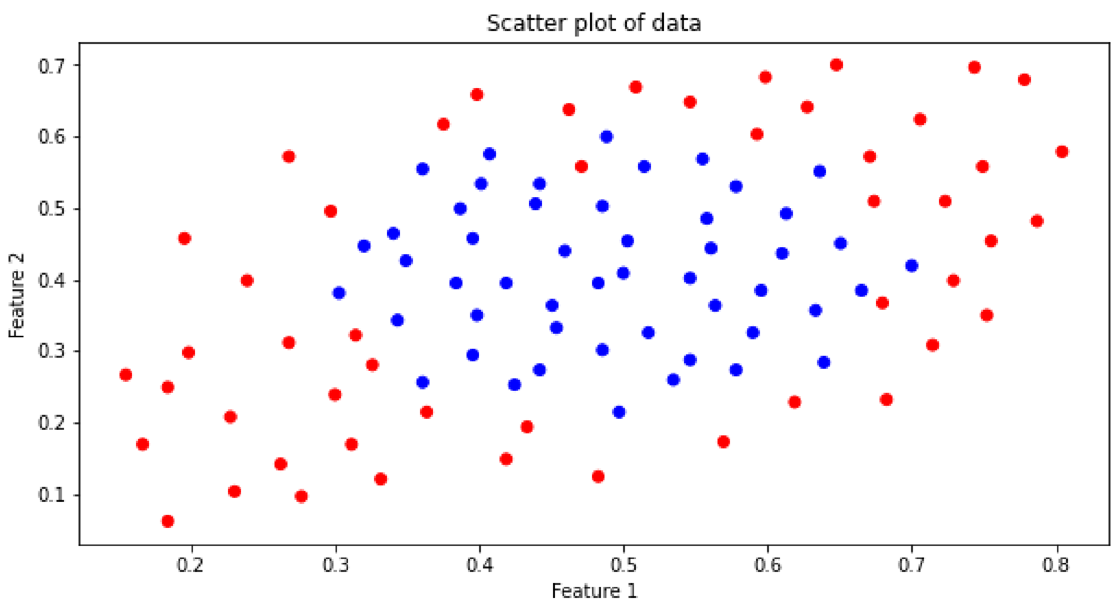
[19]:
```
file_path = "ellipse.pat"

try:
    with open(file_path, "r") as file:
        content = file.read()
        data = content.split()
        # print(data)
except FileNotFoundError:
    print("File not found!")

data = np.array(data).reshape(-1, 5).T
print(data.shape)
f1 = data[1].astype(float)
f2 = data[2].astype(float)
class_labels = data[4].astype(int) + 1 # Convert class labels from 0-indexed to␣
↪1-indexed
data_processed = np.array([f1, f2, class_labels]).T
```

```
(5, 100)
```

[20]:
```
# Separate data by class
class1_data = [sample[:-1] for sample in data_processed if sample[-1] == 1]
class2_data = [sample[:-1] for sample in data_processed if sample[-1] == 2]

# Features (replace these labels with your actual feature names)
feature_labels = ['Feature 1', 'Feature 2']

# Create a subplot of 2x2 graphs, each with a size of 10x10
fig, axs = plt.subplots(1, 2, figsize=(10, 3))

# Plot histograms for each feature
for i in range(len(feature_labels)):
    ax = axs[i]
    ax.hist([x[i] for x in class1_data], bins=10, alpha=0.5, color='blue',␣
↪label='Class 1', edgecolor='black')
    ax.hist([x[i] for x in class2_data], bins=10, alpha=0.5, color='red',␣
↪label='Class 2', edgecolor='black')
    ax.set_xlabel(feature_labels[i])
    ax.set_ylabel('Frequency')
```

```
    ax.set_title(f'{i+1}) Histograms of {feature_labels[i]} by Class')
    ax.legend()
    # ax.grid(True)

# plot scatter plot of data
plt.figure(figsize=(10, 5))
plt.title("Scatter plot of data")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.scatter(f1, f2, c=class_labels, cmap='bwr')
plt.show()
```





8.1 Results

[21]:
```
# cross_validation(np.array(data_processed), k = 3, name='ELLIPSE', plot = True)
```

[22]:
```
# run cross validation from k = 2 to 10
# collect the average accuracy for each k
k_list = [2, 3, 4, 5, 6, 7, 8, 9, 10]
avg_accuracy_knn_list = []
avg_accuracy_mlc_list = []
for k in k_list:
    avg_accuracy_knn, avg_accuracy_mlc = cross_validation(np.
 ↪array(data_processed), fold=10, k=k, name='ELLIPSE', plot=False)
    print("==================================================")
    avg_accuracy_knn_list.append(avg_accuracy_knn)
    avg_accuracy_mlc_list.append(avg_accuracy_mlc)

print(f'\nk: {k_list}')
print(f'Average accuracy for k-NN classifier: {100*np.
 ↪array(avg_accuracy_knn_list)} %')
print(f'Average accuracy for MLC classifier: {100*np.
 ↪mean(avg_accuracy_mlc_list)} %\n')
```

```
Performing 10-fold cross validation with k = 2

Fold 1
Predicted labels: [1, 2, 1, 2, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 2
Predicted labels: [1, 1, 1, 1, 2, 1, 1, 2, 1, 2]
Number of misclassified samples: 4
k-NN accuracy: 0.60
MLC accuracy: 0.70

Fold 3
Predicted labels: [2, 2, 2, 1, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 4
Predicted labels: [1, 1, 2, 1, 1, 1, 1, 2, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70
```

```
Fold 5
Predicted labels: [1, 1, 1, 2, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 6
Predicted labels: [1, 1, 1, 2, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

Fold 7
Predicted labels: [2, 1, 1, 2, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 8
Predicted labels: [2, 2, 1, 2, 2, 1, 2, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 9
Predicted labels: [1, 2, 2, 2, 2, 1, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.60

Fold 10
Predicted labels: [1, 1, 2, 2, 2, 2, 1, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

k-NN average accuracy: 79.00 %
MLC average accuracy: 74.00 %

==================================================
Performing 10-fold cross validation with k = 3

Fold 1
Predicted labels: [1, 1, 2, 1, 1, 2, 2, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.60
```

```
Fold 2
Predicted labels: [1, 2, 1, 2, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.80

Fold 3
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 4
Predicted labels: [1, 2, 2, 2, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.50

Fold 5
Predicted labels: [2, 2, 1, 2, 2, 1, 1, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 6
Predicted labels: [1, 1, 1, 2, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90

Fold 7
Predicted labels: [1, 1, 1, 1, 2, 1, 1, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.80

Fold 8
Predicted labels: [1, 2, 2, 2, 2, 2, 1, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 9
Predicted labels: [2, 1, 1, 2, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.80
```

```
Fold 10
Predicted labels: [2, 2, 1, 1, 2, 2, 2, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

k-NN average accuracy: 85.00 %
MLC average accuracy: 75.00 %

==================================================
Performing 10-fold cross validation with k = 4

Fold 1
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 5
k-NN accuracy: 0.50
MLC accuracy: 0.50

Fold 2
Predicted labels: [1, 1, 2, 1, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 3
Predicted labels: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.60

Fold 4
Predicted labels: [2, 2, 1, 1, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 5
Predicted labels: [2, 1, 2, 2, 2, 1, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 6
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90
```

```
Fold 7
Predicted labels: [1, 1, 1, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 8
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 2, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

Fold 9
Predicted labels: [2, 1, 2, 2, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.60

Fold 10
Predicted labels: [1, 2, 2, 1, 1, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

k-NN average accuracy: 81.00 %
MLC average accuracy: 74.00 %

=================================================
Performing 10-fold cross validation with k = 5

Fold 1
Predicted labels: [1, 2, 2, 1, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.80

Fold 2
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 3
Predicted labels: [1, 2, 1, 2, 2, 2, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.60
```

```
Fold 4
Predicted labels: [1, 1, 1, 1, 2, 1, 1, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 5
Predicted labels: [2, 2, 1, 1, 1, 2, 1, 1, 1, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 6
Predicted labels: [1, 2, 2, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90

Fold 7
Predicted labels: [1, 1, 1, 1, 2, 2, 2, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.60

Fold 8
Predicted labels: [2, 1, 2, 2, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.80

Fold 9
Predicted labels: [2, 2, 1, 2, 1, 2, 1, 2, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

Fold 10
Predicted labels: [1, 1, 1, 2, 1, 1, 1, 2, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

k-NN average accuracy: 85.00 %
MLC average accuracy: 76.00 %

==================================================
Performing 10-fold cross validation with k = 6
```

```
Fold 1
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90

Fold 2
Predicted labels: [2, 1, 1, 1, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 3
Predicted labels: [1, 1, 2, 1, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 4
Predicted labels: [1, 1, 1, 1, 2, 2, 2, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 5
Predicted labels: [1, 2, 1, 2, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 6
Predicted labels: [2, 1, 2, 2, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.80

Fold 7
Predicted labels: [1, 1, 1, 2, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 8
Predicted labels: [2, 1, 2, 1, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70
```

```
Fold 9
Predicted labels: [1, 1, 1, 1, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 4
k-NN accuracy: 0.60
MLC accuracy: 0.60

Fold 10
Predicted labels: [2, 2, 2, 2, 2, 2, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

k-NN average accuracy: 80.00 %
MLC average accuracy: 74.00 %

=================================================
Performing 10-fold cross validation with k = 7

Fold 1
Predicted labels: [1, 2, 1, 1, 1, 2, 2, 1, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 2
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 4
k-NN accuracy: 0.60
MLC accuracy: 0.60

Fold 3
Predicted labels: [1, 2, 1, 2, 1, 2, 1, 1, 2, 1]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

Fold 4
Predicted labels: [2, 1, 1, 1, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 5
Predicted labels: [2, 1, 1, 1, 1, 2, 2, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70
```

```
Fold 6
Predicted labels: [1, 1, 1, 2, 1, 2, 2, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 7
Predicted labels: [2, 1, 2, 1, 1, 2, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.60

Fold 8
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 2, 2, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 9
Predicted labels: [1, 1, 1, 1, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 10
Predicted labels: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.80

k-NN average accuracy: 85.00 %
MLC average accuracy: 75.00 %

==================================================
Performing 10-fold cross validation with k = 8

Fold 1
Predicted labels: [1, 1, 1, 1, 2, 2, 1, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 2
Predicted labels: [1, 2, 1, 1, 2, 1, 1, 1, 2, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70
```

```
Fold 3
Predicted labels: [1, 1, 1, 1, 2, 1, 2, 2, 2, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 4
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 2, 2]
Number of misclassified samples: 4
k-NN accuracy: 0.60
MLC accuracy: 0.60

Fold 5
Predicted labels: [2, 1, 1, 1, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 6
Predicted labels: [2, 2, 2, 2, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 7
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

Fold 8
Predicted labels: [1, 1, 2, 1, 1, 1, 2, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 9
Predicted labels: [1, 1, 1, 2, 2, 1, 2, 2, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90

Fold 10
Predicted labels: [2, 2, 1, 1, 2, 1, 2, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70
```

```
k-NN average accuracy: 78.00 %
MLC average accuracy: 74.00 %

==================================================
Performing 10-fold cross validation with k = 9

Fold 1
Predicted labels: [1, 2, 1, 1, 1, 2, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 2
Predicted labels: [2, 1, 1, 2, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 3
Predicted labels: [2, 2, 1, 1, 2, 2, 1, 1, 2, 2]
Number of misclassified samples: 5
k-NN accuracy: 0.50
MLC accuracy: 0.80

Fold 4
Predicted labels: [1, 2, 2, 2, 2, 1, 1, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 5
Predicted labels: [1, 2, 1, 2, 2, 2, 2, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.60

Fold 6
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 1, 2, 2]
Number of misclassified samples: 0
k-NN accuracy: 1.00
MLC accuracy: 0.90

Fold 7
Predicted labels: [2, 1, 1, 1, 2, 1, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70
```

```
Fold 8
Predicted labels: [1, 1, 2, 1, 1, 1, 2, 2, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

Fold 9
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 10
Predicted labels: [2, 1, 2, 2, 1, 2, 2, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.70

k-NN average accuracy: 84.00 %
MLC average accuracy: 76.00 %

=================================================
Performing 10-fold cross validation with k = 10

Fold 1
Predicted labels: [1, 2, 1, 1, 1, 1, 1, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.70

Fold 2
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 1, 2, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.80

Fold 3
Predicted labels: [1, 2, 2, 2, 1, 1, 2, 1, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90

Fold 4
Predicted labels: [2, 1, 2, 2, 1, 1, 1, 1, 1, 2]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.90
```

```
Fold 5
Predicted labels: [1, 1, 1, 1, 1, 1, 2, 1, 1, 2]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.60

Fold 6
Predicted labels: [1, 1, 2, 2, 1, 1, 2, 2, 1, 1]
Number of misclassified samples: 2
k-NN accuracy: 0.80
MLC accuracy: 0.50

Fold 7
Predicted labels: [1, 2, 2, 2, 1, 2, 1, 1, 1, 1]
Number of misclassified samples: 4
k-NN accuracy: 0.60
MLC accuracy: 0.60

Fold 8
Predicted labels: [2, 1, 1, 1, 1, 2, 1, 1, 1, 2]
Number of misclassified samples: 1
k-NN accuracy: 0.90
MLC accuracy: 0.90

Fold 9
Predicted labels: [2, 1, 2, 1, 1, 1, 1, 1, 2, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.80

Fold 10
Predicted labels: [2, 1, 2, 1, 1, 2, 1, 1, 1, 1]
Number of misclassified samples: 3
k-NN accuracy: 0.70
MLC accuracy: 0.70

k-NN average accuracy: 76.00 %
MLC average accuracy: 74.00 %

==================================================

k: [2, 3, 4, 5, 6, 7, 8, 9, 10]
Average accuracy for k-NN classifier: [79. 85. 81. 85. 80. 85. 78. 84. 76.] %
Average accuracy for MLC classifier: 74.66666666666667 %
```

[ ]: