Homework 4 Report

Group Members:
Kiri Woodruff
Angeles Marin Batana

Data preparation:
As a way of preparing our data for classification, we created a function that would classify the data objects in our csv with labels ‘A’ or ‘B’ such that 50 data objects would be labled ‘A’, and the remaining 50 would be labeled ‘B’ where data objects labeled ‘A’ would consist of individuals who are a healthy weight and label ‘B’ would consist of individuals who are at an unhealthy weight. As a part of data preparation, we wrote a program to help facilitate the labeling, where data objects would be labeled ‘A’ if they were between 50 and 80 (kg) for individuals under the age of 30, and 55-85 for individuals over 30. Class B would include individuals who were not in this healthy weight range, also for those over and under the age of 30. Our labels were A=Healthy weight and B = Unhealthy weight for all data objects in the dataset.

We assigned 50 of our total data objects to be used as training data, where we randomly selected 25 As and 25 Bs so that the remaining objects could be used as test data for predicting class labels.

Part 1: Using KNN; K=1
In order to implement the K Nearest Neighbor KNN data classification, we first computed a function to determine the euclidean distance for any 2 points. Then the steps of KNN are implemented, which are as follows:
1st, we computed the euclidean distance for a given test point and the training points for the training data with the function step_1
Then, we sorted the calculated distances so that we could identify the nearest neighbors with the function step_2.
Finally, we determined the label for the test point given the majority classes with the k-nearest neighbors. We used a binary classification for the binary labels ‘A’ and ‘B’ where the labels would be returned depending on who won the vote for neighbor when k = 1.

To test our prediction, we output the predictions as a list and to test the accuracy of our prediction, we created a funtion that would output the accuracy as a percentage given the true label and the algorithm’s prediction. Furthermore, our accuracy function calculated the overall classification accuracy for our prediction where k = 1, so our accuracy function printed the accuracy of class A, accuracy of class B, and the overall classification accuracy of the program.

Part 2: Using KNN; K=5
Finding K=5 was easier, since we already had the k nearest neighbor algorithm. All we had to do was slightly modify our prediction function for k = 5. We used the same accuracy function to test our k=5 prediction.

Part 3: Using Decision tree

In order to use the decision tree classifier in scikit-learn, we had to convert our categorical features into numerical ones, and we did this using one-hot encoding. We then trained our decision tree using our training data and printed the predictions. We used a similary accuracy function to test our decision tree predictions, only changing it minorly.

Part 4: Classifier with best performance:
Although the overall accuracy for our K=1, K=5, and decision tree classifier was equivalent, in a hypothetical situation where we would have to classify a large volume of data objects given many attributes, or more complex data, the decision tree classifier might have the best performance when compared to the KNN classification. This is because of its inherent ability to gather more complex relationships between complex data. For simple data, such as the one used in this assignment, both the decision tree classifier and the KNN classifier would produce relatively similar results, though when fewer neighbors are considered for the later classifier, the less impactful the decision of classification becomes due to this classifier relying on local neighbors to make predictions (for large datasets).

Part 5: Hands-on Experience with data classification:
In my hands-on experience with data classification, I've worked on various projects aimed at effectively categorizing and predicting outcomes based on differing attributes with various datasets. One notable project involved preprocessing and analyzing a dataset containing demographic information such as age and weight, for which I created their corresponding class labels using predetermined specifications for the class labeling.

To begin, I utilized Python and the Pandas library to preprocess the dataset, ensuring it was ready for classification tasks. This involved tasks like handling missing data, encoding categorical variables, and scaling numerical features. Once the data was prepared, I implemented classification algorithms to predict class labels based on the input features using the scikit-learn machine learning pandas library.

I employed both traditional machine learning algorithms and more advanced techniques. For instance, I used the k-nearest neighbors (k-NN) algorithm to classify data points based on their similarity to other points in the feature space, notably for various k values ranging from 1-10, allowing the exploration of various patterns in datasets. By calculating distances between data points, I could determine the most likely class for each observation.

In addition to KNN, I also implemented decision tree classifiers. These models recursively split the data based on features, creating a tree-like structure where each internal node represents a decision based on a feature, and each leaf node represents a class label.

Throughout the classification process, I paid close attention to model evaluation and performance metrics. I split the dataset into training and testing sets to assess the predictive ability of the models. By comparing predicted labels with base truth labels, I calculated metrics such as accuracy, precision, recall, and F1-score to gauge the effectiveness of the classifiers.

Lastly, I experimented with parameter tuning and model selection techniques to optimize classification performance. For instance, I varied the number of neighbors in KNN and the maximum depth of decision trees to find the most suitable configurations for the given dataset, finding new interesting patterns depending on the classification algorithm used.

Overall, my hands-on experience with data classification encompasses a comprehensive understanding of preprocessing techniques, implementation of various classification algorithms, rigorous model evaluation, and continuous optimization to achieve accurate and reliable predictions.