Evan Perez

The City College of New York

3/17/24

Senior Design 1

Dr. Yunhua Zhao

Assignment 1 Report

In this assignment, I was tasked with creating my own Python class and function for the K nearest neighbors machine learning algorithm. A brute force KNN algorithm was constructed for the binary classification of wine quality classes from a given dataset. With this dataset, I performed data preprocessing and feature selection by exploring the data features and distributions within the dataset. Finally, experiments were conducted to measure and analyze the performance of my KNN class and proprietary functions.

The K nearest neighbors is a supervised machine learning algorithm that assumes similar things within a dataset exist in close proximity to each other in terms of their vectors. For my implementation of KNN, I had to write my own functions to calculate the distance between two vectors in two ways: Euclidean distance and Manhattan distance. These are needed as we want to test different approaches to calculating the distance between two vectors in our KNN algorithm experiments. Next, I had to develop functions to evaluate the performance of my KNN algorithm, these metrics included the accuracy, generalization error, precision, recall, and f1 score. Additionally, I developed a confusion matrix visualization, which shows how often the model confuses different classes together. Three curve functions were implemented: ROC, AUC, and Recall + Precision curves.

Next, the K nearest neighbors algorithm was implemented as its own Python class. Within this class, four functions are included: fit(), euclid_dist(), manhattan_dist(), and predict(). These are used in our experiment so that we can create different models with various parameter combinations. After this was set up, I explored the wine quality dataset and determined that features 'total sulfur dioxide' and 'density'. These two features showed a high correlation with multiple other features which indicated redundancy. This was determined by using a pair plot and heatmap to show the visual relationship between each feature.

To run my final experiment in Part C, I set up a grid search to train and test KNN models with various k values [1,5,9,11], distance methods (Euclidean and Manhattan), and weight options (uniform and distance). I chose to use a standardized X_train and set my inverse distance weighting to false. The standardized X_train and non-standardized X_train were compared in separate tests and showed no effect on the results as they were identical, so I wanted to keep them in just in case another set of parameters might trigger it to cause a change. The inverse distance weighting and regular distance weighting were compared in separate tests and both came with identical results as well, thus I decided to set it to false.

After running my KNN class and model on all combinations of parameters, I analyzed my results. I was shocked to see that all the experiments gave identical results despite the changes in k-values, distance metrics, and weights. The confusion matrices all show high true negative predictions and high false positives. There were 0 true positives and false negatives in all tests, which confused me as I felt like these results were completely false. I suspect there may be issues in my predict() method in my KNN class, but after going through, making changes, and rerunning experiments, I was not able to get the scores to change. I went back to the data and changed what features I dropped, but this only decreased my F1 and recall to 0.5 and 0.5. Before, they both had values of 1.0 which seemed wrong as an F1 of 1.0 which means that the model is perfect at identifying positive and negative scores, which is not the case at all, quite the opposite actually. The test accuracy for all tests was 65% and the precision was 0.5%, which indicates it was about half correct. We can verify this with the confusion matrix, where we see that the number of treu negatives that were identified is very high, however, no true positives were predicted. This explains why the score may look decent, but in reality, it is very poor.