Github: https://github.com/ablibranix/project4
Presentation: Presentation

Part I

Introduction

The k-Nearest Neighbor (KNN) algorithm is a widely used machine learning technique for classification and regression tasks. It relies on finding the k nearest data points to a given data point in a dataset and predicting its class or value based on the class or value of those nearest neighbors. We got our data from the UC Irvine Machine Learning Repository For this first portion of the project, we wanted to determine if we could accurately predict if a person has heart disease based on various attributes one could get at a general physical. With this data, we have the opportunity to possibly reduce heart failure in the population and also detect early heart health risks to prevent heart failure in that person's future.

Methods

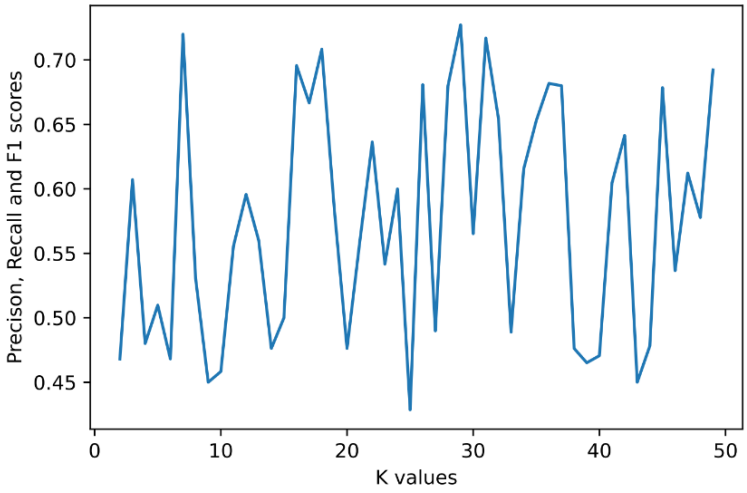
To determine the attributes to use in finding k nearest neighbor, we plotted the data for each attribute and compared the distributions between people with and without heart disease. This helped identify which attributes are most important for predicting the presence of heart disease. For example, we plotted age against heart disease status, we saw that people with heart disease tended to be older on average than those without heart disease. From conversations in class and looking at the graphs, we kept our attributes to 4.

To determine the value of k to use in the algorithm, we tested different values of k and compared the accuracy of the model for each value. We initially went with a smaller k value (10), but after then built a function which helped to check the scores of KNN for different numbers of k. When we did this, we realized our value of k was too small so we increased the value to 23 which seemed to be optimal for our dataset.

After determining our attributes and k value, we performed a cross-validation with those values 20 times. We printed out the precision, recall, F1 and support scores and then took the average of the 20 F1 scores to see if our attributes would work. We found the attributes to give reasonably high values so we kept them and continued to run more tests.

Results

From looking at the graphs from our data set, we decided to pick age, resting blood pressure (trestbps), maximum heart rate achieved (thalach), and cholesterol (chol). We picked these because they seemed to show a verifiable difference between persons who have heart disease and those who don’t. When determining the number of neighbors (k) to use, we graphed the values of k from 2 - 50 and noticed that 23 is a smaller number but would prove positive for testing.



When performing 20-fold cross-validation for k = 23, we received precision values, recall values, F1 values and support values.

	Precision	Recall	F1	Support
0	0.707317	0.674419	0.690476	43
1	0.536585	0.628571	0.578947	35
2	0.692308	0.9	0.782609	40
3	0.658537	0.627907	0.642857	43
4	0.825	0.673469	0.741573	49
5	0.75	0.75	0.75	44
6	0.588235	0.833333	0.689655	36
7	0.714286	0.625	0.666667	40
8	0.690476	0.74359	0.716049	39
9	0.695652	0.761905	0.727273	42
10	0.76087	0.729167	0.744681	48
11	0.764706	0.634146	0.693333	41
12	0.722222	0.604651	0.658228	43
13	0.674419	0.674419	0.674419	43
14	0.666667	0.820513	0.735632	39
15	0.804878	0.717391	0.758621	46
16	0.7	0.833333	0.76087	42
17	0.685185	0.822222	0.747475	45
18	0.7	0.622222	0.658824	45
19	0.815789	0.645833	0.72093	48

Our highest average F1 value from multiple iterations was 0.722380472189026. Although we could have gotten a higher value, we were low on time and kept with this value.

Part II

Introduction

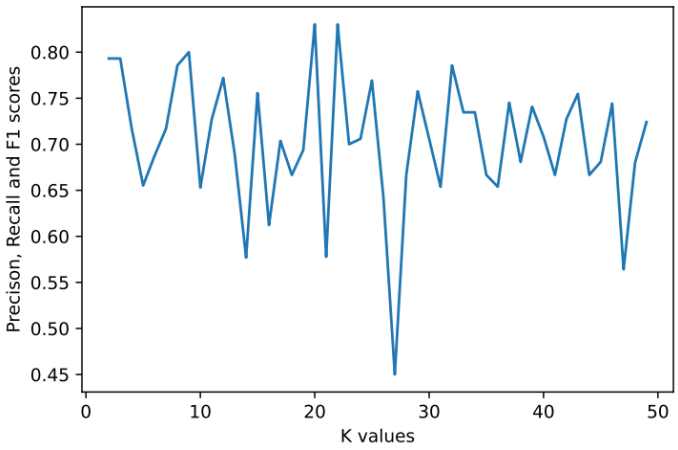
The k-Nearest Neighbor (KNN) algorithm is a widely used machine learning technique for classification and regression tasks. It relies on finding the k nearest data points to a given data point in a dataset and predicting its class or value based on the class or value of those nearest neighbors. For this second portion of the project, we wanted to determine if we could accurately predict the quality of a wine based on the chemical components of the wine. This algorithm can be used by winery owners who want to sell the best to their customers or distributors. It could possibly be used during the fermentation process to help determine when the wine is ready to be bottled.

Dataset

We gathered the data set from the UC Irvine Machine Learning Repository. We did not have to clean up the data very much since most of the data was numerical instead of information. To better separate the quality of wine, we changed the value in the 'quality' column from 1 - 10 to just 0 - 1. The wines which were of quality 1 - 5 were reassigned as a 0 and the wines with quality of 6 - 10 were reassigned as 1.

Results

From looking at the graphs from our data set, we decided to pick pH levels, alcohol level, and residual sugar amount. We picked these because they seemed to show a verifiable difference between the quality of wine being above or below 5. When determining the number of neighbors (k) to use, we graphed the values of k from 2 - 50. The graph had many spikes and dips but we picked a middle ground which was similar to our previous test. We did test numbers other than our initial pick but they did not give us good F1 scores for our cross-validation.



When performing 10-fold cross-validation for k = 25, we received precision values, recall values, F1 values and support values.

	Precison	Recall	F1	Support
0	0.6392694064	0.7446808511	0.687960688	188
1	0.6442307692	0.708994709	0.6750629723	189
2	0.6494845361	0.6923076923	0.670212766	182
3	0.7142857143	0.7105263158	0.7124010554	190
4	0.6956521739	0.7236180905	0.7093596059	199
5	0.6849315068	0.7978723404	0.7371007371	188
6	0.6911764706	0.7421052632	0.7157360406	190
7	0.6951871658	0.6806282723	0.6878306878	191
8	0.7046632124	0.7351351351	0.7195767196	185
9	0.6829268293	0.7070707071	0.6947890819	198

Our highest average F1 value from multiple iterations was 0.7010030355. We did not have much time to test other k values and did not see fit to test other attributes since they did not seem significant.