Project 4 Report

Note: all these scores are based on the time that we run them. It may be different at another run time (because we did not add a seed or set the random state!

Part 1:

Introduction

The primary goal of this analysis is creating a model named K-Nearest-Neighbors and find the best number of neighbors (k) and best features to predict whether an individual has heart disease based on several features provided in the dataset such as the individual's age, sex, resting blood pressure, cholesterol, maximum heart rate, etc. The stakeholders of our analysis are heart disease patients to have a quick simple check-up, doctors to help him/her diagnose the disease, and researchers to better understand the effects on the disease because an effective model for predicting heart disease can help professionals prevent and treat at-risk patients in time. First, we selected a subset of features that we normalized them with the greatest impact on whether an individual has heart disease based on the correlation plot, distribution plot, slope, and p-value of the T-test which they were: 'thalach'[1], 'exang'[2], 'oldpeak'[3], 'ca'[4], 'thal'[5]. Then, we used these features to create the model with training dataset for k in the range 1 to 50 with 10-fold[6] cross-validation. In each k we tested the models with the test dataset and then calculated the mean of f-scores for the class label if disease exists and then compared them to find the best k. Repeating this process led us to find the best k as 32. We used this k to train our model and obtained average values of 10-fold cross-validation, recall: 0.759, precision: 0.844, f-score: 0.795, and accuracy: 0.827. And without cross-validation (20% testing dataset, 80% training dataset): precision: 0.913, recall: 0.777, f-score: 0.84, and accuracy: 0.866. Our surprising results can be found here. Also, you may see our Github repository.

Methods

When selecting what features to use in our K-Nearest-Neighbors model, the first time, we simply decided to select the 4 or 5 features that have the greatest correlation with heart disease in the dataset. To determine which of the features have the greatest correlation with heart disease, we plotted and calculated the slopes each of the features against whether the disease was found in the individual (0 for no disease, 1 for disease). Before plotting, we made sure to standardize the data so that each of the features is weighted equally. Then we selected the top five features with the highest slopes for finding the best k for both models with 4 features or 5 features. To support this purpose, we also plotted distributions for each feature and calculated T-test to confirm they are different, one distribution for those with the disease and one distribution for those without for each feature. We then compared the two distributions against each other for each feature and verified that the features we selected had drastically different distributions for those with the disease and those without the disease.

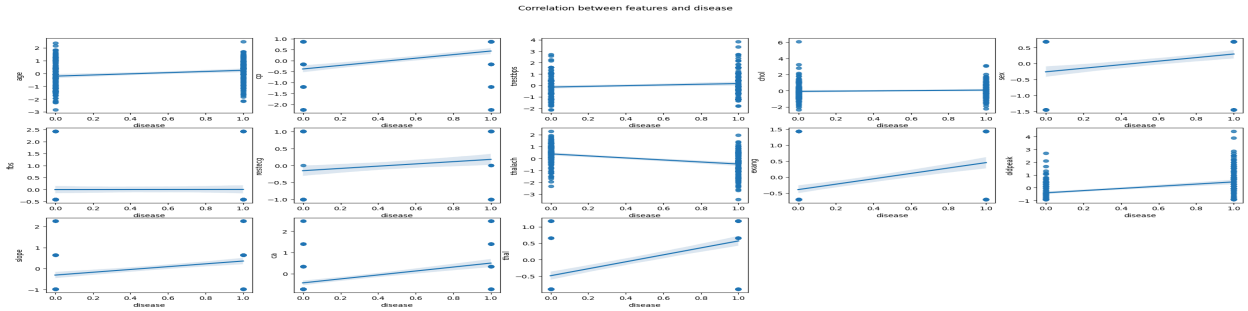
To determine the best value of k to use for our K-Nearest-Neighbors model, we decided to use 10-fold cross-validation. First, we defined a function to return the average f-score (for the class label if disease exits) of the given k value of the 10 models trained using 10-fold cross-validation. Then, for k values ranging from 1 to 50, we used this function to find an average f-score for each k value from 1 to 50 for the label "1" which is the representative of the disease. The best k value is just the k value with the highest average f-score. This

[1] Maximum heart rate achieved
[2] Exercise induced angina
[3] ST depression induced by exercise relative to rest
[4] Number of major vessels (0-3) colored by flourosopy
[5] A blood disorder called thalassemia
[6] K-fold cross-validation is a technique for evaluating predictive models. The dataset is divided into k subsets or folds.

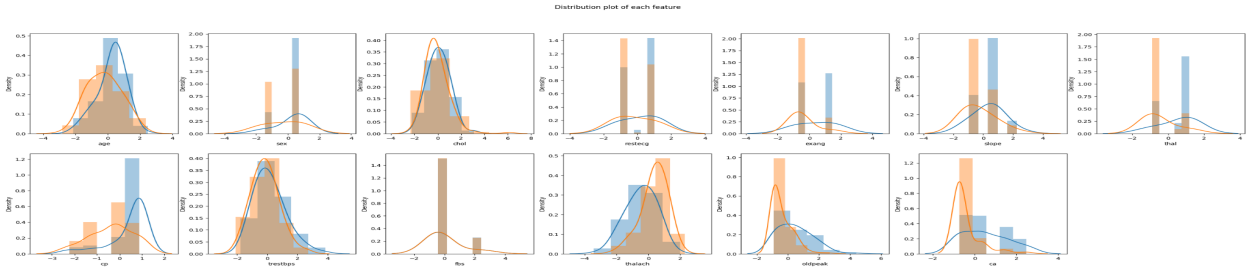
process is repeated 1000 times. An iterative version of this process is done with 10 iterations in the Jupyter Notebook[7] because multiprocessing[8] doesn't work well, but we have implemented finding the best value for k in another Python file which parallelizes this process. The average best k value is computed; 32 is our best k value.

Results

The figure below shows the plots of all the features against whether the individual has heart disease:

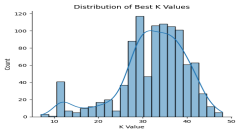


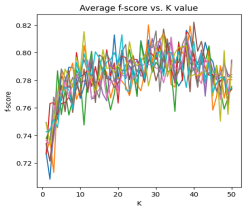
The figure above shows that the features with slopes: "thalach": -0.211, "exang": 0.210, "oldpeak": 0.211, "ca": 0.231, and "thal": 0.262, have the strongest correlation with whether an individual has heart disease or not. Therefore, these five features will be the input features supplied to the K-Nearest-Neighbors model. To confirm this, we have included a distribution plot and the T-test, which is shown below:



The p-values from the T-test confirm significant differences in feature distributions between healthy and sick individuals. Specifically, for "thalach", "exang", "oldpeak", "ca", and "thal", the p-values are 2.23e-14, 3.27e-14, 2.15e-14, 3.35e-17, and 1.39e-22, respectively, with corresponding T-statistics of -8.03, 7.97, 8.04, 8.97, and 10.64. These results validate the distinction between feature groups and support their suitability for feature selection.

The following figures show the average best f-score we got from cross-validation with different k in each iteration and distributions of the best number assigned to k. First, the best k here is 40 without finding the mean of them with 10 iterations! The next plot is the distribution of the best k values when the 10-fold cross-validation is run on the data 1000 times. The distribution plot tells us that our best k value typically sits between 25 and 40. For our best k values, we will take the average best k value which is 32. Using 32 as our k value and the features "thalach", "exang", "oldpeak", "ca", and "thal", we obtained [0.687, 0.812, 0.846, 0.833, 0.642, 0.8, 0.909, 0.846, 0.75, 0.846] with an average 0.79 for recall, [0.916, 0.928, 0.916, 0.769, 0.818, 0.888, 0.909, 0.846,





[7] Jupyter Notebook is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages. It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger.
[8] Multiprocessing is the use of two or more central processing units within a single computer system.

0.857, 0.846] with an average of 0.869 for precision, [0.78, 0.866, 0.879, 0.8, 0.72, 0.842, 0.909, 0.846, 0.799, 0.846] with an average f-score of 0.829, and [0.793, 0.862, 0.896, 0.827, 0.758, 0.896, 0.931, 0.862, 0.793, 0.862] with an average accuracy of 0.848 in our 10-fold cross-validation. Also, just for a single run without cross-validation (20% test dataset, 80% training dataset) of the split test and train data is precision: 0.904, recall: 0.791, f-score: 0.844, and accuracy: 0.883. Also, we trained the model without "exang" which the result was most of the time worse than with "exang": [0.7, 0.714, 0.823, 0.875, 1.0, 0.823, 0.571, 0.9, 0.733, 0.75] with average 0.789 for recall, [1.0, 0.909, 0.933, 0.538, 0.705, 0.933, 0.8, 0.75, 0.846, 0.5] with an average of 0.791 for precision, [0.823, 0.8, 0.874, 0.666, 0.827, 0.874, 0.666, 0.818, 0.785, 0.6] with an average of 0.773 for f-score, and [0.79, 0.827, 0.862, 0.758, 0.827, 0.862, 0.724, 0.862, 0.793, 0.724] with an average accuracy of 0.800 in our 10-fold cross-validation and without cross-validation (20% testing dataset, 80% training dataset): precision: 0.875, recall: 0.75, f-score: 0.807, and accuracy: 0.833.

Part 2:

Introduction

For our second analysis, our goal is to create a model named K-Nearest-Neighbors for a breast cancer dataset which supplied with some features such as age, menopause, tumor size, involved nodes[9], metastatic, etc, can predict whether that tumor is benign or malignant. The stakeholders of this analysis are breast cancer researchers to have a better undersatnding of the affect of each feature on benign or malignant cancers, doctors to help them to have a better dignosis based on breast test results, and cancer patients to have a quick simple check-up if they have the breast test results, because an effective model for predicting type of cancer can help save lives and further research. First, we selected a subset of features that we normalized them with the greatest impact on whether an individual has malignant breast cancer based on the correlation plot and slopes, distribution plot, and p-values of the T-test. Then, we used the subset of the features (which is: 'Menopause'[10], 'Metastasis'[11], 'Age'[12], 'Tumor Size (cm)', 'Inv-Nodes'[13]) to create the model with training dataset for k in range 1 to 50 with 10-fold cross-validation. In each k we tested the models with the test dataset and then calculated the mean of f-scores for class label if cancer is malignant and then compared them to find the best k. Repeating this process led us to find the best k as 10. We used this k to train our model and obtained average values of 10-fold cross-validation, recall: 0.817, precision: 0.988, f-score: 0.891, and accuracy: 0.909. And without cross-validation (20% testing dataset, 80% training dataset): precision: 0.992, recall: 0.875, f-score: 0.933, and accuracy: 0.926.

Dataset

For our dataset, we chose a breast cancer dataset[14]. The breast cancer dataset contains several categorical and binary variables, which for plotting purposes needed to be converted to numbers. Specifically, the "Breast", "Breast Quadrant", and "Diagnosis Result" columns of the dataset needed to be adjusted. To do this, we simply mapped the possible values of each column to integers. It should be mentioned that we dropped the rows with missing values, omitted the extra space in each string, and deleted the "S/N"[15] and "Year"[16] columns because they have no effect on our analysis. For the feature "Breast Quadrant" since we had 4 different strings named:

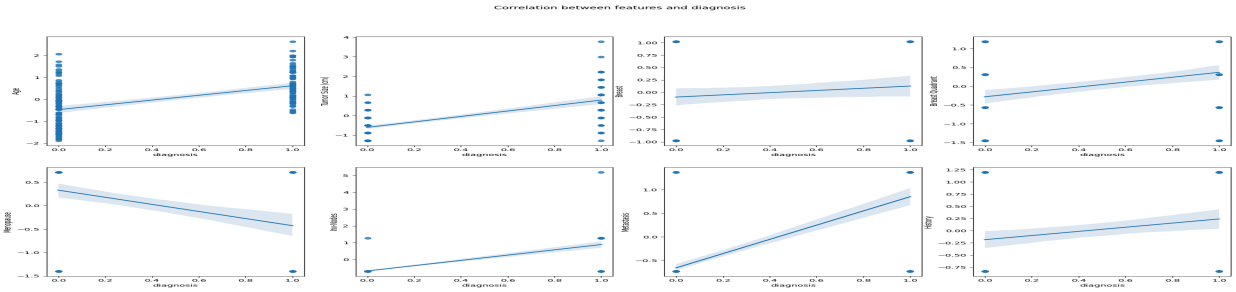
---

[9] The number of axillary lymph nodes that contain metastatic
[10] Whether the patient is pro or postmenopausal at the time diagnose
[11] If the cancer has spread to another part of the body or organ.
[12] Age of patient at the time of diagnose
[13] Involved nodes
[14] https://www.kaggle.com/datasets/fatemehmehrparvar/breast-cancer-prediction
[15] Unique identification for each patient.
[16] The year diagnosis was conducted.

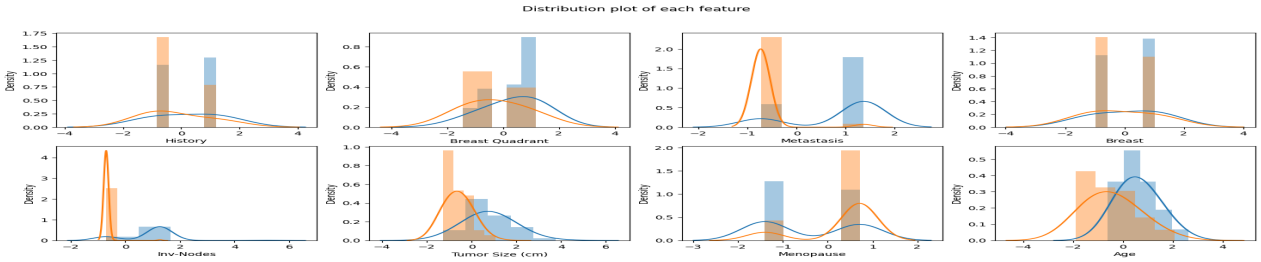
“Lower inner”, “Lower outer”, “Upper inner”, and “Upper outer” we changed them to 0, 1, 2, and 3. Also, the name of the column “Diagnosis Result” changed to “diagnosis” and the rows with the value “#” are filtered. The other features of this data set are Menopause, Metastasis, Age, Tumor Size (cm), Inv-Nodes (these features played an important role), and History.

Results

The figure below shows the correlation between each feature and whether the tumor is malignant.



The figure suggests that the features 'menopause', 'metastasis', 'age', 'tumor size', and 'inv-nodes' are most strongly correlated with whether a patient has malignant breast cancer, with respective slopes of -0.186, 0.371, 0.269, 0.343, and 0.387. Therefore, these are the features we choose for our K-Nearest-Neighbors model. This choice is supported by the figure and the p-value of the T-test below, which shows a significant difference in the distributions for the selected features in malignancy and benign individuals.



The T-test results indicate significant differences in feature distributions: "Age": (p=4.96e-17, t=9.18), "Menopause": (p=2.81e-08, t=-5.77), "Tumor Size": (p=1.57e-30, t=13.65), "Inv-Nodes": (p=3.16e-43, t=17.76), and "Metastasis": (p=4.89e-38, t=16.07).

The following figures show the average best f-score we got from cross-validation with different k in each iterations and the distribution of best k values. The best k here is 46 without finding the mode with 10 iterations! The distribution of best k values obtained by running a 10-fold cross-validation 1000 times. Because there is one bin that dominates the others in the histogram, The most suitable value to choose for best k is the mode, which is 10 in this case. Using 10 for our k value and features “Menopause”, “Metastasis”, “Age”, “Tumor Size”, and “Inv-Nodes”, we obtain [0.833, 1.0, 0.8, 1, 0.8, 0.714, 0.571, 0.75, 0.7, 1] with average values of 0.816 for recall, [1.0, 1.0, 1.0, 0.857, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0] with an average of 0.985 for precision, [0.909, 1.0, 0.888, 0.923, 0.888, 0.833, 0.727, 0.857, 0.823, 1.0] with an average of 0.885 for f-score, and [0.9, 1.0, 0.9, 0.95, 0.95, 0.9, 0.7, 0.9, 0.85, 1.0] with and average of 0.905 for accuracy. And without cross-validation (20% testing dataset, 80% training dataset): precision: 0.928, recall: 0.684, f-score: 0.787, and accuracy: 0.829.