Ann Marie Humble and Ehsan Kahrizi

Part I.

Introduction

The aim of this analysis is to develop a K-Nearest Neighbors model with optimum values of K and features from the dataset to forecast heart disease given several people metrics such as order of age, order of sex, resting blood pressure, etc. The patients are the quick check-up heart disease patients, while the doctors are the people wanting to get the right results, and the researchers are the ones studying how each element contributes.

Features are taken into account using their correlation, distribution, slope, and p-values of the t-test. Out of these, it has been noted that “thalach”[1],“exang”[2], “oldpeak”[3], “ca”[4] , and “thal”[5] are the most impactful factors. The model is subjected to training using the above features and is thereafter evaluated using a 10-fold cross-validation method which is then used to identify the optimal number of neighbors (k) from 1 to 50. This study determined the best k value to be 32, and it showed very good outcomes in terms of performance measures. Some of these metrics included precision, recall, F-score, and accuracy. Our results were surprising. When the model was validated using cross-validation, it got a recall rate of 0.759, precision of 0.844, F score of 0.795, and accuracy of 0.827. In the case of no cross-validation, the level of precision is 0.913, and the recall level is 0.777, with the F-score giving 0.84 and an accuracy of 0.866. This particular analysis bears considerable promise in providing means of prevention and therapeutic measures of active preventive measures for people at risk. Notably, all these score results are based on the time we run the model. It may be different at another run time (because we did not add a seed or set the random state!)

Methods

At the time of developing the K-Nearest-Neighbors model, we first picked the features most highly correlated with the existence of heart disease (the label ‘1’ class), first obtaining their correlation plot against disease presence and finding the slope of them after standardizing the datasets for equal weighting. Then, the top four and five features with the highest slopes were then chosen. Their distribution plots and t-test analyses (p-value) confirmed such a significant difference and were selected as such visually (by inspection of results and numbers).

Searching for the optimal k value, we used 10-fold cross-validation (each time, the dataset will be shuffled randomly to get different results) and averaged the f-scores for k in 1 to 50.

The k value at which the f-score for the disease is maximum (label for the presence of the disease is "1") was considered. This step was iterated 1000 times and resulted in the value 32 as the best k value. Because of multiprocessing limitations, an iterative procedure was applied in the Jupyter Notebook, where parallel processing was performed in a different Python file.

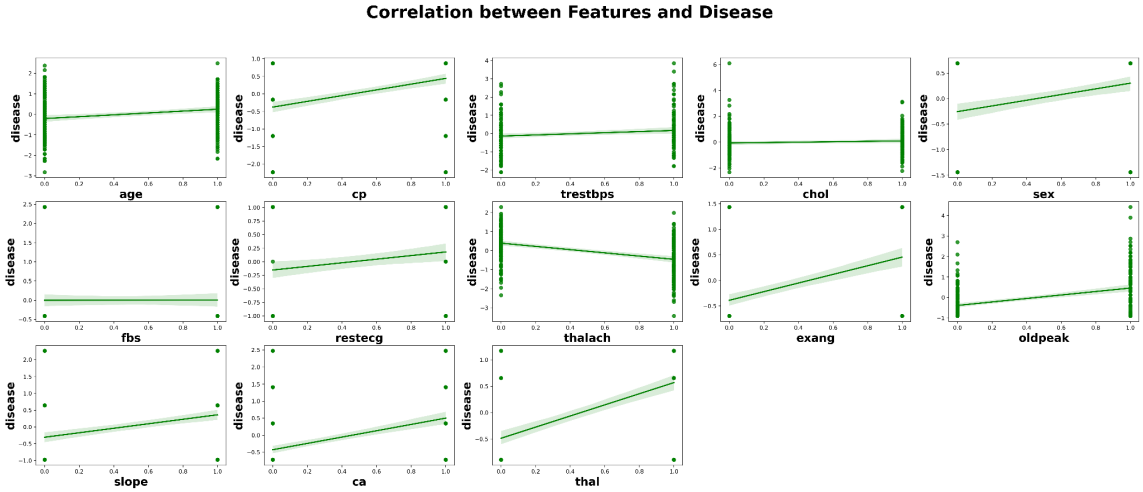
Results

These plots indicate all the features against whether the individual has heart disease. As we can see, the features with slopes: “thalach” = -0.211, “exang” = 0.210, “oldpeak” = 0.211, “ca” =

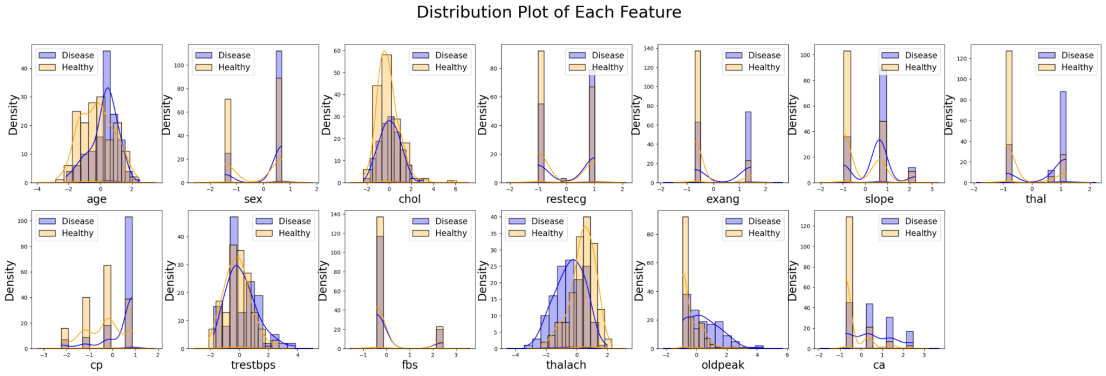
---

[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

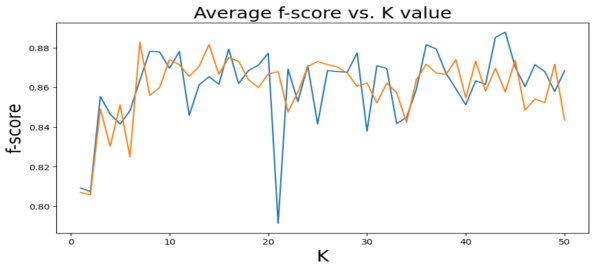
0.231, and “thal” = 0.262 have the highest correlation with whether an individual has heart disease or not. So, we used these as our input for features for the K Nearest-Neighbors model.



For validation of our results in the previous plots, we have also shown the distribution plot and the T-test in the below figure. The p-values from the T-test show a significant difference in feature distributions between healthy and sick cases. For example, in "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 is -8.03, 7.97, 8.04, 8.97, and 10.64. So, these results confirm the differences between feature groups and support their suitability for feature selection.

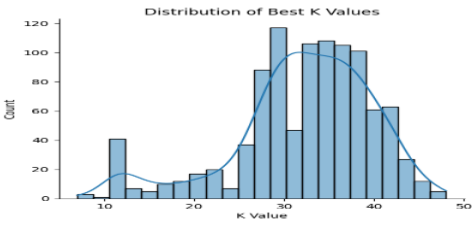


This figure indicates the average best f-score from cross-validation with different k in each iteration. The figure shows the best k here is 44 without finding their mean (or mode) with 2 iterations!



In the last plot, the distribution of the best k values when the 10-fold cross-validation is run on the data 1000 times showed. 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”, result were (0.687, 0.812, 0.846, 0.833, 0.642, 0.8, 0.909, 0.846, 0.75, 0.846) with an average of 0.79

for recall, (0.916, 0.928, 0.916, 0.769, 0.818, 0.888, 0.909, 0.846, 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 considering cross-validation - with assumptions random: 20% test dataset, 80% training dataset- of the split test and train results was: precision about 0.904, recall about 0.791, f-score about 0.844, and accuracy about 0.883. Further, we trained the model without “exang”. The result showed that most of the time, it was worse than the case when considering “exang” in which were (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 (random: 20% testing dataset, 80% training dataset): precision: 0.875, recall: 0.75, f score: 0.807, and accuracy: 0.833.



Part II.

Introduction

Universities are concerned with student dropout rates for a host of reasons. Aside from indicating the wellbeing of a university’s students, dropout rates reflect on the quality and accessibility of the university. A high dropout rate may discourage student enrollment or even affect the university’s funding. It would be useful for universities to know some of the factors that predict student dropout. With this information, they could provide support to students that are likely to drop out.

Dataset

For this analysis, we used the dataset, “Predict Students' Dropout and Academic Success” from the UC Irvine Machine Learning Repository. The dataset is a .csv file of information on students and their academic, economic, and familial attributes. When loading this file into a dataframe, we used semicolons as the delimiter. Then we erased all whitespace from the ends of the column headers. After that, we dropped all information in the dataset belonging to students marked as currently enrolled, since they have not graduated or dropped out yet. After that, we created new columns to hold the standardized values of the columns we would be using in our analysis.

Methods

We did our first round of testing with a fixed sampling (random_state=42 in our train_test_split function). For our second round of testing, we ran the model through Monte-Carlo cross-validation (without fixed sampling). Within these different testing methods, we used trial and error to determine the optimal k and the most useful variables for our model. These two methods of testing gave us very different results. This is likely because the first method told us only what would be optimal for a very specific subset of the data, while the second method gave us results generalized to the whole database. It is worth noting that the results we got from the

second method of testing gave us a much higher Average F1-Score. Doing this again, we would drop the first round of testing entirely.

Results

After our second round of testing, we determined the optimal k value to be 3. The optimal variables surprised us. We split a sample of the dataset's numeric variables into two groups: national economic data (inflation rate, unemployment rate), and student data (age at enrollment, previous qualification (grade), curricular units 2nd semester (grade), curricular units 2nd semester (enrolled)). We ran the model a number of times for the variables relating to the economy alone, the variables relating to student data alone, and both these sets of variables together. The resulting averages are in the table below. We were surprised by how similar these averages were for each group. These findings suggest that economic factors (such as the unemployment and inflation rate) have the same effect on dropout rates as a student's age, credit count, and grades. Universities interested in lowering their dropout rates can use this information by providing students with more support during periods of economic instability. They can also give extra support to students who do not have the typical age, credit amount, or grades of the average college student.

	Average Precision	Average Recall	Average F1-Score
National Economic Data	40%	30%	33%
Student Data	39%	32%	35%
All	40%	32%	36%