Predicting an Individual's Risk of Heart Disease

Authors: Phuong Vu, Yunwei Liang, Trinh Nguyen

Summary of Research Questions

1. Which machine learning algorithm can best predict the risk of heart disease?

   We will compare the accuracy of different supervised machine learning algorithms for predicting the risk of heart disease and find the algorithm with the highest accuracy. We will be using the Decision Tree Classifier, the Random Forest Classifier, and the Gaussian Naive Bayes Classifier. We will be including all patients' characteristics in the features of the machine learning model. Our models will predict the risk of a patient having heart disease. We will compare our models with the accuracy score of each model.

2. Which certain sets of features are better indicators of the high risk of heart disease than the other sets of features?

   We will implement the most optimal algorithm found in Part 1 of our research for Part 2. In Part 2, we are trying to find the impact specific set of features have on predicting the risk of heart disease. We will compute the correlation of different sets of features and risk of heart disease. We will examine each trial's accuracy, and find the set of features with the highest accuracy.

Motivation and Background:

Through the research article "The accuracy of prediction of heart disease risk based on Machine Learning Classification Techniques", we are inspired to find the correlation between patients' characteristics and their risk of heart disease. The researchers in research article tested the accuracy of different Machine Learning algorithms when predicting the risk of heart disease of a participant. We will use their dataset of participants' information to make a simplified version of their investigation. We will test and compare the accuracy of different learning machine models based on their accuracy score. We will then use the most accurate model to determine how specific set of features will affect the model's prediction of health risk. For future exploration/expansion, we could develop machine learning models most fitting for common medical diagnosis to reduce the rate at which people are being turned away from hospitals during a crisis. This

advance could also potentially lower medical costs, which is one of the biggest factors in one's decision to even seek out a doctor.

Dataset:

Link to the dataset: Heart Disease Data Set

We will use the dataset in "cleve.mod" file in the Data Folder in the attached link. This dataset was collected from Cleveland Clinic Foundation and accessed in the UCI Machine Learning Repository website. There are 14 attributes describing each of 303 studies in the dataset. It shows the relationship among people's different traits like sex, chest pain type, resting blood pressure, etc… and their risks of having heart disease. For our machine learning models, we will use the column "class att" as the label. Note that "class att" indicates a healthy individual with "H" and at-risk individual with "S1", "S2", "S3", and "S4".

Methodology:

1. Determine the best algorithm
    a. Transform data from a mod file into an accessible format. Filter missing data out of the set. One-hot encoding the dataset if necessary.
    b. Select all columns as features except "class att"--this is the health risk--to be the model's label. Include all attributes as the features.
    c. **Set up and train a Decision Tree Classifier Model**
        i. Follow the steps from lecture:
            1. Unpack train_test_split into training and testing sets with a 8:2 ratio relatively.
            2. Use DecisionTreeClassifier to build the model.
            3. Fit the model with the training set.
        ii. Test the model with the test set and record the accuracy score.
    d. **Set up and train a Random Forest Classifier Model**
        i. Import numpy and convert the label from the dataframe to a Numpy array. Remove the identifying names for the features from the dataframe, then turn the features into a Numpy array.
        ii. Splitting the data into 80% training set and 20% testing set using `train_test_split()` from sklearn's model selection.
        iii. Train and fit the model with training data using `model.fit()`
        iv. Record the accuracy score.
        v. Follow the link for a more informative method: https://towardsdatascience.com/random-forest-in-python-24d0893d51c0

e. **Set up and train a Gaussian Naive Bayes Model**
    i. Getting the dummy encoding for the features using `get_dummies()` from pandas.
    ii. Splitting the data into 80% training set and 20% testing set using `train_test_split()` from Skicit-learn's model selection. Establish a baseline for the model so that it can optimize its classification approach.
    iii. Create a Gaussian Naive Bayes Model using `GaussianNB()` from the Skicit-learn's `naive_bayes`.
    iv. Train and fit the model with training data using `model.fit()`
    v. Test and record the accuracy with `accuracy_score()` using the testing set.
    vi. Follow the link for a more informative method: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

2. Using the best algorithm to determine the best set of features
    a. Use the prepared data from Part 1 and the most accurate machine learning model(will be determined by).
    b. Using matplotlib, create a bar plot of the feature importances. This will help us visualize the impact of each feature on the model's prediction. Based on the results, come up with different combinations of features from the dataset.
    c. Use the relative instruction from Part 1 for the most accurate method to test the different sets of features.
    d. Calculate and record each accuracy score.
    e. Plot a scatter chart based on the model's prediction to compare the predicted values to the actual values.

Work Plan:

1. Set up the starter files using Jupyter Notebook and share access
    a. Starter files include main.py, cleveland.mod (finish by February 28th)
    b. Break down the main.py into two main parts to address two main research questions
2. Responsibilities
    a. All contributors in meeting: first machine learning model in part 1 finish by March 2nd
        - Finish transforming data into accessible format and filtering the data
        - Split out the feature and label columns
        - Set up and train a Decision Tree Classifier Model
        - Record the accuracy score
    b. Trinh: Part 1

- Finish setting up the Random Forest Classifier model by March 3rd
- Finish splitting the data and training the model by March 5th
- Finish recording the accuracy score of the model by March 6th

c. Yunwei: Part 1
- Finish setting up the Gaussian Naive Bayes Classifier model by March 4th
- Finish splitting the data and training the model by March 6th
- Finish recording the accuracy score of the model by March 7th

d. Phuong: Part 2
- Finish making bar plot of the feature importances and prediction by March 7th
- Finish testing different sets of features using the result of part 1 by March 9th
- Finish calculate, record accuracy scores and make chart to compare the predicted and actual values by March 10th

3. Time estimates
   a. Finish writing all the code for transforming data, plotting graphs and building machine learning model by March 10th
   b. Finish writing report of the result by March 13th
   c. Finish preparing materials and practicing for in-class presentation by March 19th

Questions(optional):

How can we convert the .mod or .data file into .csv one?

Link to the original Research Paper:

https://www.sciencedirect.com/science/article/pii/S235291481830217X#bib22