CSC111 Project 2 Proposal:
Saving Lives: Predicting Cardiovascular Disease

Sudharshan Palaniyappan, Thanush Lingeswaran, Jackie Liang, Divnoor Chatha

March 6, 2024

Problem Description and Research Question

How can we predict someone's chances of having cardiovascular disease? Consequently, how can we recommend future steps and goals accordingly? The idea we are implementing is predicting cardiovascular disease using multiple data points given by the user. The user should know a few important pieces of information about themselves. For example, their age, weight, cholesterol levels, and if they smoke or not etc. The reason why we decided to choose this topic is because of the fact that heart disease is the leading cause of death in men, women and many racial and ethnic groups (CDC). It's so dangerous that the centre of disease control and prevention states how "'One person dies every 33 seconds in the United States from cardiovascular disease'" (CDC). We want to help individuals by using our program to predict their chances of having heart disease by comparing with patients who have been diagnosed by a professional, in order to potentially save their lives. We know this is important, because early diagnosis is extremely important to help control the disease, as "Four out of five people don't know they have heart failure until their symptoms are severe enough to put them in the emergency room" (Booth). Early detection is the difference between life or death as late detection might not be maintainable, and the patient's quality of life will drop massively. This is why we need to focus on this issue, as early detection of the disease will save lives. Overall, the importance of this issue is what strongly motivated us to choose such a project, in order to help detect these issues earlier on, and recommend steps accordingly.

Computational Plan

We plan to create the prediction model using machine learning. Starting by filtering the dataset, we can get rid of any outliers, missing values, or invalid data. In addition, we can get rid of any 'noise' in the data, in other words, the meaningless information which we do not require to compute our desired outcome. This dataset quality control can be performed using methods from the pandas library in Python. The pandas library is used for data wrangling and analysis, and we can use methods such as `read_csv()`, `dropna()`, and `drop_duplicates()`.

Next, we must identify which variables (categorical or quantitative) have a high correlation to cardio-vascular disease. To do this, we can perform linear regression analysis to get correlation coefficients between certain variables. In addition to linear regression, we can visualise the relationship between the covariates (the independent variables for prediction) using confusion matrices to further analyse and note any strong relationships. From the list of coefficients we obtain and the relationships visualised via confusion matrices, we can choose the variables which seem to have the greatest impact on cardio-vascular disease.

Moving forward, to build the model itself, we can use the scikit-learn library in python, which allows us to build the initial decision tree. The scikit-learn library is an open source machine learning library. In order to build the tree, we must split the dataset into training and testing and use only the training data to build the tree. To measure accuracy of the model, we use the model and run it on the testing data to measure how it performs against data which it was not trained on using an accuracy score function. We can repeat this process indefinitely, tweaking small aspects of the model, such as the number and type of variables we use until we obtain a desirable accuracy score. The classification tree we obtain as a result of this process will be the primary aspect of the project that will use trees. As the tree is essentially the backbone of the model, we hope this is enough incorporation of trees in our project.

After building the model itself, we must also account for multicollinearity, which occurs when 2 independent variables

are too strongly correlated. This can cause our model to be less reliable as the effects of the variables are confounding, implying they may have no real correlation to cardio-vascular disease. To address this issue, we can analyse the condition number of our model, which indicates whether or not our model is in the presence of multicollinearity. In addition, we could also use backward selection to perform the same task but much less rigorously.

The final step of our plan is to create an interactive user experience. This can be done with the user inputting the required biomarkers and feeding it through our model to obtain an accurate conclusion. This conclusion can be in the form of the percentage chance of the user obtaining heart disease or simply the probability generated by our model. We can also recommend future goals and steps which will need to be taken, in correspondence to their chances of having cardiovascular disease. Both of the above aspects will be displayed on the graphical user interface. Finally, we may additionally add a final graph, which compares the health statistics of the user in comparison to the ideal healthy human at the user's age. Modifications to the graph can be incorporated to include colour indication of individual health aspects (green for healthy statistic, red for worrying statistic). We will be using tkinter to help create a graphical user interface (GUI) for the project. The GUI will be used to help receive input from the user for questions regarding their health, and will boost the overall aesthetic of the program. We will be using the base aspects of tkinter which includes creating window and widget classes, with examples including the "entry" and "text" widget classes which allow for text entry of differing lengths from the user. Additionally, we will be using plotly to help create a graph which compares the health statistics/data points of the user to the ideal healthy human at the corresponding age of the user. This graph would be the ideal visualisation for the user, allowing for them to easily understand their results.

The dataset we will use from kaggle consists of 70, 000 entries of patient data we will use to create our model. Some of the biomarkers contained within this dataset include age, height, weight, cholesterol level, and alcohol intake. This is a sample of what 1 row from the dataset looks like:

id	**18393**
age(in days)	168
gender	2
height	156 cm
weight	85.0 kg
ap_hi	140
ap_lo	90
cholesterol	3
gluc	1
smoke	0

Note: Many of the concepts we require in our steps, such as dealing with linear regression, confusion matrices, and filtering data using pandas dataframes were taught in STA130. 2/4 group members have taken this course last semester and we will make sure everyone understands each step we take to build our model to ensure the workload is fair.

Works Cited

[1] Amos, David. "Building Your First Python GUI Application with Tkinter." Realpython.com, realpython.com/python-gui-tkinter/building-your-first-python-gui-application-with-tkinter.

[2] Booth, Stephanie. "Early Heart Failure Diagnosis Is Key." WebMD, 2023, www.webmd.com/heart-disease/heart-failure/early-diagnosis-heart-failure.

[3] CDC. "Heart Disease Facts." Centers for Disease Control and Prevention, 15 May 2023, www.cdc.gov/heartdisease/facts.htm: :text=Heart

[4] Pandas. "Python Data Analysis Library." Pydata.org, 2018, pandas.pydata.org/.

[5] Plotly. "Plotly Express in Python." Plotly.com, plotly.com/python/plotly-express/.

[6] scikit-learn. “Scikit-Learn: Machine Learning in Python.” Scikit-Learn.org, 2019, scikit-learn.org/stable/.

[7] Ulianova, Svetlana, et al. “Cardiovascular Disease Dataset.” Www.kaggle.com, 2019, www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.