Heart Disease Prediction

Zeinab Alazzawi & Hussain Alhassan

[Zalazzawi1@student.gsu.edu](mailto:Zalazzawi1@student.gsu.edu) & [Halhassan2@student.gsu.edu](mailto:Halhassan2@student.gsu.edu)

Data Set: <https://www.kaggle.com/fedesoriano/heart-failure-prediction>

Code: <https://colab.research.google.com/drive/1Mvin4iU3GHTOwlizLc5jdNyAaH9SDKXi#scrollTo=q805dfF6bjg_>

**Problem Statement -** How to predict who has an increased chance of developing a heart disease based on previous patients’ data.

**Data Sources –** We used Kaggle to get our data (<https://www.kaggle.com/fedesoriano/heart-failure-prediction>) and we primarily used Age, Sex, ChestPainType, RestingBP, and HeartDisease columns.

**Data Preparation –** We dropped all the null columns and converted our strings into numeric fields. To create our linear regression model we converted our strings into new numeric fields such as chest pain type: ASY is now 1. As well as Sex, M is 1 and F is 2. Surprisingly our data had no null values, it was already cleaned up from Kaggle, but we had too many columns we weren’t going to use in our data so we removed RestingECG, Cholesterol , FastingBS, MaxHR, ExerciseAngina, Oldpeak, ST\_Slope , Unnamed: 12. We weren’t sure what Unnamed:12 was but it kept coming up into the data, it wasn’t in our original csv. For the visualization part, we wanted to show the patients who have heart disease and their demographic, so we focused the heart data based on heart disease and if its value was 1.

**Modeling -**  Here are the methods we used

* Getting all data where Heart Disease is True.

Input:

Heart\_Disease = heartdata[heartdata['HeartDisease']==1]

Heart\_Disease

Output:

Table

Description automatically generated

* Plotting Age vs Blood Pressure

Input:

Heart\_Disease.plot(x='RestingBP',y='Age',kind = 'scatter', c='red')

plt.title('Age vs Resting Blood Pressure')

plt.show()

Output:

Chart, scatter chart

Description automatically generated

* Plotting relationship between Age and Gender. We used .plot to plot our scatterplot

Input:

Heart\_Disease.plot(x='Age',y='Sex',kind = 'scatter', c='red')

plt.title('Age vs Gender')

plt.show()

Output:

Chart

Description automatically generated with low confidence

* Plotting Chest Pain Types by Age. We used .plot to plot our scatterplot

Input: Heart\_Disease.plot(x='ChestPainType',y='Age',kind = 'scatter', c='red')

plt.title('Age vs Chest Pain Type')

plt.show()

Output:

Chart, scatter chart

Description automatically generated

* Replacing Sex string into numeric values. Used the inplace tool

Input:

heartdata['Sex'].replace('M', 1 ,inplace=True)

heartdata['Sex'].replace('F', 2 ,inplace=True)

heartdata['Sex']

Output:

A picture containing table

Description automatically generated

* Replacing strings in ChestPainType into numeric values. Used the inplace tool

Input:

heartdata['ChestPainType'].replace('ASY', 1 ,inplace=True)

heartdata['ChestPainType'].replace('ATA', 2 ,inplace=True)

heartdata['ChestPainType'].replace('NAP', 3 ,inplace=True)

heartdata['ChestPainType'].replace('TA', 4 ,inplace=True)

heartdata['ChestPainType']

Output:

Graphical user interface, text

Description automatically generated

* Partitioning our data. Used iloc, drop, and array

Input:

X = heartdata.iloc[:, :1].values

y = heartdata.iloc[:, -1].values

#Split data into X and y

Train\_X = np.array(heartdata.drop(['Age','Sex','HeartDisease'],1))

Train\_y = np.array(heartdata[['HeartDisease']])

print(Train\_X)

print(Train\_y)

Output:

A picture containing background pattern

Description automatically generated

* Predicting the Data. Used reg.predict, asarray, and reshape.

Input:

prediction = reg.predict(np.asarray([2,11]).reshape(1, -1))

print('Prediction:', prediction)

Output:

A picture containing logo

Description automatically generated

* Test Data. Used iloc, drop, and array

Input:

X = df.iloc[:, :1].values

y = df.iloc[:, -1].values

Test\_X = np.array(df.drop(['Age','Sex','HeartDisease'],1))

Test\_y = np.array(df[['HeartDisease']])

print(Test\_X)

print(Test\_y)

Output:

Table

Description automatically generated

* Get Accuracy- we used sklearn.metrics’s mean squared error, same one from individual assignment 3

Input:

from sklearn.metrics import mean\_squared\_error  
mean\_squared\_error(Test\_y\_predicted, Test\_y)

Output:



**Evaluation –** The steps we used to evaluate our data are:

1. **Cleaning the data**
   1. Removed all the null values from the data set to avoid any errors. We placed age column into age bins to visualize it as a chart.
2. **Changing the strings into numeric**
   1. Changed strings into numeric data, such as Sex, Age, Chest Pain Type, and Heart Disease so that way there are no issues with linear regression.
3. **Plot the relationships** 
   1. We used scatterplot to show the relationship between Age vs Gender, Age vs blood pressure, and age vs chest pain type. We also wanted to visualize it in a chart method so we used bar charts to show chest pain vs heart disease as well as who had heart disease. Age vs Heart disease to show which age gender has the most heart disease rates. Last but not least, gender vs heart disease to show our main gender that gets heart disease.
4. **Partition and predict the data and test data**
   1. We used the same method from individual assignment 3, we split our data into x and y variables as well as test x and y variables. We used reg.predict as well to predict our variables.
5. **Get the accuracy score**
   1. For the accuracy score, we used the mean squared error method to get the accuracy.

**Recommendations** – our primary message for the decision makers is to create a prediction model to catch heart disease before

see which patients have an increased chance of developing heart disease. For example, if they have a patient that’s 55 and has non-anginal chest pain, they may have an increased chance of heart disease so should get their body checked out earlier on to catch anything.