Heart Failure Prediction Dataset

Computational Intelligence in Engineering Applications (CIEA)

Prepared by:
Muhammad Azlan (BS-22-IB-103130)
Abdul Ahad (BS-22-MB-100452)

Instructor: Dr. Sufi Tabassum Gul

Date of Submission:
December 11, 2024

Abstract

In this project we were given the task to make a heart failure prediction model. And the algorithm we used to train our model was Linear Regression. We first got some test data of 918 patients with 11 different features from kaggle.com . In order to make an accurate model we first tried to sort out the data and remove any null, empty or duplicate values and if found we would replace the null data with the mean of the correct values. We also plotted bar graphs and histograms for finding the diversity of the actual data. We then plotted pi graphs for the people having heart disease and people not having heart disease and then related to the different features. Then we checked if there were any anomalous values in our data like people having 0 cholesterol etc. And if found with any values replace them with the median of correct values in the data. After that we used logistic regression to train our model by importing from the sklearn library and used 500 iterations for the learning. We used 20% data for testing and 80% for learning. And then we tested our trained network by inputting our own data and then testing how accurately the neural network predicts about heart failure.

Contents

1 State of the Art / Problem Background

Heart failure is a very common disease among individuals and can vary from person to person in how it affects a person. There are many factors in which heart failure can occur and a patient suffering from heart failure can have different symptoms that can help us predict whether someone will have a heart failure or not. Each symptom carries a different weight for having heart failure. In the United States, more than 5 million people suffer from HF, with 550 000 new cases diagnosed annually. In China, 8.9 million people have HF, with a prevalence rate of 1. 3% for those over 35 years of age. The global mortality rates and increasing prevalence of HF make it a significant public health concern, with annual costs estimated at $29 billion due to high hospitalization rates and unsatisfactory prognoses. Predicting mortality can help doctors create appropriate treatment plans, prevent worsening conditions, reduce medical expenses, and improve quality of life. [1]

1.1 Indicators for Heart Failure:

HF is linked to common indicators like difficulty breathing, swelling in the legs, and feeling tired, alongside physical signs such as crackling sounds in the lungs during examination and increased pressure in the jugular veins [2] sex, age, cholesterol level, whether or not you have chest pain, and if you have chest pain what kind of chest pain you have. Heartbeat rate, ECG, and blood pressure are also important indicators used to tell about heart failure.

1.2 Existing Heart Failure Prediction Models:

Previous models that were used in heart failure prediction models were

- Meta Analysis
- Graph Based Attention Model (GRAM)
- PCP-HF white women Model
- Reverse Time Attention Model

1.3 Limitations of current model:

The model that we made is from a previous data available on the internet. We used 80% of the data for training of the model and used the remaining 20% for testing for our model. But the data that was available to us was only of 918 different patients and we had only 11 features to study for the training. In order to get an accurate model able to predict data from patients there should be more data for the network to train on and there should also be more features available because Heat Failure does not only depend on those 11 features. There are many more indicators on which heart failure depends. That is why the model that we made will not always give 100% accurate results but the results of our trained model will be satisfactory enough to be able to use in some cases.

2 Related Work

From inception to 3 November 2022 MEDLINE and EMBASE databases were searched for studies of multivariable models derived, validated and/or augmented for HF prediction in community-based cohorts. Discrimination measures for models with c-statistic data from more than 3 cohorts were pooled by Bayesian meta-analysis, with heterogeneity assessed through a 95% prediction interval (PI). Risk of bias was assessed using PROBAST. We included 36 studies with 59 prediction models. In meta-analysis, the Atherosclerosis Risk in Communities (ARIC) risk score (summary c-statistic 0.802, 95% confidence interval [CI] 0.707-0.883), GRaph-based Attention Model (GRAM; 0.791, 95% CI 0.677-0.885), Pooled Cohort equations to Prevent Heart Failure (PCP-HF) white men model (0.820, 95% CI 0.792-0.843), PCP-HF white women model (0.852, 95% CI 0.804-0.895), and Reverse Time Attention model (RETAIN; 0.839, 95% CI 0.748-0.916) had a statistically significant 95% PI and excellent discrimination performance. The ARIC risk score and PCP-HF models had significant summary discrimination among cohorts with a uniform prediction window. 77% of model results were at high risk of bias, certainty of evidence was low, and no model had a clinical impact study. [3]

2.1 Meta Analysis:

Meta-analysis is a statistical tool that allows the analysis of results from various scientific studies, which are often not performed in the same place or using the same method. The data used in meta-analysis may be proprietary or may be obtained from literature or various databases. Meta-analysis is a crucial part of many systematic reviews, although not all systematic reviews include a meta-analysis. Therefore, meta-analysis should not be equated with systematic reviews. It is not incorrect to say that meta-analysis synthesizes the results from several studies and yields a new set of results. In other words, meta-analysis itself can, under certain conditions, be considered a method of producing new data. [4]

2.2 Graph Based Attention Model:

GRAM, a method that infuses information from medical ontologies into deep learning models via neural attention. Considering the frequency of a medical concept in the EHR data and its ancestors in the ontology, GRAM optimizes the medical concept by adaptively combining its ancestors via attention mechanism (i.e. weighted sum of the representations of ancestors). The attention mechanism is trained in an end-to-end fashion with the neural network model that predicts the onset of diseases. [5]

2.3 PCP-HF White Women Model:

Race- and sex-specific 10-year risk equations for HF were derived and validated from individual-level data from 7 community-based cohorts with at least 12 years of follow-up. Participants who were recruited between 1985-2000, between 30 to 80 years, and were free of cardiovascular disease at baseline were included to create a pooled cohort (PC) and randomly split for derivation and internal validation. Model performance was also assessed in 2 additional cohorts.In the derivation sample of the PC (n=11771), 58% were women, 22% were black with a mean age 52±12 years, and HF occurred in 1339 participants. Predictors of HF included in the race-sex specific models were age, blood pressure (treated or untreated), fasting glucose (treated or untreated), body mass index, cholesterol, smoking status, and QRS duration. The PC equations to Prevent HF (PCP-HF) model had good discrimination and strong calibration in internal and external validation cohorts. A web-based tool was developed to facilitate clinical application of this tool. [6]

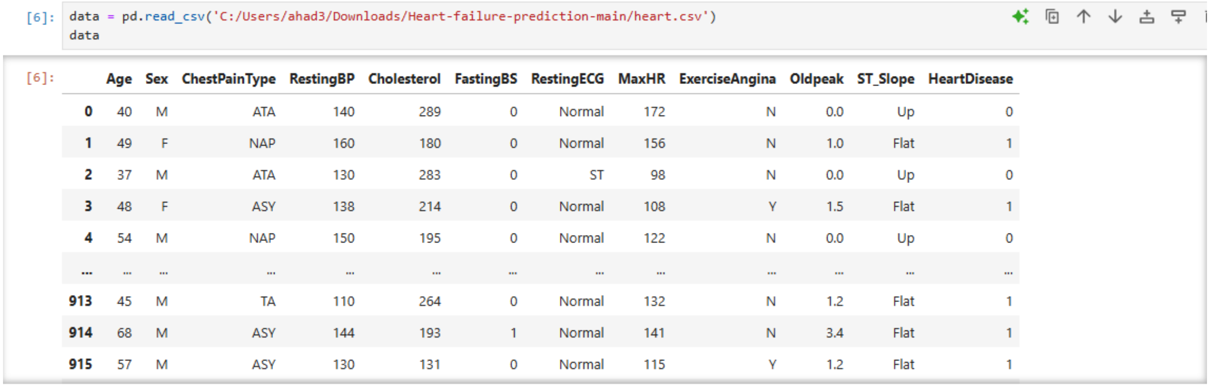
3 Exploratory Data Analysis (EDA)

3.1 Dataset Analysis

3.1.1 Reference Data:

We imported our data from Kaggel.com [7] which was in the form of a CSV file and we converted it to a matrix so that we can have a proper analysis of the data. The data turned out to be a 918 X 12 Matrix having data of 918 different patients and each patient's 11 features were described and in the last column it was told weather the patient had a heart failure or not as described in figure 1. The 11 features that were described in the data were:

- Age: age of the patient [years]
- Sex: sex of the patient [M: Male, F: Female]
- ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- RestingBP: resting blood pressure [mm Hg]
- Cholesterol: serum cholesterol [mm/dl]
- FastingBS: fasting blood sugar [1: if FastingBS ¿ 120 mg/dl, 0: otherwise]
- RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or - ST elevation or depression of ¿ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
- MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
- ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
- Oldpeak: oldpeak = ST [Numeric value measured in depression]
- ST Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- HeartDisease: output class [1: heart disease, 0: Normal]



```
[6]: data = pd.read_csv('C:/Users/ahad3/Downloads/Heart-failure-prediction-main/heart.csv')
     data
```

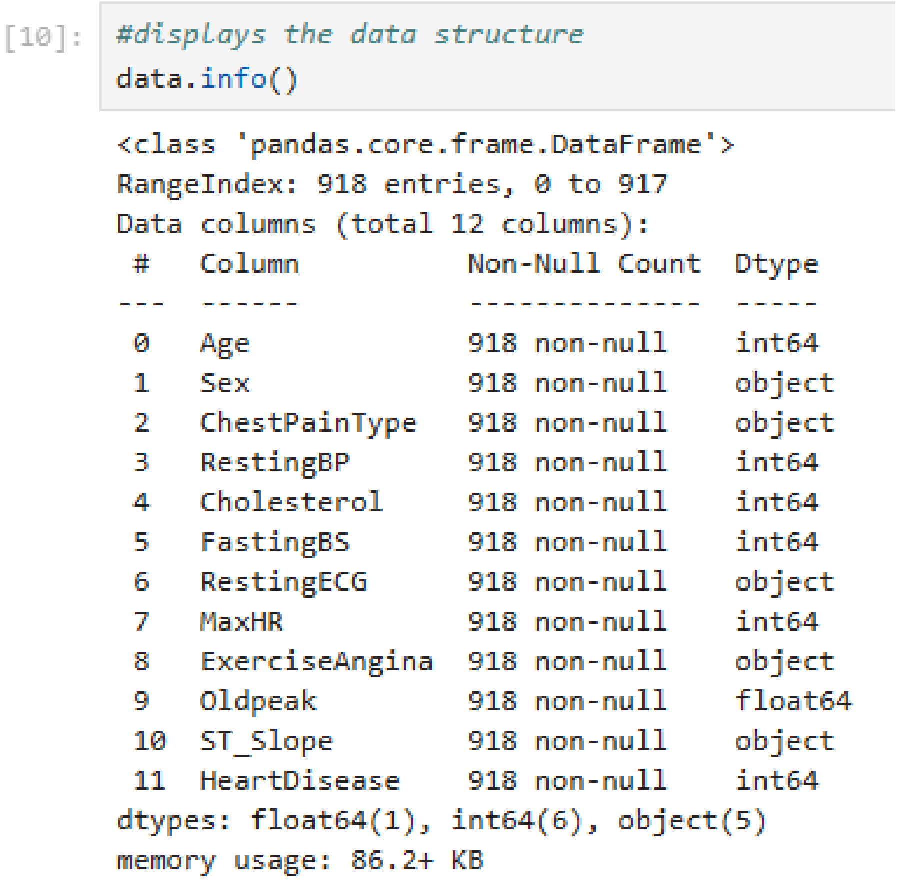
[6]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
...	...	...	...	...	...	...	...	...	...	...	...	...
913	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
914	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
915	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1

Figure 1: Reference data

3.1.2 Checking for Null values:

After the data was imported we checked weather there were any null entries in our data and if there were any null entries we would replace them with the mean of the non-null entries. But as in figure 2 luckily there were no null entries in our data.



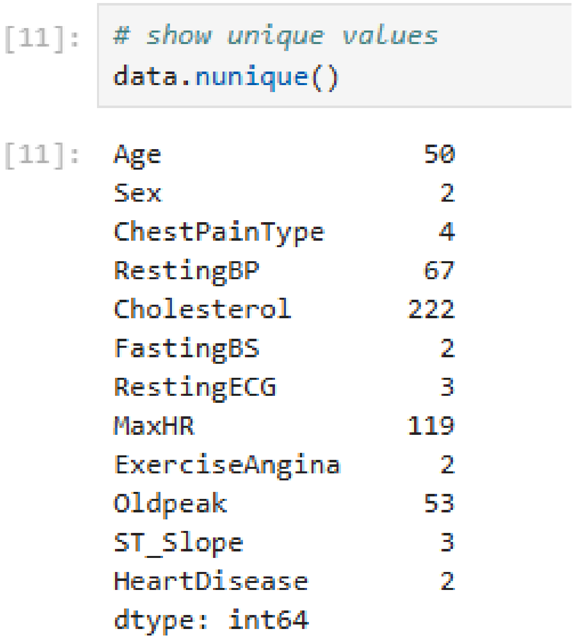
```
[10]: #displays the data structure
      data.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 918 entries, 0 to 917
      Data columns (total 12 columns):
       #   Column          Non-Null Count  Dtype
      ---  ------          --------------  -----
       0   Age             918 non-null    int64
       1   Sex             918 non-null    object
       2   ChestPainType   918 non-null    object
       3   RestingBP       918 non-null    int64
       4   Cholesterol     918 non-null    int64
       5   FastingBS       918 non-null    int64
       6   RestingECG      918 non-null    object
       7   MaxHR           918 non-null    int64
       8   ExerciseAngina  918 non-null    object
       9   Oldpeak         918 non-null    float64
       10  ST_Slope        918 non-null    object
       11  HeartDisease    918 non-null    int64
      dtypes: float64(1), int64(6), object(5)
      memory usage: 86.2+ KB
```

Figure 2: Checking Null values in the data

3.1.3 Checking for Unique values:

After looking for null values we found all the different values we had in our data to see how much diversity was available in our data as in figure 3. We also found out the description of data i.e the mean of all values of all the features, standard deviation, minimum values, maximum values etc.



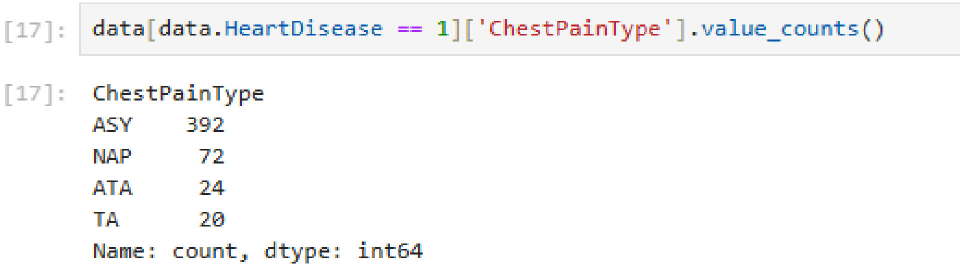
```
[11]:  # show unique values
       data.nunique()

[11]:  Age                50
       Sex                 2
       ChestPainType       4
       RestingBP          67
       Cholesterol       222
       FastingBS           2
       RestingECG          3
       MaxHR             119
       ExerciseAngina      2
       Oldpeak            53
       ST_Slope            3
       HeartDisease        2
       dtype: int64
```

Figure 3: Checking Unique values in the data

3.1.4 Calculating values of features for heart patients:

We then looked for the individual amount of data for heart patients and calculated how many of each feature corresponds to a person having heart failure. for example in all of the 918 people 508 people had heart failure and from them 458 people were men and only 50 were women. This tells that being a male would have more risk of having heart failure as compared to female. We found this relation for all the features and one of the outputs is displayed in figure 4.



```
[17]:  data[data.HeartDisease == 1]['ChestPainType'].value_counts()

[17]:  ChestPainType
       ASY    392
       NAP     72
       ATA     24
       TA      20
       Name: count, dtype: int64
```

Figure 4: Amount of heart patients having different kinds of chest pain

3.1.5 Co-Relation Map:

A co-relation map is a graph that tells the relation of one feature to the other in with respect to the output. A co-relation map is plotted for all the values of features to all other values of the features. For example how Age related with Gender in order for a person to have hear disease or how Chest pain type relates with cholesterol level in order for a person to have heart disease. The co-relation map of our data is given in Figure 5. The darker the color is of a cell in the co-relation map the less dependent it is to that feature. The lighter the color is of the cell in the co-relation map the more dependent it is to that feature.

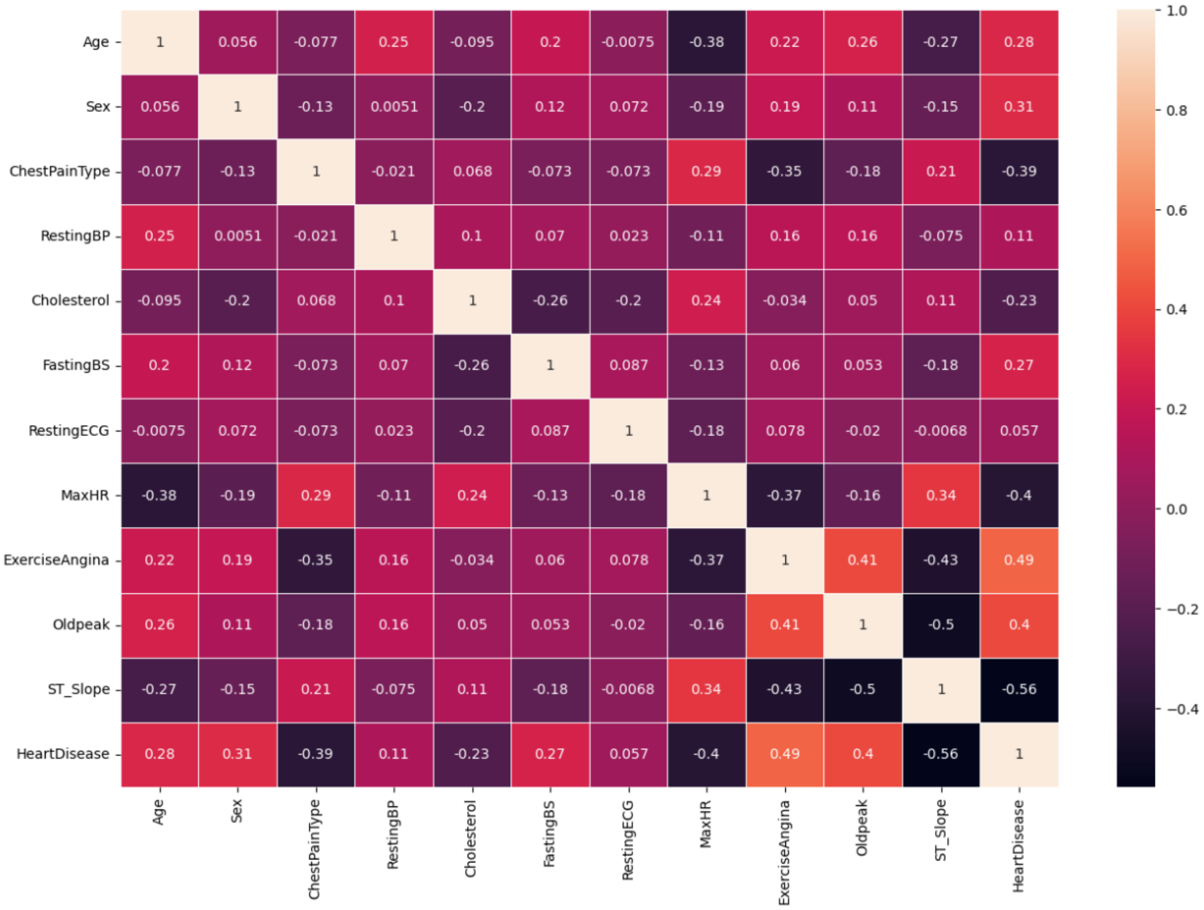


Figure 5: Co-Relation Map of reference data for a person having heart disease

3.1.6 Pi Graphs and Histograms of dataset:

After all of that we plotted the pi graphs of the data for all the features with respect to a person having heart disease or not. So for that we got pi graphs for each feature. Since the cholesterol levels and resting BP values had a lot of variety we could not plot their pi graphs so we used histograms to plot these 2 features. The outputs for the pi graphs is given in figure 6 and figure 7 and the Histograms are given in figure 8.

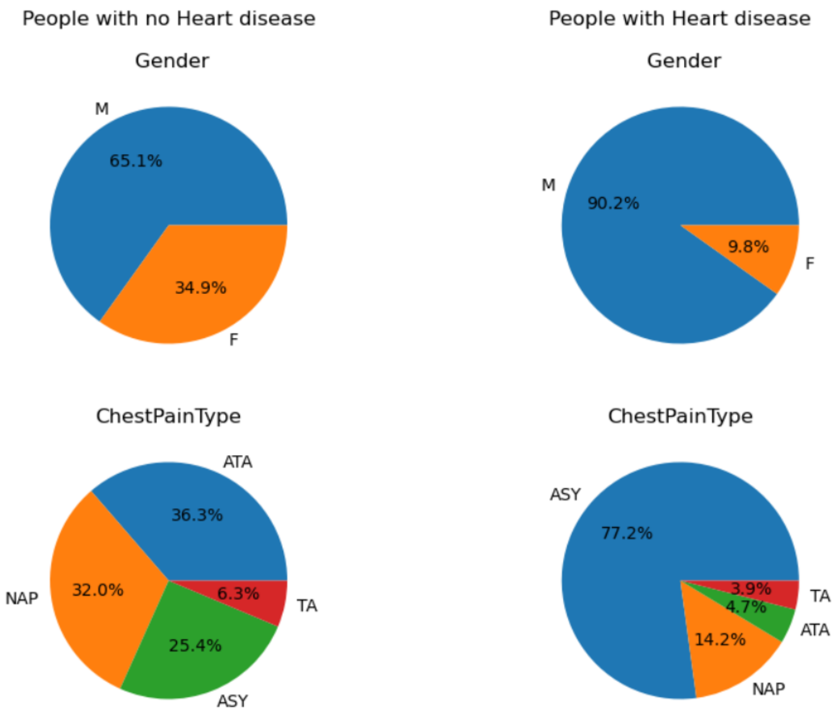


Figure 6: Pi Graph of gender and chest pain type

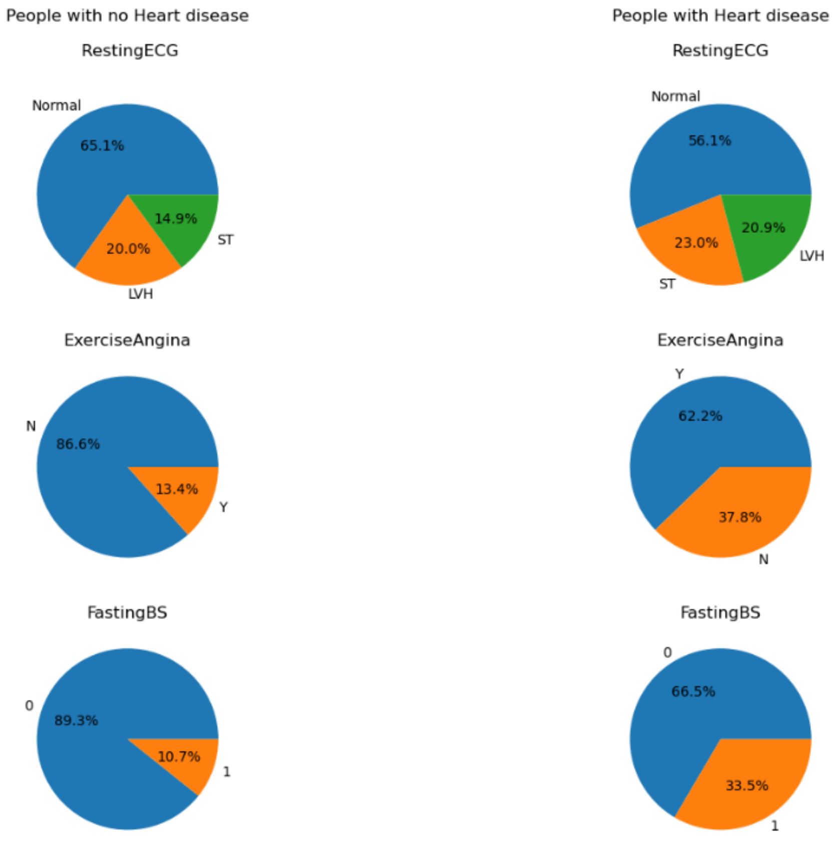


Figure 7: Pi Graph of ECG, Exercise Angina and Fasting BS

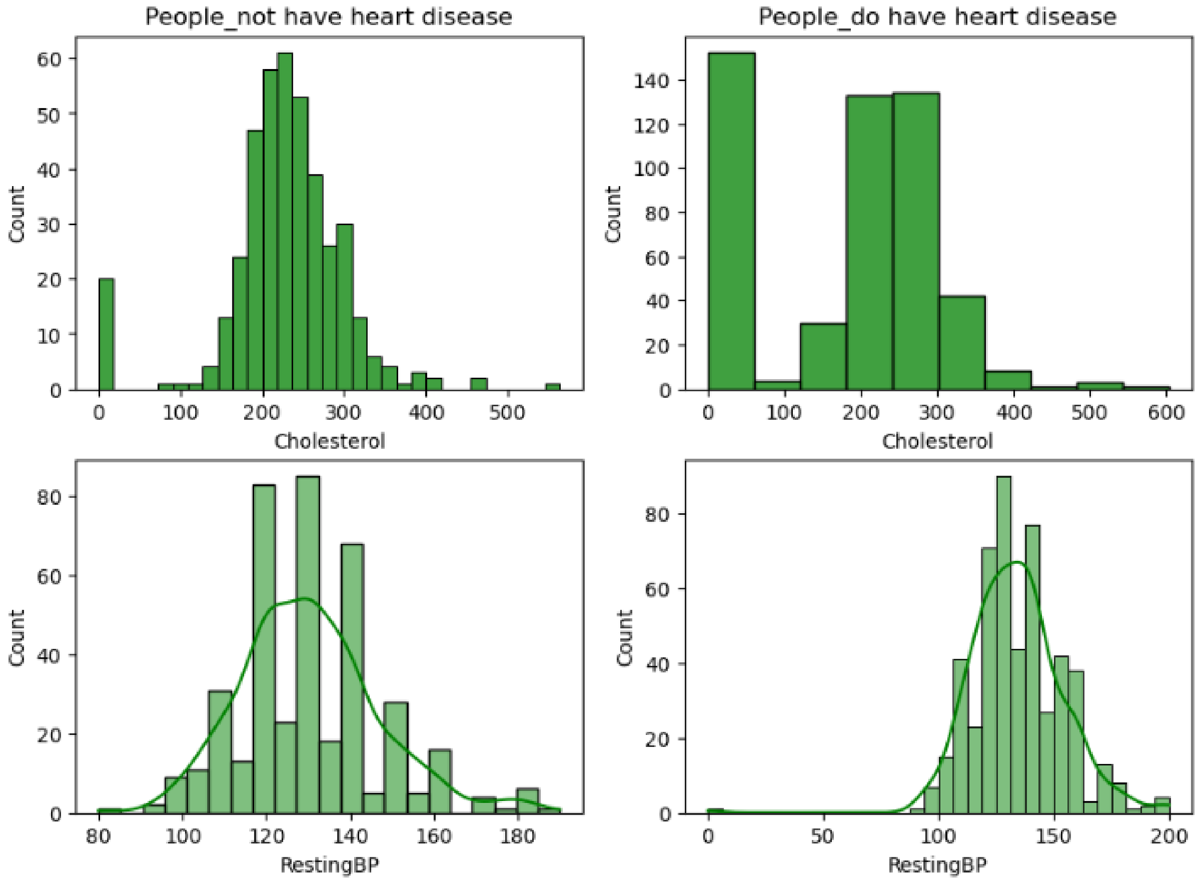


Figure 8: Histograms of Cholesterol and Resting BP Values

4 Explanation of Selected Classifier

4.1 Logistic Regression:

A logistic regression model describes and estimates the relationship between 1 binary dependent variable, also known as an outcome variable, and 1 or more independent variables, also known as covariates or explanatory variables. Logistic regression models are versatile, have a powerful interpretation, and have been used to describe phenomena in diverse areas of medical and nonmedical research. Similar to other regression models, a logistic regression model is often used to evaluate predictors and to adjust for confounders and/or interactions. These models are used to analyze retrospective data, including case-control studies, as well as to create prediction algorithms, which can be depicted in nomograms or online calculators that communicate the probability of an event, eg toxicity or lightbulb failure in the 2 previously noted examples. [8]

4.2 Working:

Logistic Regression is a classification modeling technique to tell about which class a data belongs to. We use Logistic Regression to tell about Categorical Data Analysis. And we use previous data to predict weather a new data will belong to one category or the other. We use sigmoid function to classify the data into one of two categories either the data belongs to one category or the data belongs to other category. And we assign a threshold value and if the value of the sigmoid function is above the threshold value then the data belongs to one category and if it is less then the threshold then it belongs to the other category. The sigmoid function is given as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

The plot for the sigmoid function is given as:

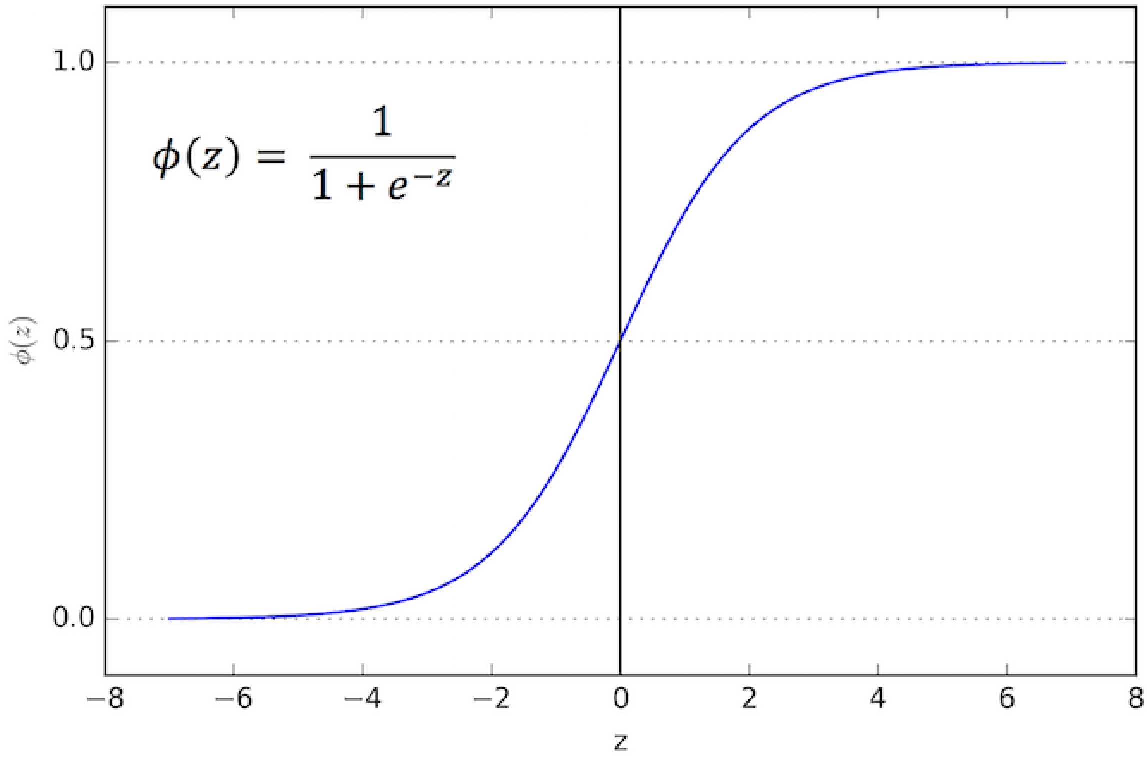
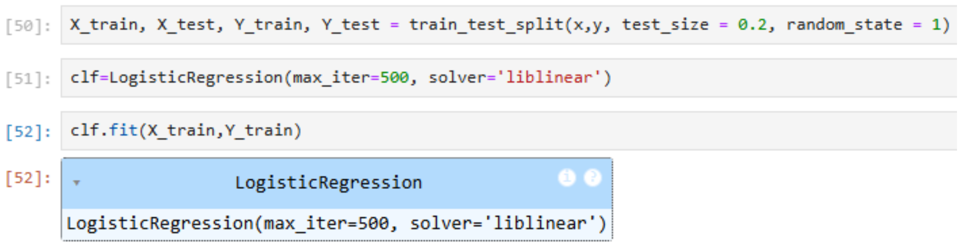


Figure 9: Plot of Sigmoid Function

The central mathematical concept that underlies logistic regression is the logit—the natural logarithm of an odds ratio. The simplest example of a logit derives from a 2 × 2 contingency table. Consider an instance in which the distribution of a dichotomous outcome variable (a child from an inner city school who is recommended for remedial reading classes) is paired with a dichotomous predictor variable(gender). [9]

4.3 Implementation in Python:

We used logistic regression by importing the library of sklearn and then imported logistic regression from there. Then we splitted our data into two parts. One part was used for training and the other part was used for testing. We used 80% of our data for training and the other 20% for testing. The accuracy that we achieved at the end was greater than 90% so it means that our model can predict data with very high precision. We did 500 iterations to perform our training in order to get satisfactory results and we used ”Liblinear” as our solver because it is used to train data for binary outputs and since we had only 2 results at the output weather a person can have heart failure or the person cannot have heart failure. So that’s why we can use Liblinear as our solver as you can see in the figure 10. The ROC graph we got after applying logistic regression is:



```
[50]: X_train, X_test, Y_train, Y_test = train_test_split(x,y, test_size = 0.2, random_state = 1)

[51]: clf=LogisticRegression(max_iter=500, solver='liblinear')

[52]: clf.fit(X_train,Y_train)

[52]: LogisticRegression
      LogisticRegression(max_iter=500, solver='liblinear')
```

Figure 10: Snippet of code implementing logistic regression

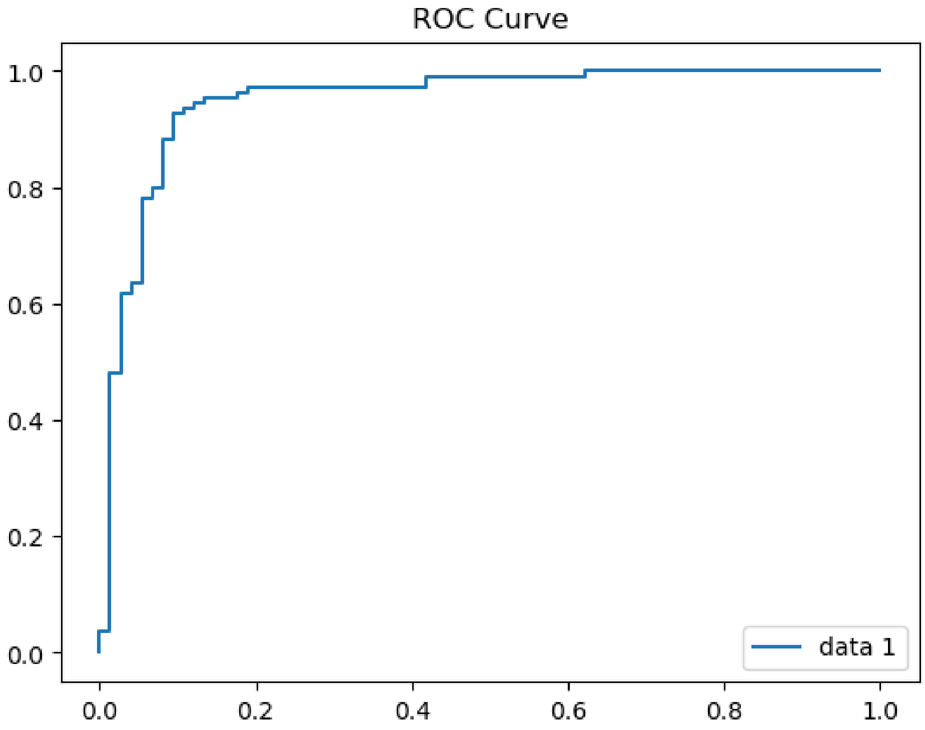


Figure 11: ROC graph of the model

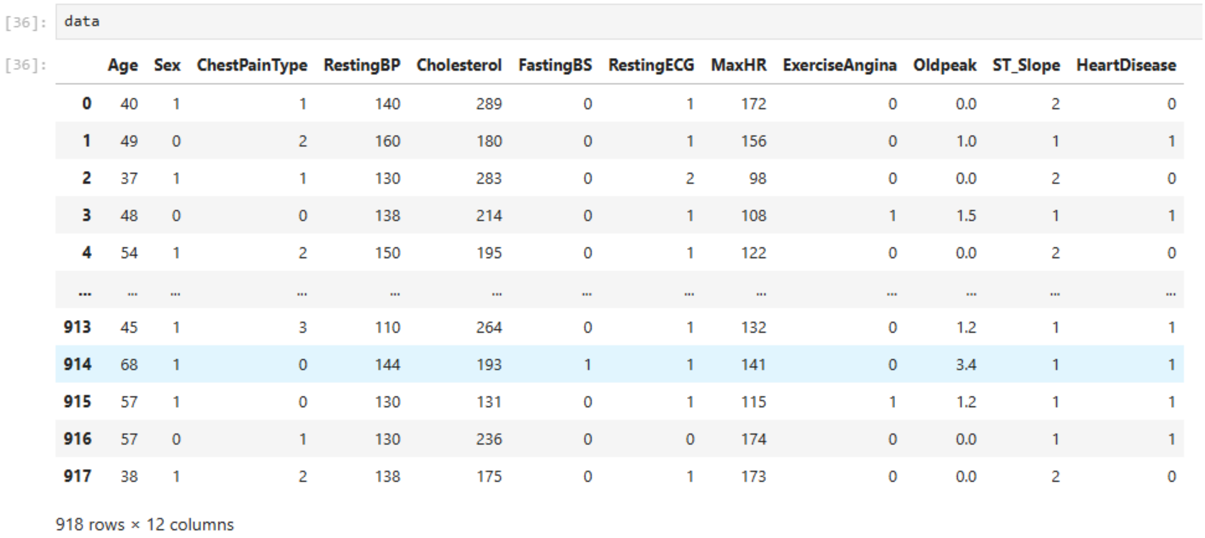
5 Implementation

5.1 Tools used:

We used python to implement our training model and we used jupyter notebook in order to run our code in different sections. Python also helped us use different built in libraries so that we could process our data very easily and apply different commands on our data. We used kaggel.com [7] to get our data in the form of a CSV file and then we imported that CSV file into jupyter notebook. We also used sklearn library in python to import our learning algorithm which was logistic regression and using python we could easily implement our learning algorithm without writing down each and every step.

5.2 Pre-Processing of data:

There were some values in our data which were in string format and could not be processed by our algorithm in that way. So we needed to encode them to some integer values so that it could be processed later on. After encoding the data all the columns of the data changed to either integer or floating point values as you can see in figure 11.



```
[36]: data
```

[36]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	1	1	140	289	0	1	172	0	0.0	2	0
1	49	0	2	160	180	0	1	156	0	1.0	1	1
2	37	1	1	130	283	0	2	98	0	0.0	2	0
3	48	0	0	138	214	0	1	108	1	1.5	1	1
4	54	1	2	150	195	0	1	122	0	0.0	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...
913	45	1	3	110	264	0	1	132	0	1.2	1	1
914	68	1	0	144	193	1	1	141	0	3.4	1	1
915	57	1	0	130	131	0	1	115	1	1.2	1	1
916	57	0	1	130	236	0	0	174	0	0.0	1	1
917	38	1	2	138	175	0	1	173	0	0.0	2	0

918 rows × 12 columns

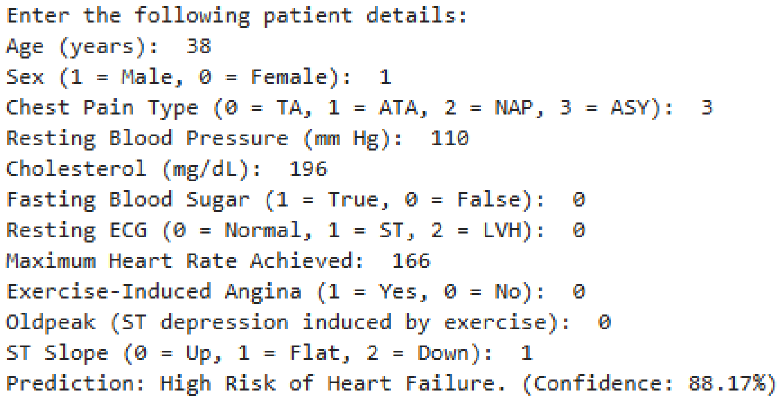
Figure 12: Encoded dataset

After that we found the 3 smallest and 3 largest values for each feature in the dataset and checked weather there was any unrealistic values in our data and if there were for any feature they would be replaced by the median of the values of that feature. We did this for all of the 11 features in our dataset.

5.3 Training and Testing:

After all of this we exported our model into an external file in our computer using a pickle file and then we imported our pickle file back into our code and then gave the model some input data of our own and the model had to predict weather the person will have heart failure or not.

In figure 13 you can see that we inputted the data of a random person and the model predicted with 88.17% surety that the person has had a heart failure in the past.



```
Enter the following patient details:
Age (years):  38
Sex (1 = Male, 0 = Female):  1
Chest Pain Type (0 = TA, 1 = ATA, 2 = NAP, 3 = ASY):  3
Resting Blood Pressure (mm Hg):  110
Cholesterol (mg/dL):  196
Fasting Blood Sugar (1 = True, 0 = False):  0
Resting ECG (0 = Normal, 1 = ST, 2 = LVH):  0
Maximum Heart Rate Achieved:  166
Exercise-Induced Angina (1 = Yes, 0 = No):  0
Oldpeak (ST depression induced by exercise):  0
ST Slope (0 = Up, 1 = Flat, 2 = Down):  1
Prediction: High Risk of Heart Failure. (Confidence: 88.17%)
```

Figure 13: Inputting data manually and getting prediction

5.4 Confusion Matrix:

After training and testing we plotted the confusion matrix of our model which tells about the prediction of the data.
The top left box of confusion matrix tells weather the model correctly predicted as negative. The model accurately predicted 66 samples correctly as negative.
The top right tells about false positives and there were only 8 of them
The bottom left tells about false negative results.
And the bottom right tells about true positives and there were 102 of those samples.

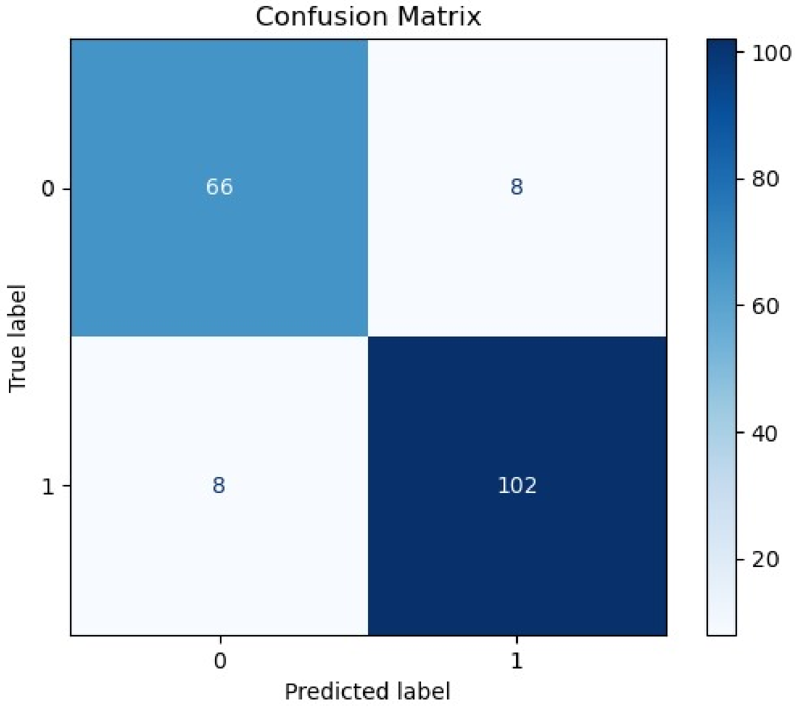


Figure 14: Plot of Confusion Matrix

6 Conclusions and Comments

In conclusion we used logistic regression to train a model coded in python using jupyter notebook and some builtin python libraries to and trained from online available data to predict weather a person will have a heart failure or not by judging the person on 11 different parameters. We leaned a lot from this project including how to implement neural networks in real problems and how we can utilize neural networks to make predictions for patients facing certain symptoms and tell before hand that a person might get a heart failure if they have certain diseases. This project can also be used in different medical fields to accurately predict weather a person will have a heart failure or not with respect to the 11 given parameters and how we can save many people who might suffer from this disease.

References

[1] M. Saqib, P. Perswani, A. Muneem, H. Mumtaz, F. Neha, S. Ali, and S. Tabassum, "Machine learning in heart failure diagnosis, prediction, and prognosis: review," *Ann Med Surg (Lond)*, vol. 86, no. 6, pp. 3615–3623, May 2024.

[2] M.-S. Kim, J.-H. Lee, E. J. Kim, D.-G. Park, S.-J. Park, J. J. Park, M.-S. Shin, B. S. Yoo, J.-C. Youn, S. E. Lee, S. H. Ihm, S. Y. Jang, S.-H. Jo, J. Y. Cho, H.-J. Cho, S. Choi, J.-O. Choi, S. W. Han, K. K. Hwang, E. S. Jeon, M.-C. Cho, S. C. Chae, and D.-J. Choi, "Korean guidelines for diagnosis and management of chronic heart failure," *Korean Circ J*, vol. 47, no. 5, pp. 555–643, Sep. 2017.

[3] R. Nadarajah, T. Younsi, E. Romer, K. Raveendra, Y. M. Nakao, K. Nakao, F. Shuweidhi, D. C. Hogg, R. Arbel, D. Zahger, Z. Iakobishvili, G. C. Fonarow, M. C. Petrie, J. Wu, and C. P. Gale, "Prediction models for heart failure in the community: A systematic review and meta-analysis," *Eur J Heart Fail*, vol. 25, no. 10, pp. 1724–1738, Jul. 2023.

[4] B. K. Hackenberger, "Bayesian meta-analysis now - let's do it," *Croat Med J*, vol. 61, no. 6, pp. 564–568, Dec. 2020.

[5] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "GRAM: Graph-based attention model for healthcare representation learning," *KDD*, vol. 2017, pp. 787–795, Aug. 2017.

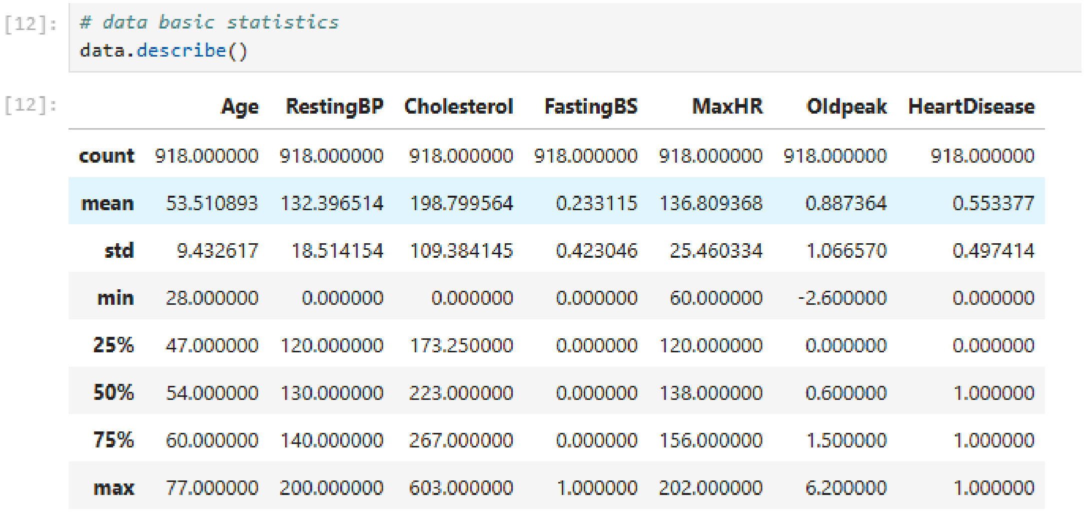
[6] S. S. Khan, H. Ning, S. J. Shah, C. W. Yancy, M. Carnethon, J. D. Berry, R. J. Mentz, E. O'Brien, A. Correa, N. Suthahar, R. A. de Boer, J. T. Wilkins, and D. M. Lloyd-Jones, "10-year risk equations for incident heart failure in the general population," *J Am Coll Cardiol*, vol. 73, no. 19, pp. 2388–2397, May 2019.

[7] Kaggle, "Heart failure prediction dataset," 12 2024, accessed: 2024-12-11. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

[8] E. C. Zabor, C. A. Reddy, R. D. Tendulkar, and S. Patil, "Logistic regression in clinical studies," *International Journal of Radiation Oncology, Biology, Physics*, vol. 112, no. 2, pp. 271–277, Feb. 2022.

[9] J. Peng, K. Lee, and G. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research - J EDUC RES*, vol. 96, pp. 3–14, 09 2002.

Appendix

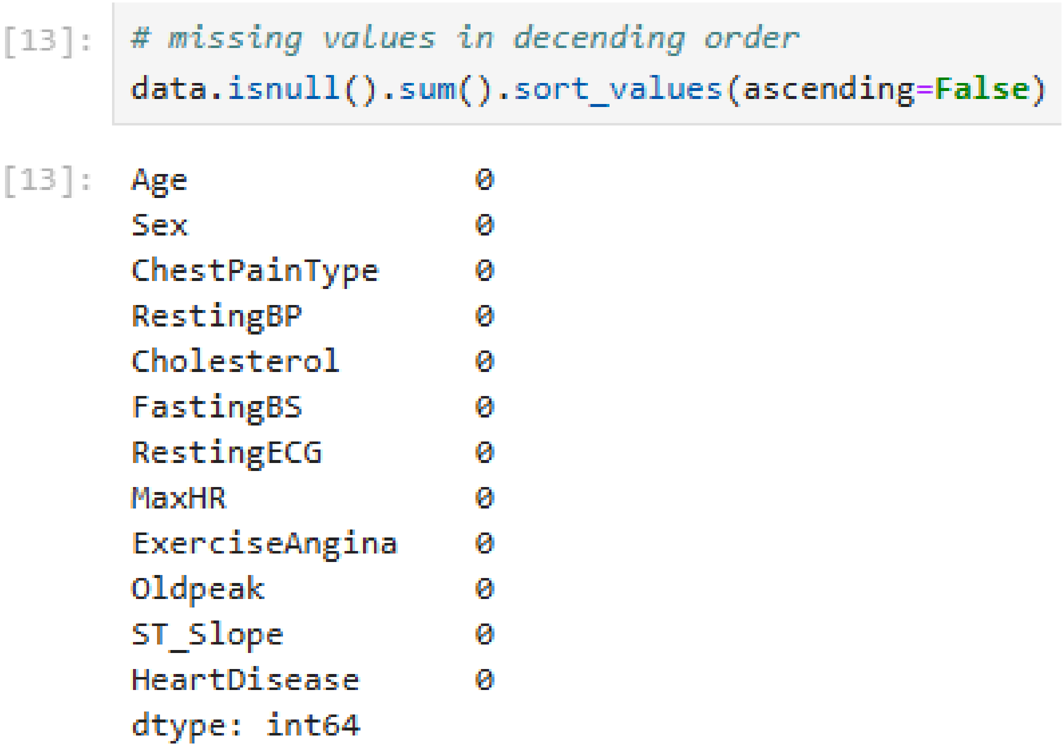


```
[12]: # data basic statistics
      data.describe()
```

[12]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

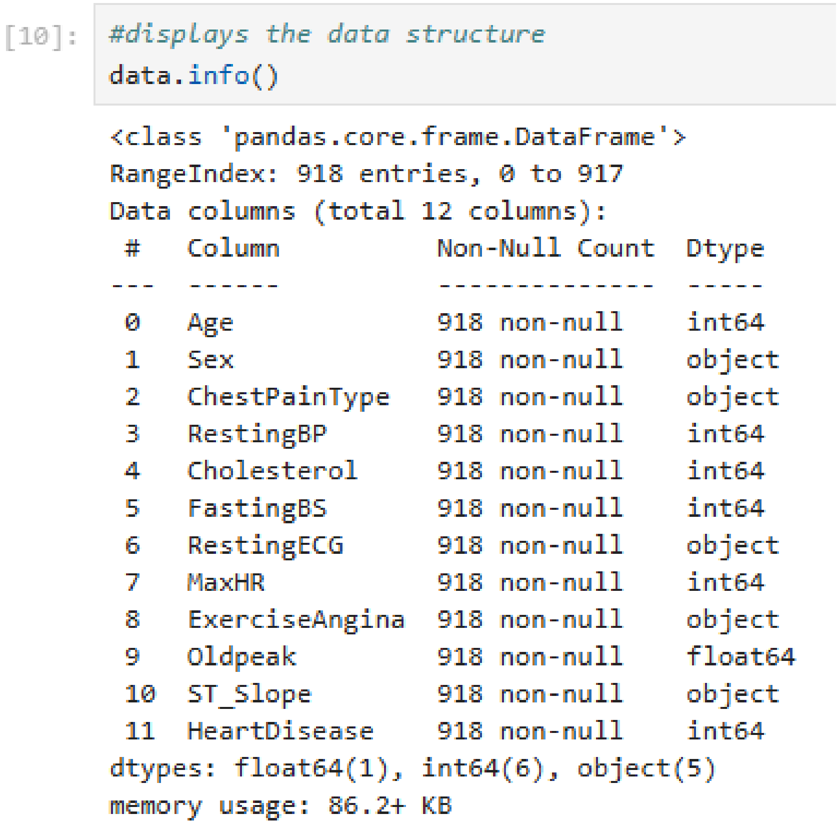
Figure 15: Description of data



```
[13]: # missing values in decending order
      data.isnull().sum().sort_values(ascending=False)
```

```
[13]: Age               0
      Sex               0
      ChestPainType     0
      RestingBP         0
      Cholesterol       0
      FastingBS         0
      RestingECG        0
      MaxHR             0
      ExerciseAngina    0
      Oldpeak           0
      ST_Slope          0
      HeartDisease      0
      dtype: int64
```

Figure 16: Showing no null values in our data



```
[10]: #displays the data structure
      data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Age             918 non-null    int64  
 1   Sex             918 non-null    object 
 2   ChestPainType   918 non-null    object 
 3   RestingBP       918 non-null    int64  
 4   Cholesterol     918 non-null    int64  
 5   FastingBS       918 non-null    int64  
 6   RestingECG      918 non-null    object 
 7   MaxHR           918 non-null    int64  
 8   ExerciseAngina  918 non-null    object 
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object 
 11  HeartDisease    918 non-null    int64  
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

Figure 17: Changing datatypes after encoding

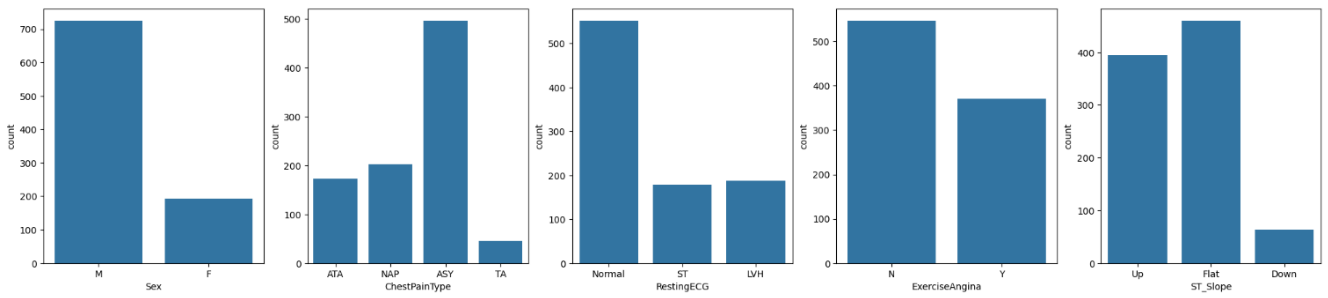


Figure 18: Bar graphs of features