Data Mining Project

Machine Learning for Arrhythmia Classification

Rohan Reddy Pathi

Data Analytics Engineering
College of Engineering
Northeastern University
Boston

Instructor

Prof. Srinivasan Radhakrishnan

4th April 2024

What is Arrhythmia

Arrhythmia refers to an irregular heartbeat, where the heart may beat too fast (tachycardia), too slow (bradycardia), or with an irregular rhythm. The heart's electrical system controls the rate and rhythm of heartbeats. In a healthy heart, electrical signals travel in a coordinated manner, prompting the heart's chambers to contract and pump blood efficiently. However, in arrhythmias, this electrical signaling is disrupted, causing the heart to beat irregularly.

Arrhythmias can occur due to various factors, including heart disease, electrolyte imbalances, high blood pressure, stress, certain medications, congenital heart defects, and other medical conditions. While some arrhythmias are harmless and may not require treatment, others can be life-threatening and may require medical intervention.

What is ECG

An electrocardiogram (ECG or EKG) is a non-invasive medical test that records the electrical activity of the heart over a period of time. It's a commonly used tool for diagnosing various heart conditions, including arrhythmias.

During an ECG, electrodes are placed on the skin of the chest, arms, and legs. These electrodes detect the electrical signals generated by the heart each time it beats. The signals are then recorded and displayed on a graph, showing the heart's electrical activity as a series of waves.

Its primary role lies in identifying normal heart rhythms and detecting abnormalities, including arrhythmias. By analyzing the ECG waveform, healthcare providers can discern irregularities indicative of various arrhythmias such as atrial fibrillation, ventricular tachycardia, or bradycardia. Additionally, ECGs offer detailed insights into the characteristics of arrhythmias, including their duration, frequency, and pattern, aiding in treatment decision-making.

Data Source and Objective

The dataset utilized in this project is accessible via the UCI Machine Learning Repository and can be accessed at the following address:
https://archive.ics.uci.edu/ml/datasets/Arrhythmia.

The primary objective of this project is to predict whether an individual is afflicted with arrhythmia and if so, to classify it into one of the 12 available groups.

Data Description

Comprising 452 distinct examples, the dataset spans across 16 classes. Within these examples, 245 pertain to individuals categorized as ”normal”. Additionally, the dataset encompasses 12 different types of arrhythmias, with notable representations including ”coronary artery disease” and ”right bundle branch block”.

The dataset encompasses 279 features, encompassing various patient attributes such as age, sex, weight, height, alongside pertinent information extracted from electrocardiograms. It is worth noting that the number of features is relatively high in comparison to the available number of examples.

Below are the description of the variables in the dataset:

Features:

1. Age: Age in years , linear
2. Sex: Sex (0 = male; 1 = female) , nominal
3. Height: Height in centimeters , linear
4. Weight: Weight in kilograms , linear
5. QRS duration: Average of QRS duration in msec., linear
6. P-R interval: Average duration between onset of P and Q waves in msec., linear

7. Q-T interval: Average duration between onset of Q and offset of T waves in msec., linear
8. T interval: Average duration of T wave in msec., linear
9. P interval: Average duration of P wave in msec., linear

   Vector angles in degrees on front plane of:, linear
10. QRS
11. T
12. P
13. QRST
14. J
15. Heart rate: Number of heart beats per minute ,linear

    Of channel DI:
    Average width, in msec., of: linear

16. Q wave
17. R wave
18. S wave
19. R' wave, small peak just after R
20. S' wave
21. Number of intrinsic deflections, linear
22. Existence of ragged R wave, nominal
23. Existence of diphasic derivation of R wave, nominal
24. Existence of ragged P wave, nominal
25. Existence of diphasic derivation of P wave, nominal

26. Existence of ragged T wave, nominal
27. Existence of diphasic derivation of T wave, nominal
28. Of channel DII: 28 .. 39 (similar to 16 .. 27 of channel DI)
29. Of channels DIII: 40 .. 51
30. Of channel AVR: 52 .. 63
31. Of channel AVL: 64 .. 75
32. Of channel AVF: 76 .. 87
33. Of channel V1: 88 .. 99
34. Of channel V2: 100 .. 111
35. Of channel V3: 112 .. 123
36. Of channel V4: 124 .. 135
37. Of channel V5: 136 .. 147
38. Of channel V6: 148 .. 159
    Of channel DI: Amplitude , * 0.1 milivolt, of
39. JJ wave, linear
40. Q wave, linear
41. R wave, linear
42. S wave, linear
43. R' wave, linear
44. S' wave, linear
45. P wave, linear
46. T wave, linear
47. QRSA , Sum of areas of all segments divided by 10, (Area= width * height / 2), linear

48. QRSTA = QRSA + 0.5 * width of T wave * 0.1 * height of T wave. (If T is diphasic then the bigger segment is considered), linear
49. Of channel DII: 170 .. 179
50. Of channel DIII: 180 .. 189
51. Of channel AVR: 190 .. 199
52. Of channel AVL: 200 .. 209
53. Of channel AVF: 210 .. 219
54. Of channel V1: 220 .. 229
55. Of channel V2: 230 .. 239
56. Of channel V3: 240 .. 249
57. Of channel V4: 250 .. 259
58. Of channel V5: 260 .. 269
59. Of channel V6: 270 .. 279

Target Classes
In the Arrhythmia dataset, there are 16 target classes, each corresponding to a specific cardiac condition:

1. Normal - 245 instances
2. Ischemic changes (Coronary Artery Disease) - 44 instances
3. Old Anterior Myocardial Infarction - 15 instances
4. Old Inferior Myocardial Infarction - 15 instances
5. Sinus tachycardia - 13 instances
6. Sinus bradycardia - 25 instances
7. Ventricular Premature Contraction (PVC) - 3 instances
8. Supraventricular Premature Contraction - 2 instances

9. Left bundle branch block - 9 instances
10. Right bundle branch block - 50 instances
11. 1st degree Atrioventricular block - 0 instances
12. 2nd degree AV block - 0 instances
13. 3rd degree AV block - 0 instances
14. Left ventricular hypertrophy - 4 instances
15. Atrial Fibrillation or Flutter - 5 instances
16. Others - 22 instances

These classes represent the distribution of different cardiac conditions present in the dataset, with varying frequencies of occurrence.

Understanding the Data

In the dataset, all features are represented as continuous float values, except for 'sex', 'heart rate', and 'class'. For 'sex', the values are encoded as 0 and 1, representing male and female respectively. Regarding 'class', it spans from 1 to 16, corresponding to the cardiac conditions outlined in the previous section.

Data Preprocessing

At first glance, the data had no missing values, but upon further investigation, it was found that the missing values in the data were filled with a question mark character (?). So, I first replaced all the (?) with NaN from the numpy package.

The total number of null values in the dataset were found to be 408, with the majority originating from the QRST column. Since the QRST column pertains to the QRS-T angle, which measures the difference in mean vectors of depolarization and repolarization derived from an ECG, and lacks relevant data, the decision was made to drop the entire column. For

the remaining missing values, they are imputed using the mean of the respective columns.

As the data is already encoded, the values in categorical columns like Sex, Heart Rate and classes were conerted into float, so we revert them back to integer datatype.

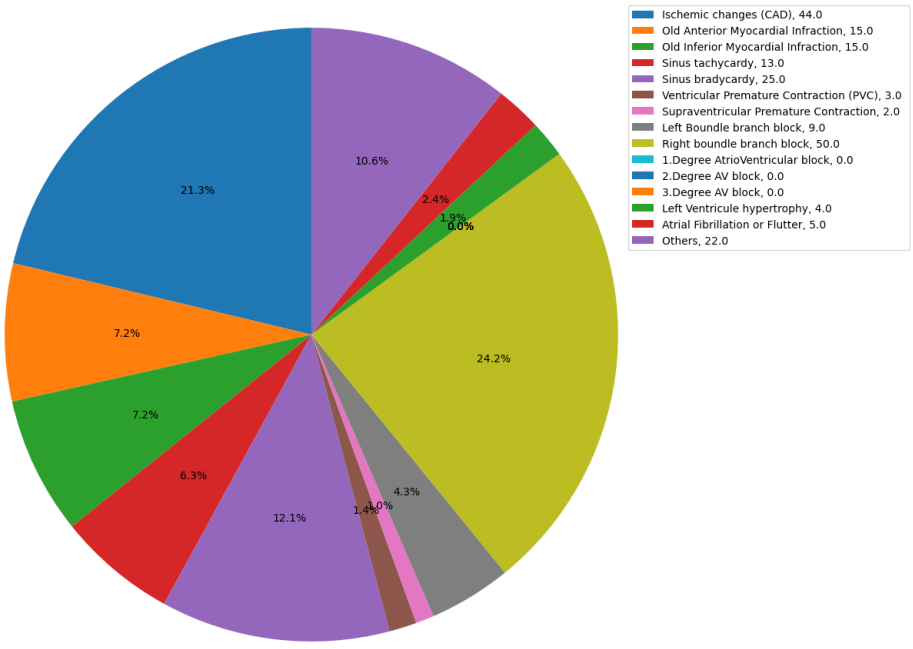
To generate the final dataset, the columns are given names and we relocate the target attribute from the dataframe to a separate object.

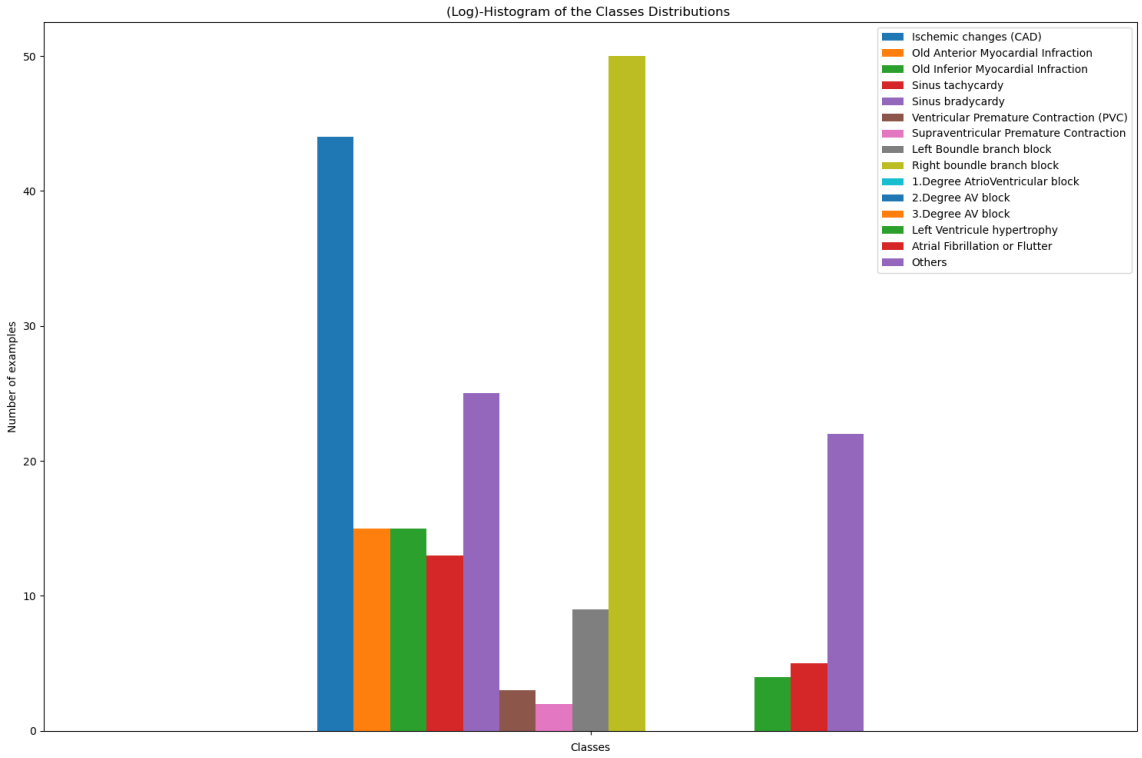
Exploratory Data Analysis

During exploratory data analysis (EDA), the distribution of both classes and features was investigated to gain insights from the dataset. Various methods and visualizations were employed to conduct this analysis.

Distribution of classes:

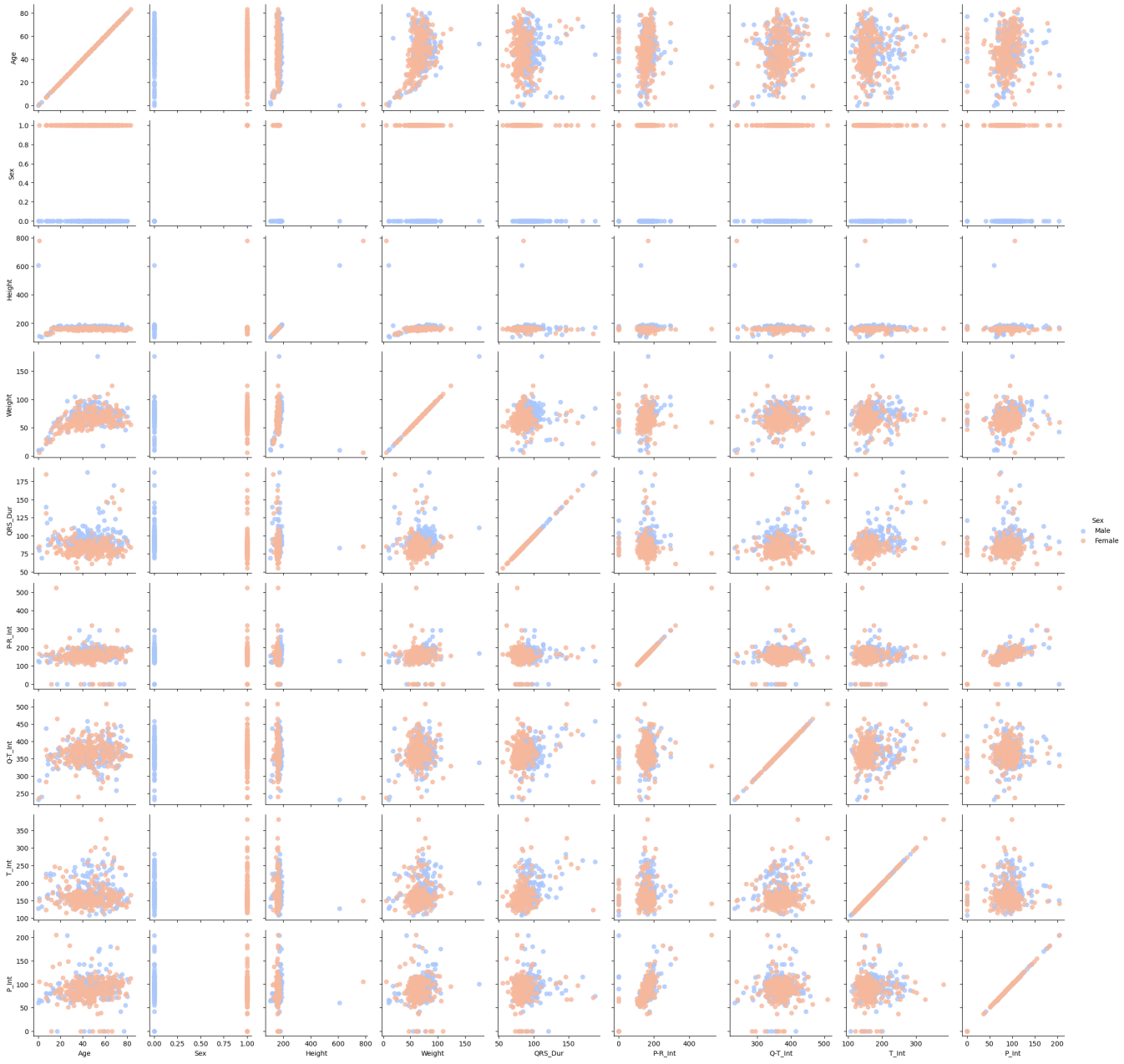
The distribution of classes within the dataset offers valuable insights. However, it's worth noting that the dataset exhibits bias, as half of its instances belong to the 'normal' class. So, the 'normal' class is excluded when the distribution is visualized.





Upon analyzing the pie chart and histogram, it becomes evident that 'Right Bundle Branch Block' and 'Ischemic Changes (CAD)' are the most prevalent classes among all. Conversely, classes such as '1st Degree Atrioventricular Block', '2nd Degree AV Block', and '3rd Degree AV Block' are entirely absent from the dataset.

Pairwise relationships and outliers in features: We will first use a pair plot to get an overview of the relationship between the features. As there are a 279 features in the dataset, we cannot visualize all of them in a pair plot, so let's take a subset of the features which are easily interpretable such as Age, Sex, Height, Weight, QRS Duration, P-R, Q-T, T and P intervals.



Each subplot in the grid shows the distribution of one variable on the x-axis and another variable on the y-axis. The diagonal plots along the main diagonal are uni-variate distributions of each variable.

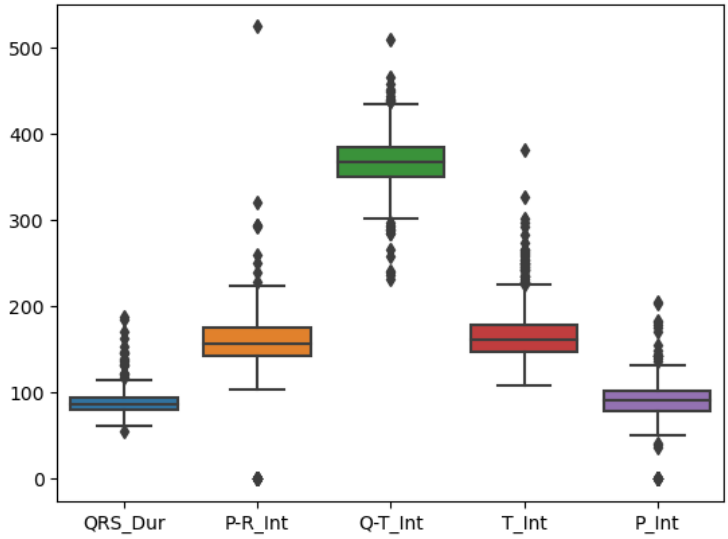
There is a positive correlation between height and weight. This is evident by the upward sloping trend in the scatter plot in the lower right corner of the grid.

But one of the main insights gained from the pairplot is that there are outliers in the data. There are 2 records with heights 780 and 608, this looks

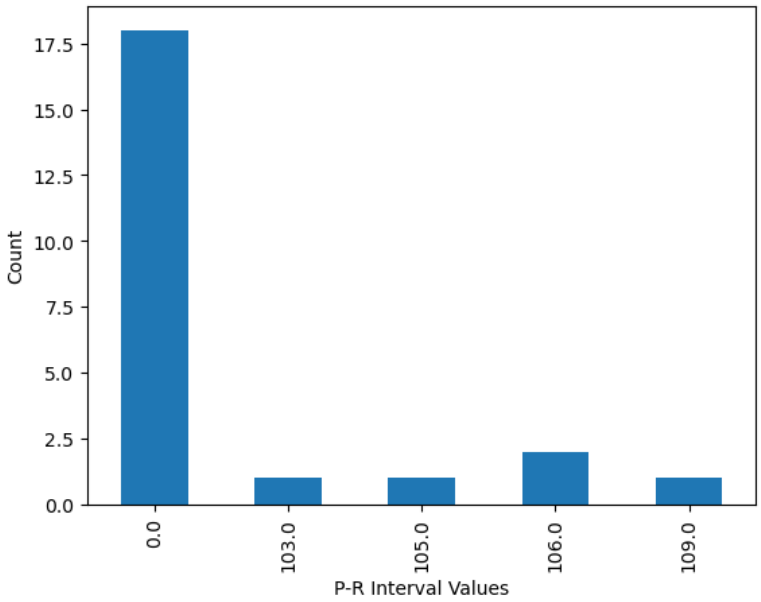
unnatural. The tallest person ever lived was 272cm. So, we can replace the heights 780 and 608 with 180 and 108 respectively.

The weight does not have any outliers. The maximum weight in the dataset is 176 kilograms, which is not impossible.

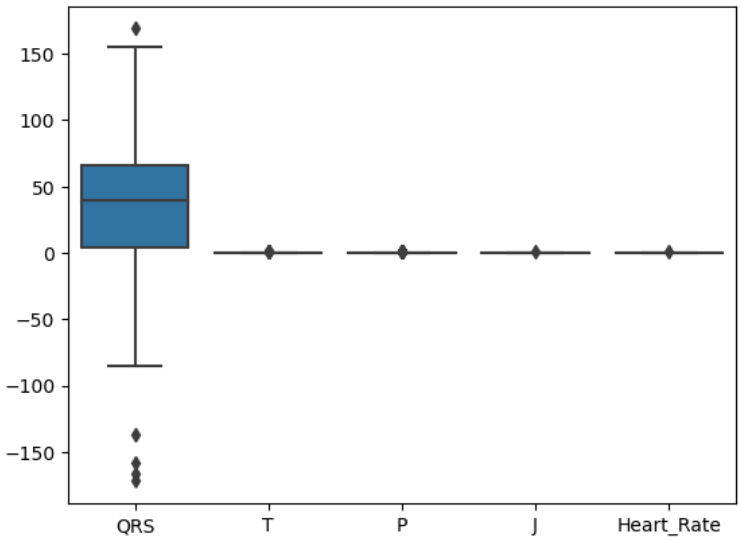
Visualizing rest of the variables using Boxplots:

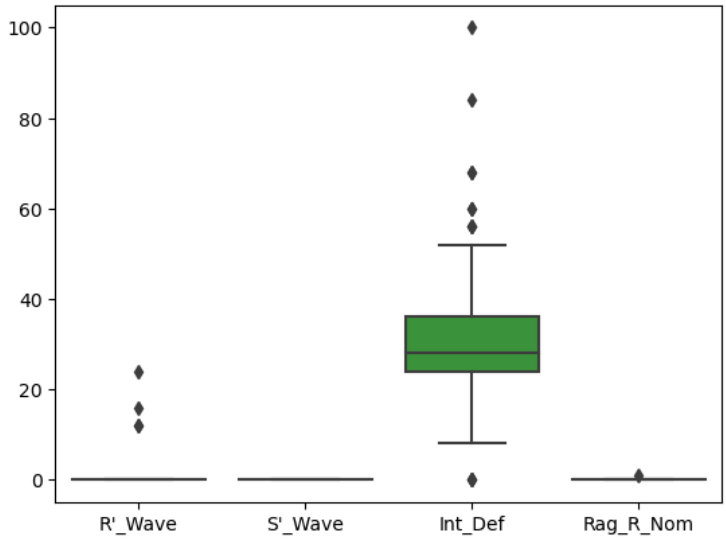


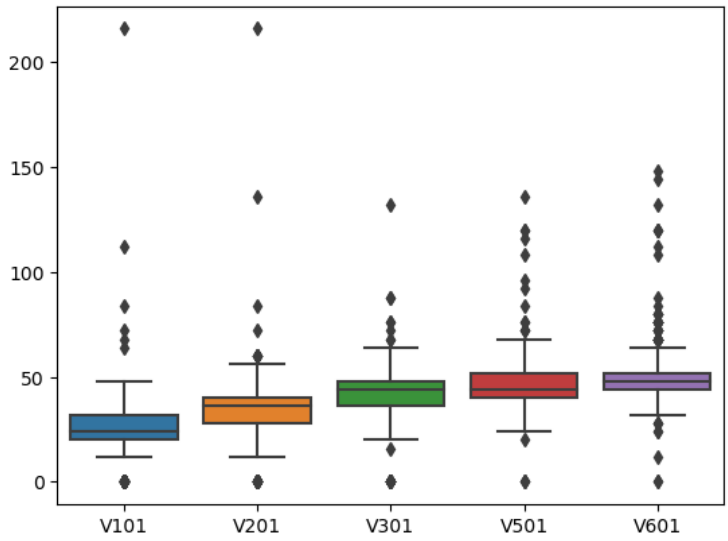
The P-R interval is the period measured in milliseconds, that extends from the beginning of the P-wave until the beginning of the QRS-complex. The normal range of the P-R interval is between 120 and 200 milliseconds. The people who are not in the normal range might have Arrhythmia.



Here we can observe that there are 18 records with P-R interval value as 0. These can be considered outliers due to human error or machine erorrs. But, we will keep them, because the number of records in the dataset is already less, and we cannot remove the column because it a key variable in the dataset.







V101 has an outlier, but when we look at other sets (V201, V301, V501) we can see that there's an outlier similarly. Since the data is heavily biased, We can't say these outliers should be dropped.

For example, when we look at our data, we can see that class number 8 (Supraventricular Premature Contraction) has only 2 instances. Or class number 3 (Ventricular Premature Contraction (PVC)) has only 3. The outliers appearing with our plots might belong to these instances and needs to be kept.

Data Splitting

In machine learning, it's crucial to divide your data into two sets: training and testing. The training set is used to train the model, while the testing set is used to evaluate its performance on unseen data.

This project used the hold-out method for this split. In this method, the data is randomly divided into two portions:

Training set (80%): This larger portion is used to train the machine learning model. The model learns patterns and relationships within the data to make predictions. Testing set (20%): This unseen data is used to assess the model's generalizability on new data. The model's performance on the testing set reflects how well it might perform on real-world data.

The training set, denoted by X-train and y-train, contained 361 data

points. This data was used to train the classification models.

The testing set, denoted by X-test and y-test, contained 91 data points. This unseen data was used to evaluate the performance of the trained models.

Feature Scaling

As the data contains both categorical and numerical data, we have to separately apply feature scaling methods. For the numerical data, we use StandardScalar to obtain 0 mean and unit variance. For the categorical variables, we usually use One Hot Encoding or Label Binirizer to convert the categorical variables into 1's and 0's. But in our case, the binary variables are already encoded as 0's and 1's. So, we just have to apply standard scaling to the numeric variables.

Principle Component Analysis

As our dataset contains a lot of features, applying Principal Component Analysis (PCA) can be beneficial. PCA identifies a smaller set of new features, called principal components, that capture most of the important information from the original features. This can be useful if the original features are redundant or highly correlated. By using these principal components instead of the original features, we can reduce the complexity of the model without losing significant data.

By using PCA, we have reduced the features from 278 to 121 with an explained variance of 98%. This indicates that the remaining features were irrelevent and redundant.

Model Implementation

1 Random Forest

Random Forest is a versatile ensemble learning algorithm renowned for its effectiveness in classification tasks, particularly with multi-class vari-

ables. Its strength lies in its ability to mitigate overfitting, thanks to the construction of multiple decision trees on random subsets of the data. By incorporating a voting mechanism for classification or averaging for regression, it synthesizes diverse predictions, reducing variance and enhancing model reliability. Moreover, its resilience to high-dimensional data makes it a robust choice for complex datasets, as it can efficiently handle numerous input variables without sacrificing performance. Furthermore, Random Forest implicitly conducts feature selection by considering random subsets of features at each split, accentuating the importance of informative attributes. Renowned for its robustness, high accuracy, and adaptability across various domains, Random Forest stands as a dependable tool for tackling multi-class classification challenges.

2 Support Vector Machines

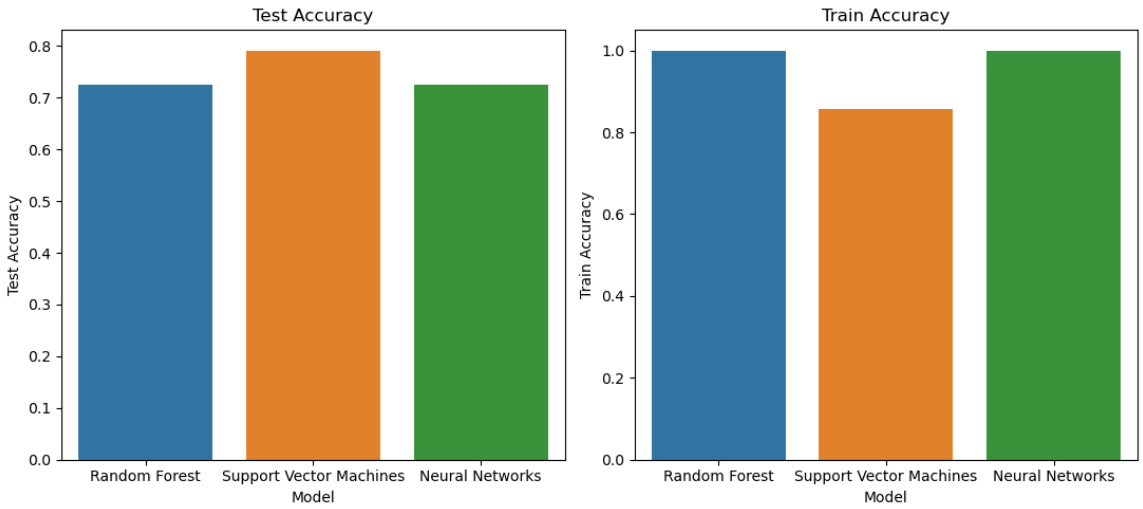
Support Vector Machines (SVMs) are a powerful and widely used supervised learning algorithm, particularly favored for classification tasks. SVMs aim to find the optimal hyperplane that separates data points into different classes while maximizing the margin between the classes. This hyperplane is the one that has the maximum distance to the nearest data points of any class, making it robust to noise and outliers. Additionally, SVMs can handle both linearly separable and non-linearly separable data by using a technique called the kernel trick. This involves transforming the input space into a higher-dimensional space where the data becomes linearly separable, allowing SVMs to find complex decision boundaries. SVMs are known for their ability to generalize well to unseen data, making them suitable for various applications, including text classification, image recognition, and bioinformatics. Furthermore, SVMs have a regularization parameter that helps control overfitting, making them robust even in cases of small training datasets. Overall, SVMs offer a versatile and effective solution for classification tasks, providing strong performance and robustness across different domains and datasets.

3 Neural Networks

Neural Networks are a fundamental concept in machine learning, renowned for their ability to model complex relationships and patterns in data. In classification tasks, neural networks work by learning the mapping between input features and output classes through a series of interconnected layers of neurons. These layers typically include an input layer, one or more hidden layers, and an output layer. Each neuron in a layer receives input from the neurons in the previous layer, applies a transformation function (such as the sigmoid or ReLU function), and passes the result to the neurons in the next layer. During training, neural networks adjust the weights and biases of connections between neurons using optimization algorithms like gradient descent, minimizing a loss function that measures the difference between predicted and actual outputs. This process of forward propagation and backward propagation of errors allows neural networks to iteratively learn to classify data accurately. One of the key advantages of neural networks is their ability to automatically learn hierarchical features from raw data, eliminating the need for manual feature engineering. Moreover, deep neural networks with multiple hidden layers, known as deep learning models, have demonstrated remarkable performance in various classification tasks, including image recognition, natural language processing, and speech recognition. However, training deep neural networks often requires large amounts of labeled data and computational resources. Despite these challenges, neural networks remain a powerful tool for classification, offering unparalleled flexibility and scalability in modeling complex relationships.

BaseLine Evaluation of the Models

The Random Forest, Support Vector Machines and Neural Networks are first trained without tuning the hyperparameters. The results are as follows:



It can be observed that the Random Forest and Neural Network produce 100% accurate results with Training data, but they have lower accuracy with test data. On the other hand, SVM has lower accuracy with training data, but has comparatively better accuracy with test data.

Hyperparameter Tuning

After implementing the models, now we perform hyper-parameter tuning to enhance the accuracy and performance of the model. We use GridSearchCV to iterate the search space and train the model in an iterative way.

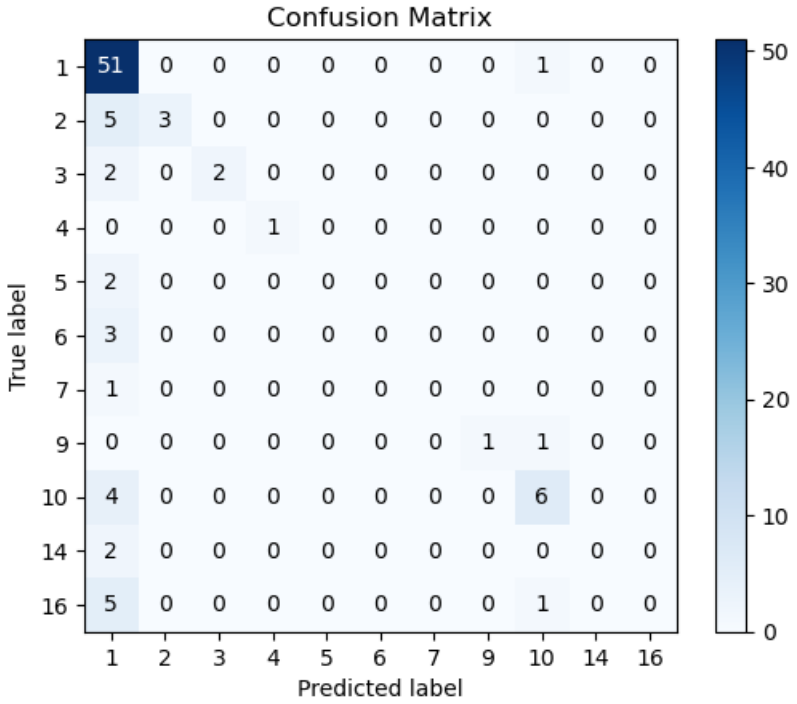
Evaluation Metrics

Random Forest

The overall accuracy of the model is 0.7, which means that the model has correctly classified 70% of the data. Class 1 has relatively high precision and recall, indicating that it's well-predicted by the model. Class 2 has high precision but low recall, meaning that when the model predicts class 2, it's usually correct, but it misses many actual class 2 instances. Class 3 also has high precision but low recall. Class 4 has perfect precision and recall, but it has only one instance in the dataset. Classes 5, 6, 7, 14, and 16 have poor

performance with zero precision, recall, and F1-score. Classes 9 and 10 have high precision but relatively low recall. The weighted average F1-score is 0.63, indicating moderate overall performance. The overall accuracy of the model is 70%, which means that 70In summary, while the model performs well for some classes it struggles with others, especially those with fewer instances.

	Model	Accuracy	Recall	Precision	F-1 Score
0	Random Forest	0.703297	0.703297	0.626667	0.631244
1	SVM	0.604396	0.604396	0.648707	0.606655
2	Neural Networks	0.725275	0.725275	0.709095	0.704674

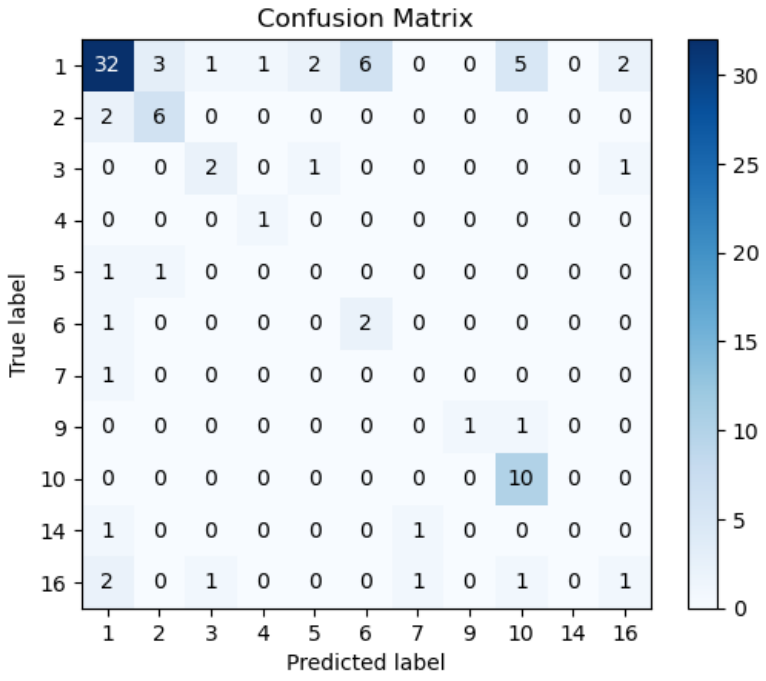


Support Vector machines

The evaluation of Support Vector Machines (SVM) performance reveals a mixed outcome across different classes. While classes such as 1, 2, and 3 exhibit moderate precision and recall, indicating a reasonably accurate classification, others like classes 4, 6, and 16 show varying levels of deficiency, with low precision and/or recall. Notably, class 4 demonstrates a paradoxical scenario of low precision but perfect recall, implying a propensity to

correctly identify all instances of the class while also misclassifying other instances. Additionally, classes 5, 7, and 14 have negligible precision, recall, and F1-score, suggesting significant misclassification or an inability to detect instances of these classes altogether. However, classes 9 and 10 perform relatively well with high precision and recall. Overall, the model achieves a weighted average F1-score of 0.61, indicating moderate performance, with an accuracy of 60%, reflecting the proportion of correctly predicted instances.

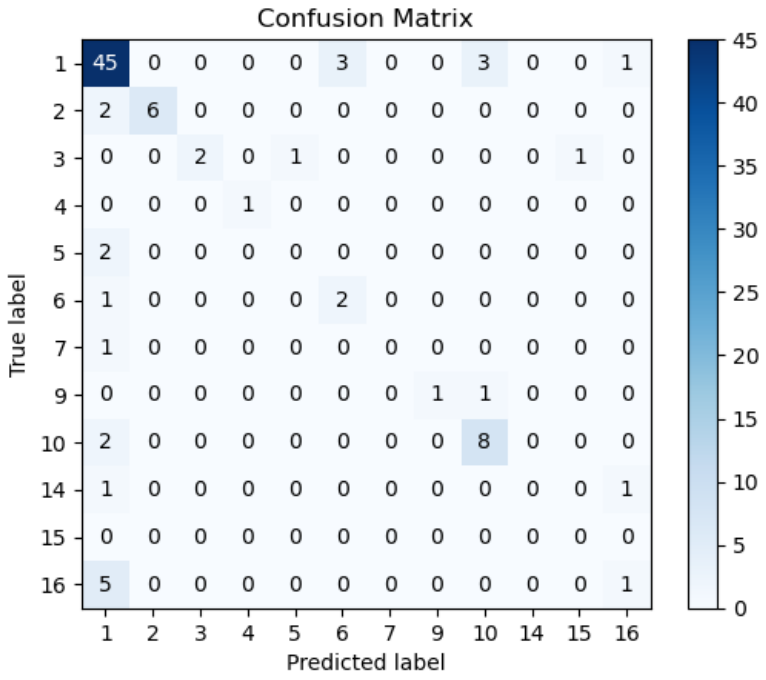
	precision	recall	f1-score	support
1	0.80	0.62	0.70	52
2	0.60	0.75	0.67	8
3	0.50	0.50	0.50	4
4	0.50	1.00	0.67	1
5	0.00	0.00	0.00	2
6	0.25	0.67	0.36	3
7	0.00	0.00	0.00	1
9	1.00	0.50	0.67	2
10	0.59	1.00	0.74	10
14	0.00	0.00	0.00	2
16	0.25	0.17	0.20	6
accuracy			0.60	91
macro avg	0.41	0.47	0.41	91
weighted avg	0.65	0.60	0.61	91



Neural Networks

The assessment of neural networks' performance based on the metrics reveals a diverse performance across different classes. While classes 1, 2, and 10 demonstrate relatively high precision and recall, indicating accurate classification, others like classes 3, 6, and 16 display varying levels of deficiency, with lower precision and/or recall. Notably, class 4 achieves perfect precision and recall, suggesting precise identification of all instances. However, classes 5, 7, 9, and 14 exhibit negligible precision, recall, and F1-scores, indicating significant misclassification or an inability to detect instances of these classes effectively. The absence of instances for class 15 precludes its evaluation. Overall, the neural network model attains a weighted average F1-score of 0.70 and an accuracy of 73%, indicating an effective performance.

	precision	recall	f1-score	support
1	0.76	0.87	0.81	52
2	1.00	0.75	0.86	8
3	1.00	0.50	0.67	4
4	1.00	1.00	1.00	1
5	0.00	0.00	0.00	2
6	0.40	0.67	0.50	3
7	0.00	0.00	0.00	1
9	1.00	0.50	0.67	2
10	0.67	0.80	0.73	10
14	0.00	0.00	0.00	2
15	0.00	0.00	0.00	0
16	0.33	0.17	0.22	6
accuracy			0.73	91
macro avg	0.51	0.44	0.45	91
weighted avg	0.71	0.73	0.70	91

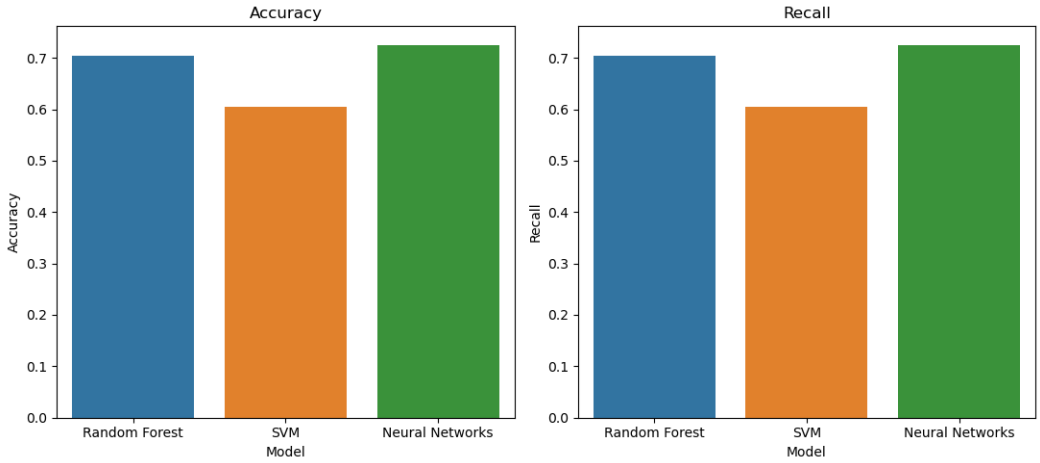


Model Evaluation and Comparative Analysis

The performance of the selected models are compared based on various performace metrics as shoun in the table below:

	Model	Accuracy	Recall	Precision	F-1 Score
0	Random Forest	0.703297	0.703297	0.626667	0.631244
1	SVM	0.604396	0.604396	0.648707	0.606655
2	Neural Networks	0.725275	0.725275	0.709095	0.704674

After performing Hyperparameter tuning, the performance of the models seem to go down. This can happen due to a limited search space or due to less training data. In this particular case, it is a fact that the number of records are less when compared to the number of features, but something that is much more important to notice is that the number of classes is more too.



But, we can say that SVM performed better than Neural Networks and Random Forest before performing hyper parameter tuning. But, after hyper parameter tuning, Random forest and neural networks seem to perform better with an accuracy above 70%.

In conclusion we can say that Neural Networks perform best in this scenario with an excellent accuracy of 72%.

References

- Guvenir,H., Acar,Burak, Muderrisoglu,Haldun, and Quinlan,R.. (1998). Arrhythmia. UCI Machine Learning Repository. https://doi.org/10.24432/C5BS32.