
# MANIPAL INSTITUTE OF TECHNOLOGY

(A constituent Institute of MANIPAL UNIVERSITY)

**MANIPAL - 576 104, KARNATAKA, INDIA**


## *Prediction of Cardiac Arrhythmia*

***Submitted By-***

**Varun Kathuria-140905124**

**Prakhar Thapliyal-140905468**

January 2017 – May 2017

# Prediction and Classification of Cardiac Arrhythmia

Varun Kathuria, Prakhar Thapliyal
Dept. of Computer Science, Manipal Institiute of Technology
CSE 4010 Project Report

## I. ABSTRACT

Cardiac Arrhythmia is a of a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal. It is the leading cause of death for both men and women in the world. In this project, we plan to predict Cardiac Arrhythmia based on a patient's medical record. Our objective is to classify a patient into one of the Arrhythmia classes like Tachycardia and Bradycardia based on his ECG measurements and help us in understanding the application of machine learning in medical domain. After appropriate feature selection we plan to solve this problem by using Machine Learning Algorithms namely K Nearest Neighbour, Logistic Regression, Naïve Bayes and SVM .

## II. INTRODUCTION

The total number of deaths due to cardiovascular diseases read 17.3 million a year according to the WHO causes of death. Thus, how to predict cardiac arrhythmia in real life is of great significance. In this project, we plan to develop a machine learning system that can classify a patient into different cardiac arrhythmic classes. The diagnosis of cardiac arrhythmia can be classified into various classes based on the *Electrocardiogram(ECG)* readings and other attributes. First class will refer to the normal patient while other classes shall represent different classes of cardiac arrhythmia like Tachycardia, Bradycardia and Coronary artery diseases. This is a supervised learning problem.

## III. DATA SET

The dataset for the project is taken from the UCI Repositoryhttps://archive.ics.uci.edu/ml/datasets/Arrhythmia There are (452) rows, each representing medical record of a different patient. There are 279 attributes like age, weight and patient's ECG related data. General attributes like age and weight have discrete integral values while other ECG features like QRS duration have real values.

The variable Class is our target variable. There are in total 13 classes –

| No. | CLASS | INSTANCES |
|---|---|---|
| 1 | Normal | 245 |
| 2 | Ischemic changes (Coronary Artery) | 44 |
| 3 | Old Anterior Myocardial Infarction | 15 |
| 4 | Old Inferior Myocardial Infarction | 15 |
| 5 | Sinus tachycardia | 13 |
| 6 | Sinus bradycardia | 25 |
| 7 | Ventricular Premature Contraction (PVC) | 3 |
| 8 | Supraventricular Premature Contraction | 2 |
| 9 | Left bundle branch block | 9 |
| 10 | Right bundle branch block | 50 |
| 11 | Left ventricle hypertrophy | 4 |
| 12 | Atrial Fibrillation or Flutter | 5 |
| 13 | Others | 22 |

## IV. SURVEY

The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 13 groups. For the time being, there exists a computer program that makes such a classification. However there are differences between the cardiolog's and the programs classification. Taking the cardiolog's as a gold standard we aim to minimize this difference by means of machine learning tools.

## V. SCOPE

These machine learning techniques can be deployed in hospitals where a large dataset is available and can help the doctors in making more precise decisions and to cut down the number of causalities due to heart diseases in the future.

## VI. METHODOLOGY

### A. Feature Selection:

Firstly, we removed some of the categorical features that were 95% of time indicating either all 0's or all 1's. If any training instance has a missing value for a given attribute, we set it as the mean of the value plus or minus the standard deviation for that attribute related to the class it belongs to.

If for a given attribute majority of values are missing, then we discard that attribute and remove it from our training set.

The features can be grouped into 5 blocks –

- features concerning biographical characteristics, i.e., age, sex, height, weight and heart rate.
- features concerning average wave durations of each interval (PR interval, QRS complex, and ST intervals).
- features concerning vector angles of each wave.
- features concerning widths of each wave.
- features concerning amplitudes of each wave.

### B. Random Forests and Decision Trees:

We implement a *Random Forest* classifier. The model works by continually sampling with replacement a portion of the training dataset, and fitting a decision tree to it. The number of trees refer to the number of times the dataset is randomly sampled. Moreover, in each sampling iteration, a random set of features are selected. In decision trees, each node refers to one of the input variables, which has edges to children for all possible values that the input can take. Each leaf corresponds to a value of the class label given the values of the input variables represented by the path from the root node to the leaf node. The number of trees and the number of leaves are learned via cross validation.

### C. Principal Component Analysis:

PCA is being used to identify patterns in the data and then expressing the data in such a way to highlight similarities and differences. Primarily we are using PCA to reduce the number of dimensions by identifying the more important features i.e. the principal components. The number of principal components is less than or equal to the smaller of the number of original variables. The first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

# VII. MODELS

## A. KNN (K-Nearest Neighbours):

$$D(a,b) = \sqrt{\sum_{i}^{n}(b_i - a_i)^2}$$

We used KNN because it is simple to implement & very straight forward. Here, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. This is done by measuring distances between the object and its neighbors. The following formula shows a representation of simple Euclidian distance, where 'a' and 'b' are the respective positions of the object and one of its neighbours. KNN is very sensitive to irrelevant or redundant features because all features contribute to the similarity and thus to the classification. This was improve by careful feature selection described previously. The results are summarized below –

| Training-Testing Size | K Neighbours | Training Accuracy | Test Accuracy |
|---|---|---|---|
| 80%-20% | 6 | 67 % | 61 % |
| 70%-30% | 6 | 65 % | 62 % |

Table 1: KNN Classification with PCA

| Training-Testing Size | K Neighbours | Training Accuracy | Test Accuracy |
|---|---|---|---|
| 80%-20% | 6 | 65 % | 64 % |
| 70%-30% | 6 | 67 % | 62 % |

Table 2: KNN Classification with RF

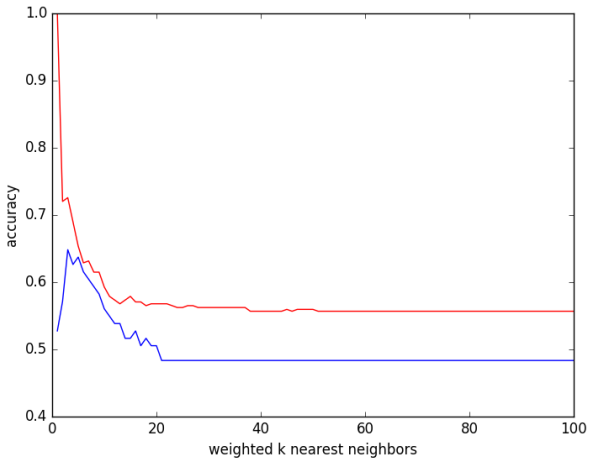


Image 1: KNN accuracy training vs testing

## B. Logistic Regression:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}(1-y^{(i)})\log(1-h_\theta(x^{(i)})) + y^{(i)}\log h_\theta(x^{(i)})\right]$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{1} 1\left\{y^{(i)}=j\right\}\log p(y^{(i)}=j|x^{(i)};\theta)\right]$$

Since the logistic regression is used for binary classification of datasets with categorical dependent features, in order to apply logistic regression to our multi-class dataset, we firstly classified our instances into two major classes, class 1 (which contained all the instances with "class 01" label) and class NOT-1 (which contained the instances for all the other classes). We classified our data in this way, because about half of our instances were labeled as class 01. The results of our implementation are summarized below-

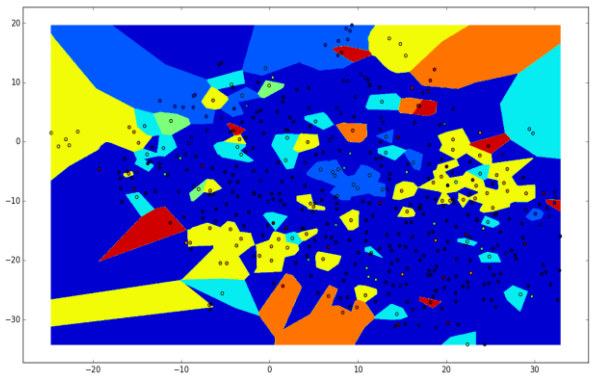

Image 2: Logistic regression classification

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 90 % | 74 % |
| 70%-30% | 89 % | 71 % |

Table 3: Logistic Regression with PCA

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 96 % | 73 % |
| 70%-30% | 98 % | 72 % |

Table 4: Logistic Regression with RF

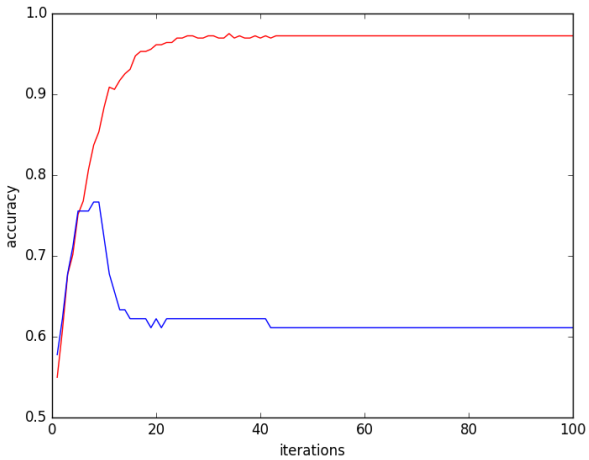


Image 3: Logistic regression training vs testing

## C. Naïve – Bayes Classifier:

We assume that all $x_j|y = k$ are independent. Then using formulas of conditional probability, we have the probability of the whole training set:

$$P(x, y) = \prod_{i=1}^{m} \left( \prod_{j=1}^{n} \phi_{j,x_j^{(i)}|y=y^{(i)}} \right) \phi_{y^{(i)}} \qquad (5)$$

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors.We implemented our own Naive Bayes binomial and multinomial classifiers in Python. We use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction. In the first case, the training-testing data was split 80% - 20% and in the second case, the training-testing data was split 80% - 20%. The results are summarised below –

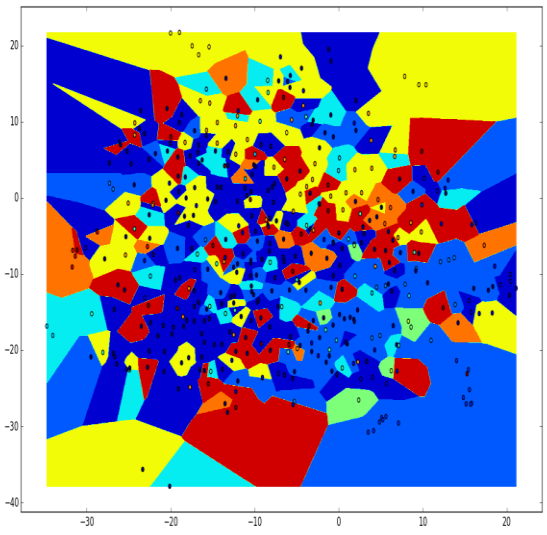

Image 4: Naïve Bayes Classification

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 79 % | 62 % |
| 70%-30% | 78 % | 65 % |

Table 5: Naïve-Bayes with PCA

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 75 % | 63 % |
| 70%-30% | 74 % | 62 % |

Table 6: Naïve-Bayes with RF

### D. SVM (Support Vector Machines ) :

In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line. You can make classifications using this line. By plugging in input values into the line equation, we calculate whether a new point is above or below the line. We tried both the polynomial and the linear kernels for the SVM and found out that the linear kernel outperformed the polynomial kernel.

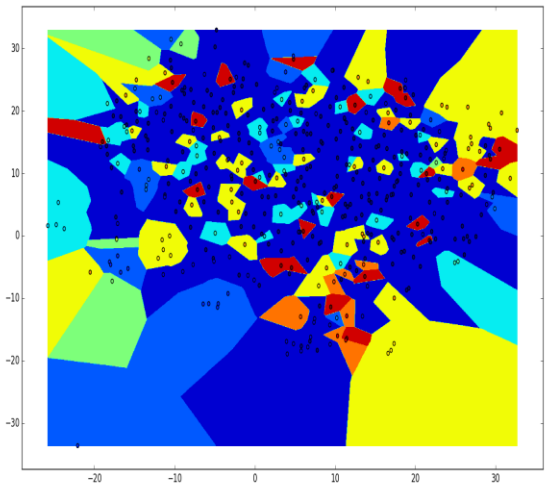

Image 5: SVM Classification

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 99 % | 75 % |
| 70%-30% | 99 % | 71 % |

Table 7: SVM with PCA

| Training-Testing Size | Training Accuracy | Test Accuracy |
|---|---|---|
| 80%-20% | 100 % | 71 % |
| 70%-30% | 99 % | 70 % |

Table 8: SVM with RF

### E. Weighted KNN:

A refinement of the KNN classification algorithm is to weigh the contribution of each of the K neighbours according to their distance to the query point, giving greater weight to closer neighbours. The results are summarized below-

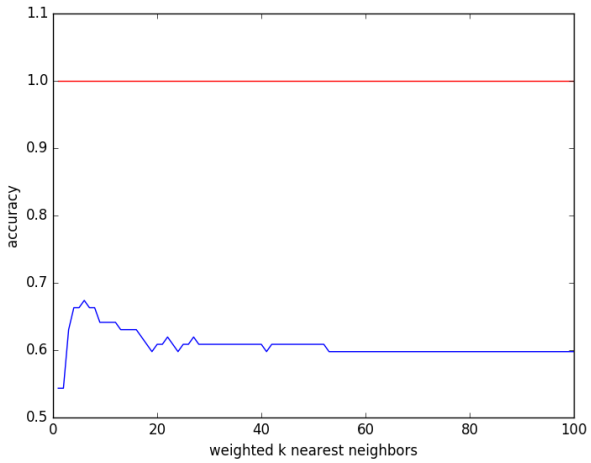


Image 6: Weighted KNN accuracy training vs testing

| Training-Testing Size | K Neighbours | Training Accuracy | Test Accuracy |
|---|---|---|---|
| 80%-20% | 6 | 99 % | 65 % |
| 70%-30% | 6 | 99 % | 64 % |

Table 9: Weighted KNN Classification with PCA

| Training-Testing Size | K Neighbours | Training Accuracy | Test Accuracy |
|---|---|---|---|
| 80%-20% | 6 | 99 % | 70 % |
| 70%-30% | 6 | 99 % | 66 % |

Table 10: Weighted KNN Classification with RF

## VIII. RESULTS

The main objective of this project was to develop a system that could robustly detect an arrhythmia. The second objective of this project was to develop a method to robustly classify an ECG trace into one of 13 broad arrhythmia classes. We report our performance for each of the five methods using two different methodologies. We show results for each algorithm, as well as vary other parameters for better results.

## IX. ANALYSIS

It is clear from the above data that the SVM and Logistic Regression algorithms are capable of automatically detecting arrhythmias with reliable accuracy (**Training Data = 98 % and Testing Data=73%**). Furthermore, Random Forests consistently performs better than PCA in terms of feature selection. Our general approach in this project was as follows. We started with KNN and we tried to obtain maximum accuracy for different values of K ranging from 3 to 13. Then we used Logistic Regression which uses the sigmoid function and we ran it using Gradient descent and Newton's method. Logistic regression gave comparatively better results with average accuracy around 73 %. Naïve-Bayes classifier gave poor results due to problem of lack of enough training examples (452) and excessive number of features. SVM using linear kernels gave the best results with average accuracy of classification around 99 % for training set and 73 % for testing set.

## X. ACKNOWLEDGEMENT

We are highly grateful to our professors Dr.Srikanth Prabhu and Muralikrishna SN for their continued guidance and support throughout the course of this project.

## XI. REFERENCES

[1].http://en.wikipedia.org/wiki/Cardiac_dysrhythmia

[2].http://www.cdc.gov/dhdsp/data_statistics/fact_sheets/docs/fs_heart_disease.pdf

[3].Cunningham 2007. k-Nearest Neighbor Classifiers. Technical Report UCD-CSI-2007-4. University College Dublin

[4] Roger VL et al. Heart disease and stroke statistics—2012 update: a report from the American Heart

[5] http://www.texasheart.org/HIC/Topics/Cond/bbblock.cfm

[6] http://archive.ics.uci.edu/ml/

[7] UCI machine learning repository (2013), http://archive.ics.uci.edu/ml