MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL, INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Session 2018-19

Minor Project Report
On

Prediction and Classification of Cardiac Arrhythmia

Under the Guidance of

PROF DHIRENDRA PRATAP SINGH

SUBMITTED BY:

SARTHAK SARASWAT	SCH NO – 161112003
SUDHANSHU RANJAN	SCH NO– 161112046
PRAVEEN TIWARI	SCH NO – 161112084
RAJENDRA SISODIYA	SCH NO – 161112094

CERTIFICATE

This is to certify that the project report carried out on “**Prediction and Classification of Cardiac Arrhythmia**” by the 3rd year students:

SARTHAK SARASWAT	SCH NO – 161112003
SUDHANSHU RANJAN	SCH NO– 161112046
PRAVEEN TIWARI	SCH NO – 161112084
RAJENDRA SISODIYA	SCH NO – 161112094

Have successfully completed their project in partial fulfillment of their Degree in Bachelor of Technology in Computer Science and Engineering.

PROF DHIRENDRA PRATAP SINGH

(Minor Project Mentor)

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project Documentation Entitled as “**Prediction and Classification of Cardiac Arrhythmia**” is an authentic documentation of our own original work and to best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal or elsewhere, is explicitly acknowledged in the report.

SARTHAK SARASWAT	SCH NO – 161112003
SUDHANSHU RANJAN	SCH NO– 161112046
PRAVEEN TIWARI	SCH NO – 161112084
RAJENDRA SISODIYA	SCH NO – 161112094

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide **PROF DHIRENDRA PRATAP SINGH** for his valuable help and guidance. We are thankful for the encouragement that he has given us in undertaking this project. His rigorous evaluation and constructive criticism is of great assistance.

We are also grateful to our respected director for permitting us to utilize all the necessary facilities of the college.

Needless to mention is the additional help and support extended by our respected HOD, Dr. Meenu Chawla, in allowing us to use the departmental laboratories and other services.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind co-operation and help.

Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing the much needed support and encouragement.

CONTENTS

LIST OF FIGURES

ABSTRACT

Rapid advancements in technology have facilitated early diagnosis of diseases in the medical sector. One of the most prevalent medical conditions that demands early diagnosis is cardiac arrhythmia. ECG signals can be used to classify and detect the type of cardiac arrhythmia. This paper introduces a novel approach to classify the ECG data into one of the sixteen types of arrhythmia using Machine Learning. The proposed method uses the UCI Machine Learning Repository dataset of cardiac arrhythmia to train the system on 279 different attributes. In order to increase the accuracy, the method uses Principal Component Analysis for dimensionality reduction, Bag of Visual Words approach for clustering and compares different classification algorithms like Support Vector Machine, Random Forest, Logistic Regression and K-Nearest Neighbor algorithms, thus choosing the most accurate algorithm, Support Vector Machine.

INTRODUCTION

In India, a death is recorded every 33 seconds due to heart attack. In the past few decades, coronary heart disease, hypertension and other cardiovascular disease have become a global threat to human life. In our country, this phenomenon is getting increasingly severe due to the aging of population, living environment and unhealthy food consumption.

ECG provides the information which is needed to identify the problems and hence it becomes important when developing an advanced diagnostic system.

Objective

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 and compared the results in order to get the suitable algorithm for correctly predicting the class cardiac arrhythmia.

About Model

In this paper, the aim is to develop a hybrid model which uses various machine learning techniques like principal component analysis, Bag of Words model and various classification algorithms. Using this model, it is possible to classify an ECG signal to one of the 16 classes of arrhythmia, where class 1 means normal ECG signal, classes 2 to 15 are different types of arrhythmia and class 16 refers to the rest of unclassified ones. The use of machine learning will help in greater accuracy and high potential to detect severe cardiac arrhythmia possibilities.

LITERATURE SURVEY

Our proposed model makes use of this concept in cardiology—

A. Performance Analysis of Artificial Neural Networks for

Cardiac Arrhythmia Detection

The paper takes in an ECG signal and converts the analog signal to a digital signal. The system has extracted 8 beats from each ECG signal sampled at 2223 samples per second and classified these beats. The next step was signal pre-processing which was denoising of loaded raw ECG signal. The system then extracts just three features from the signal; QRS complex duration, RR interval both normal and the one averaged over 8 beats. These features were further used by ANN classifiers such as Naive Bayes and Multi-class SVM to predict the class of the arrhythmia. The results were compared and the accuracy of each of the algorithm is calculated.

B. Identifying Best Feature Subset For Cardiac Arrhythmia Classification

This paper presents a model which is divided into two parts - filter part and wrapper part. The filter part deals with feature selection from the cardiac arrhythmia dataset of the UCI machine learning repository. These help in identifying the best features without taking any assistance of a classification algorithm, but rather, just using a set of presumed criteria.

The feature selection model presented makes use of both, filter and wrapper techniques of feature selection. For judging the relative importance of each feature, an improved F-score is calculated for each and every feature, which produces a superset of features that can be used. Sequential Forward Search is then used for finding the final subset of most important features. Following this, SVM and KNN are used for classification of cardiac arrhythmia using the new list of features

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	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

TABLE:1 CLASSES OF CARDIAC ARRYTHYMIA

SCOPE

The UCI dataset used consisted of already extracted features. The future scope of this paper involves directly extracting features from an ECG signal. Apart from the 16 possible classes that were taken in consideration in this paper, the ECG signal can be classified into a different set of cardiac arrhythmias.

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.

METHODOLOGY AND WORK DESCRIPTION

Feature Selection:

From the dataset, out of the 279 features present, it was infeasible to extract all the features. This is because many features used some information that is not accessible to the doctors while analysing ECG reports of patient. Hence, the dataset was narrowed with the help of Principal Component Analysis (PCA).

Principal component analysis

Principal component analysis is a method of extracting variables that influence the final decision the most and provide as much as information as possible. The aim of PCA in this paper is to reduce the dataset containing large amount of dimensions and find out features with low dimensions. A principal component is a combination of the normalized linear original predictors in a dataset.

Let us assume a predictor set as:

$Y^1, Y^2, ... Yn$

The principal component can be written as:

$Z^1 = \Phi^{11}Y^1 + \Phi^{21}Y^2 + \Phi^{31}Y^3 + ... + \Phi n^1 Yn$

Z^1 is first principal component Φn^1 is the loading vector that comprises of loadings (Φ^1,Φ^2..) of first principal component. The loadings are restricted to a unit sum of square. The reason being that large variance can be caused due to high magnitude of the loadings. Φn^1 defines the direction of the principal component (Z^1) along which maximum variance of the data is observed. It gives rise to a line in n dimensional space which is in close proximity to the m observations. Average squared Euclidean distance is used to measure the closeness.

X^1...Xn are normalized predictors; that have zero mean and unit standard deviation.

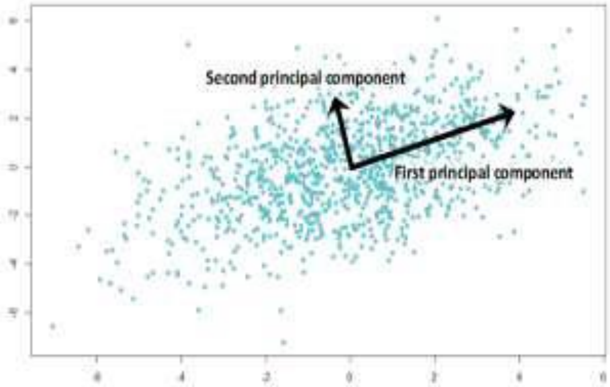


Fig. 2. *The first two principal components in two dimensional graph*

The variability captured by the first component is directly proportional to the information captured by that component.

The first principal component results in a line which is nearest to the data i.e. the minimum sum of squared distance between a data point and the line. The first principal component outputs a line which is nearest to the data i.e. the minimum value obtained by summing the squared distance between a data point and the line.

A scree plot is developed to find factors which capture most of the data variability. The values are represented in decreasing order. By plotting a cumulative variance plot, we get a further clearer picture of the number of components required.

The plot in Fig. shows 150 components depicting around 99% variance in the dataset. Therefore, using PCA the 279 predictors were reduced to 150 with the same explained variance.

MODELS

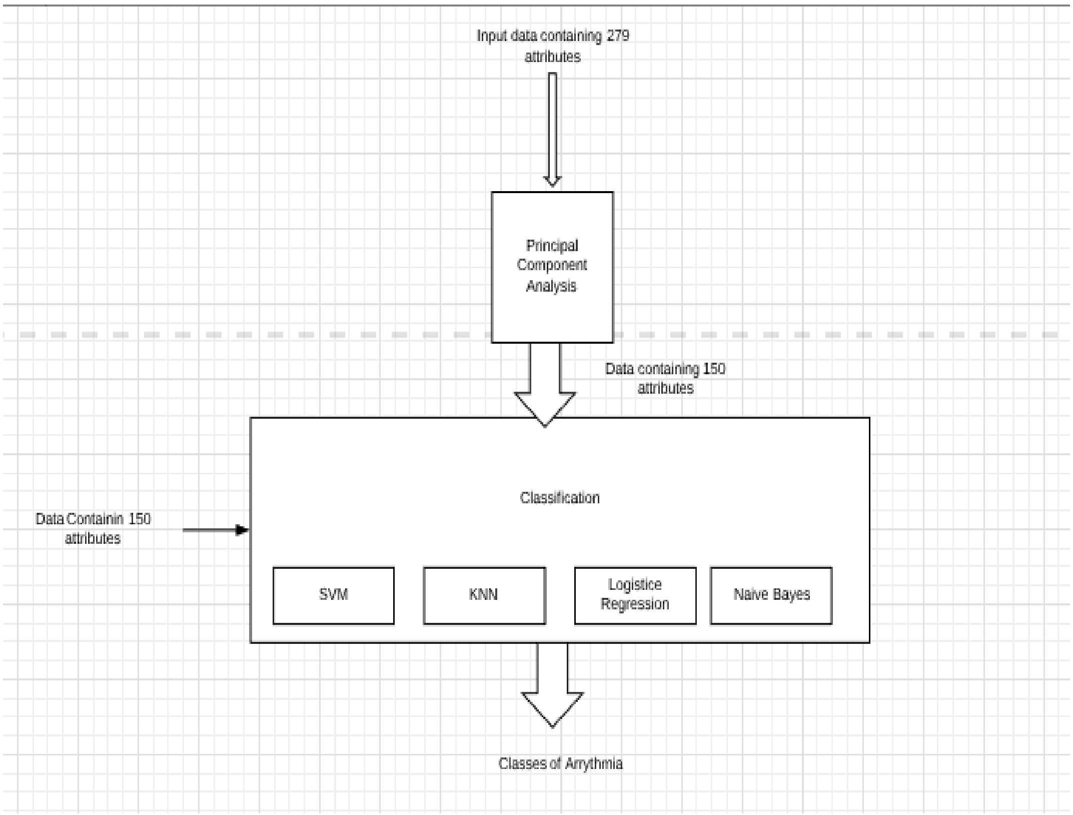
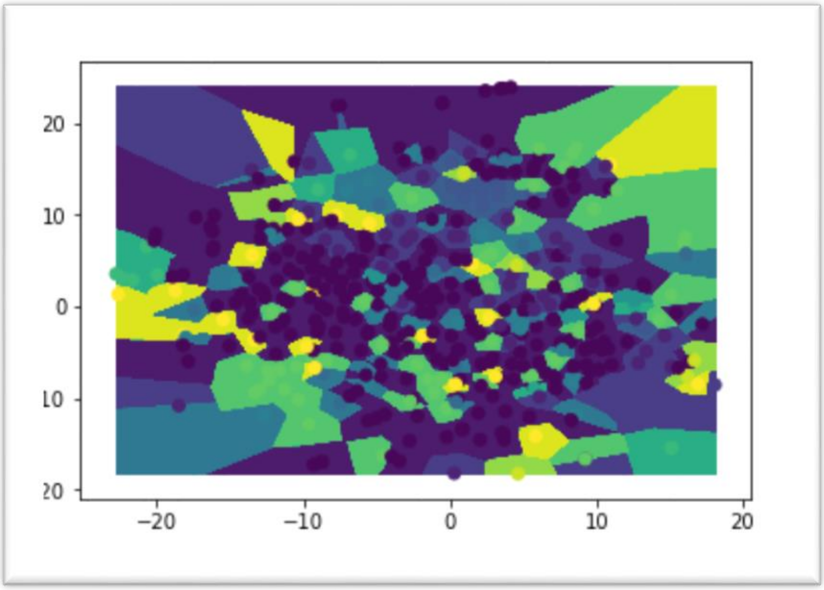

Input data containing 279 attributes
Principal Component Analysis
Data containing 150 attributes
Data Containin 150 attributes
Classification
SVM
KNN
Logistice Regression
Naive Bayes
Classes of Arrythmia

FIGURE 3: MODELS AND ALGORITHM USED

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 neighbours, with the object being assigned to the class most common among its k nearest neighbours. This is done by measuring distances between the object and its neighbours. 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
70%-30%	6	100 %	55.14 %

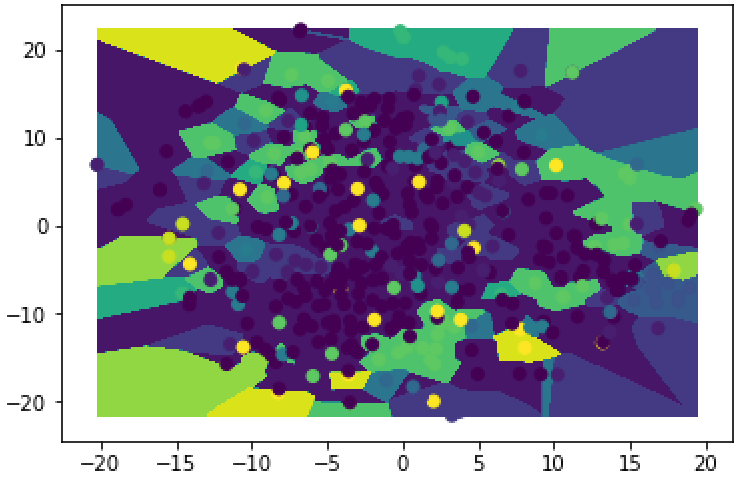
FIGURE 4: KNN Classification with PCA

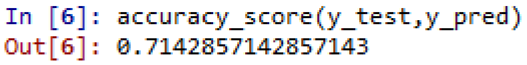
B. Logistic Regression:

$$
\begin{aligned}
J(\theta) &= -\frac{1}{m}\left[\sum_{i=1}^{m}(1-y^{(i)})\log(1-h_\theta(x^{(i)}))+y^{(i)}\log h_\theta(x^{(i)})\right]\\
&= -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{1}1\left\{y^{(i)}=j\right\}\log p(y^{(i)}=j|x^{(i)};\theta)\right]
\end{aligned}
$$

Logistic regression hypothesis gives the output as a estimated probability .A threshold value is set to and based upon a this threshold a estimated probability can be classified into class. For ex. let us threshold value is 0.5 then a 0.6 estimated probability then input is considered as in class 1 whereas a input with estimated probability 0.3 is considered as in class 0.

Logistic regression hypothesis uses a sigmoid function. We need to maximize the probability by minimizing loss function .Decreasing the cost will increase the maximum likelihood. Values of Coefficients (beta) that minimize the error in the probabilities predicted by the model to those in data.





```
In [6]: accuracy_score(y_test,y_pred)
Out[6]: 0.7142857142857143
```

Training-Testing Size	Training Accuracy	Test Accuracy
70%-30%	88.92 %	72 %

FIGURE 5: LOGISTIC REGRESSION Classification with PCA

C. Naïve – Bayes Classifier

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

In Naive Bayes algorithm we assumes that predictors are independent and uses the bays theorem for classification purpose. We calculate posterior probability and a class with highest posterior probability is outcomes.

Here

$$P(c|x)=(P(x|c)P(c))/P(x)$$

Where

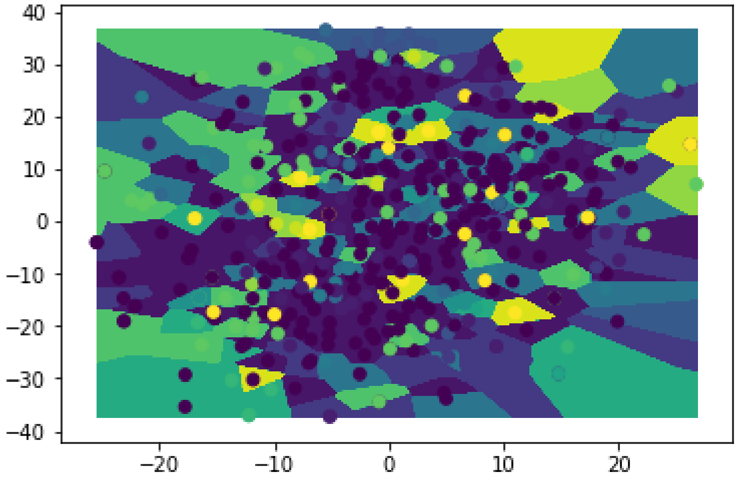
P (c|x) is the posterior probability of class x

P(x) is the prior probability of predictors

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

this algorithms convert input into frequency tables and then with the help of calculated prior probabilities , we calculate posterior probabilities and consider a highest among them as outcome. The results are summarised below –



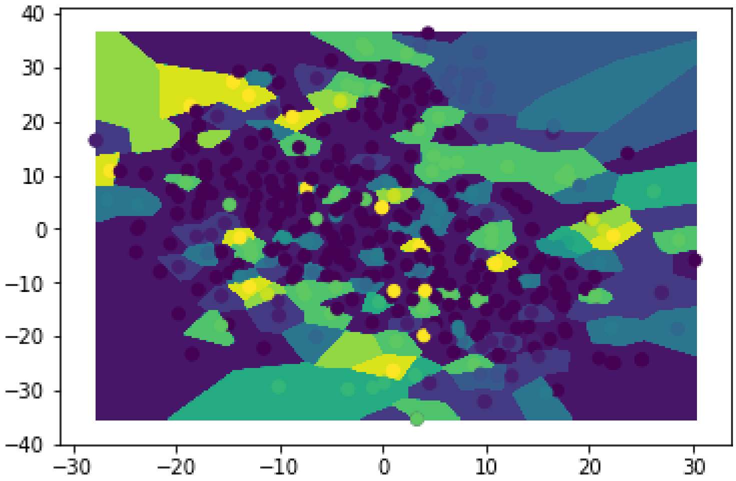
Training-Testing Size	Training Accuracy	Test Accuracy
70%-30%	70.56 %	55.88 %

FIGURE 6: Naïve Bayes Classification with PCA

D. SVM (Support Vector Machines)

In SVM we find a Hyperplane for N dimensional space where N is the number of features, this hyper plane is a line for 2 dimensional and a plane is considered as a hyperplane for 3 Dimension. Points falling in same side of a hyperplane is considered as in a same class.by plugging input values into the equation of hyperplane, we can predict the class of a input.

In SVM our aim is to maximize the margin of hyperplane from the points. For this purpose we have Support vectors, this are the points closer to hyperplane and influence its position.



Training-Testing Size	Training Accuracy	Test Accuracy
70%-30%	95.88 %	**72.05** %

FIGURE 7: SVM Classification with PCA

Tools & Technology To Be Used

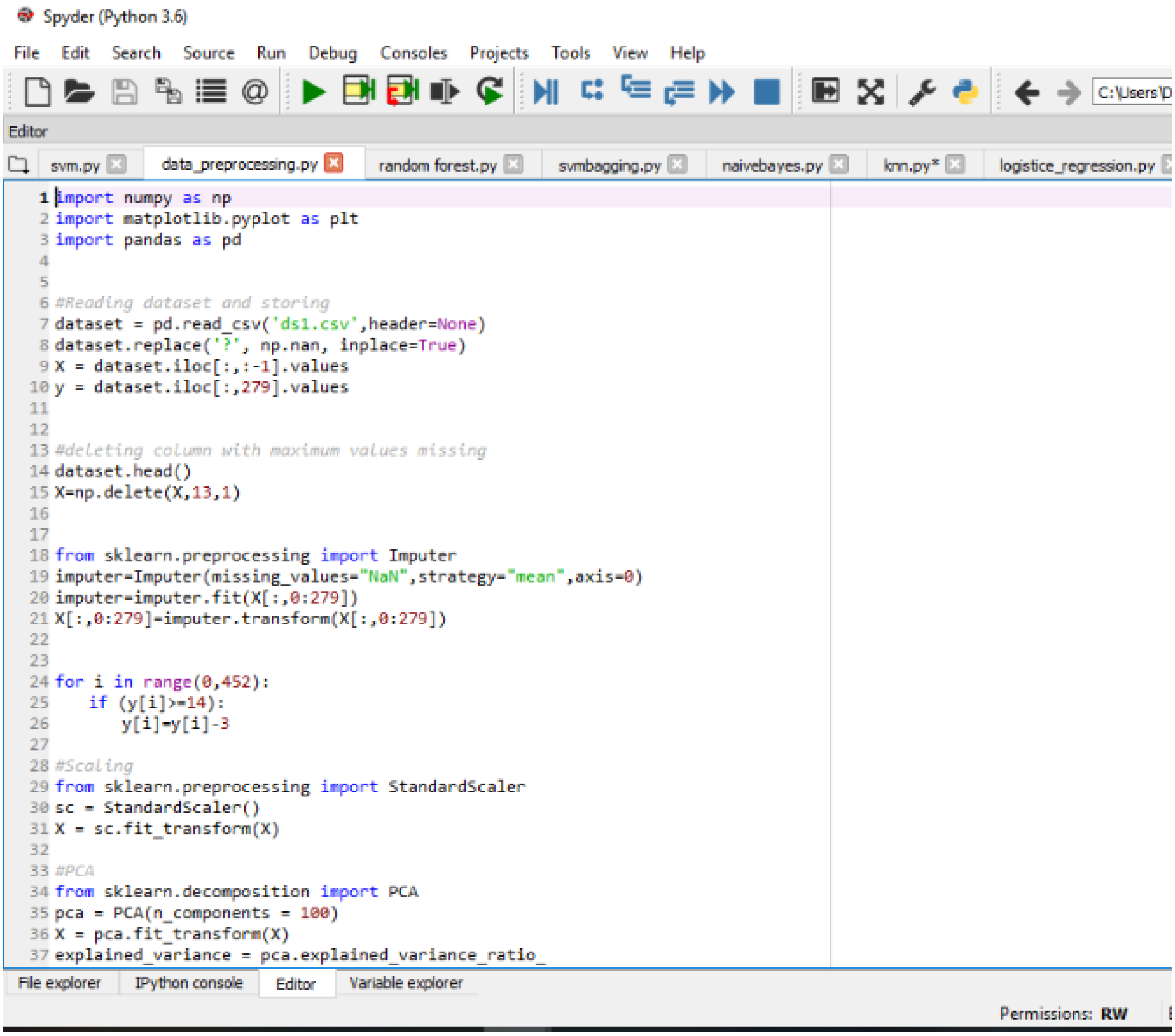
Software:

- Anaconda
- Jupiter Notebook
- WEKA
- MATLAB
- TENSOR FLOW

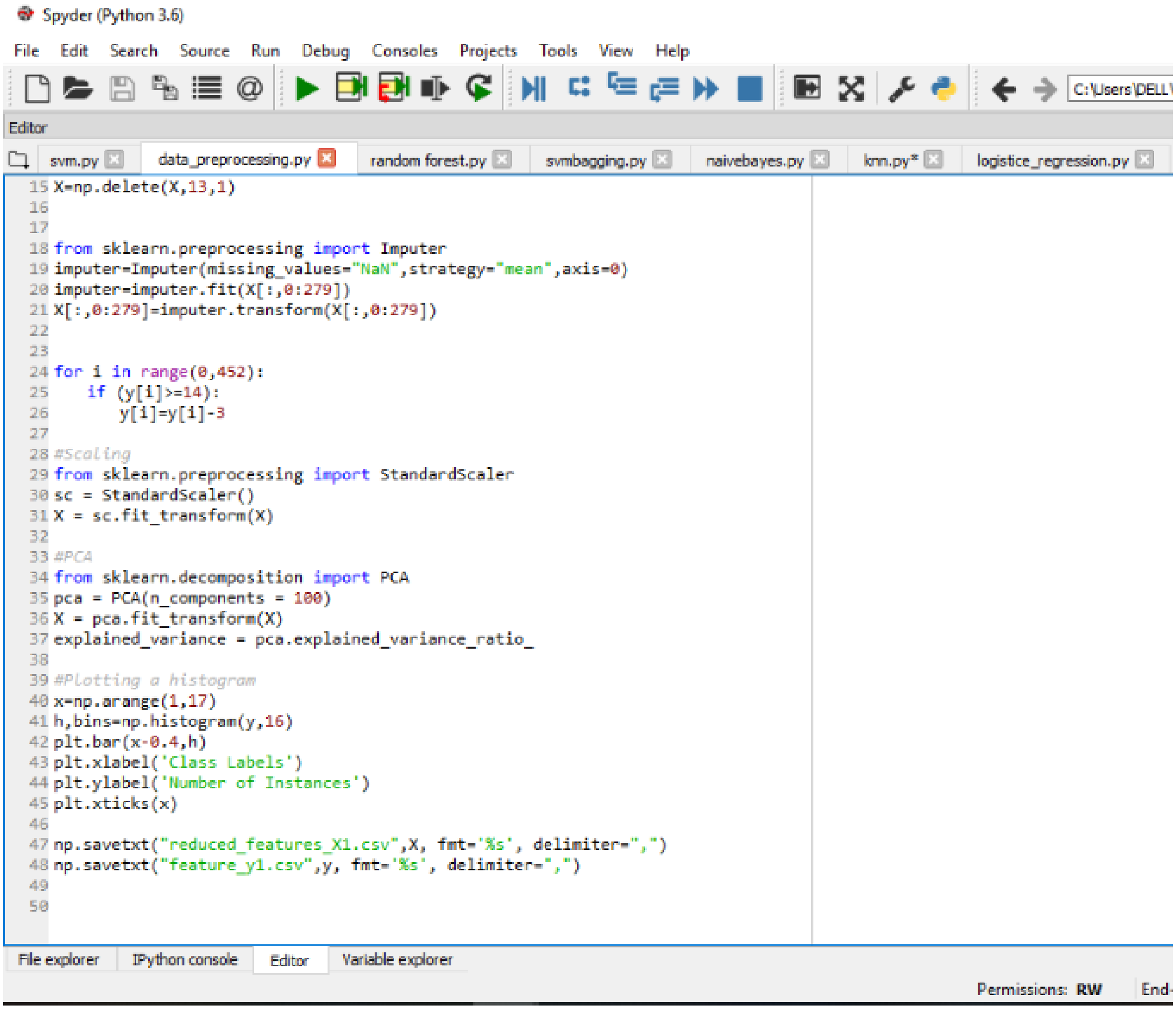
Hardware:

- WINDOWS
- RAM-at least 4GB
- PROCESSOR-(i5-i7) 7th generation

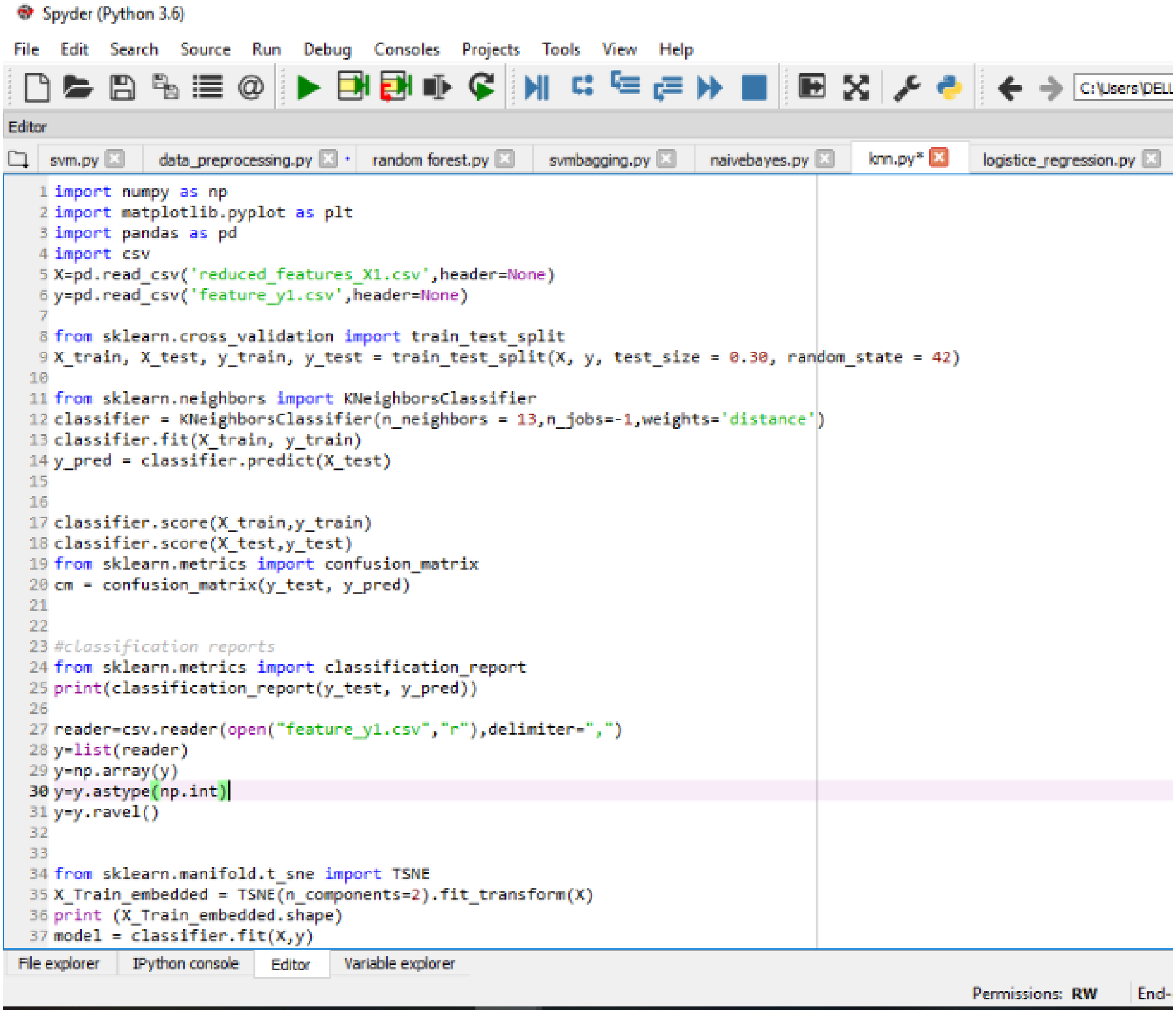
Implementation and Coding



```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


#Reading dataset and storing
dataset = pd.read_csv('ds1.csv',header=None)
dataset.replace('?', np.nan, inplace=True)
X = dataset.iloc[:,:-1].values
y = dataset.iloc[:,279].values


#deleting column with maximum values missing
dataset.head()
X=np.delete(X,13,1)


from sklearn.preprocessing import Imputer
imputer=Imputer(missing_values="NaN",strategy="mean",axis=0)
imputer=imputer.fit(X[:,0:279])
X[:,0:279]=imputer.transform(X[:,0:279])


for i in range(0,452):
    if (y[i]>=14):
        y[i]=y[i]-3

#Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)

#PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 100)
X = pca.fit_transform(X)
explained_variance = pca.explained_variance_ratio_
```



```python
X=np.delete(X,13,1)


from sklearn.preprocessing import Imputer
imputer=Imputer(missing_values="NaN",strategy="mean",axis=0)
imputer=imputer.fit(X[:,0:279])
X[:,0:279]=imputer.transform(X[:,0:279])


for i in range(0,452):
    if (y[i]>=14):
        y[i]=y[i]-3

#Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)

#PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 100)
X = pca.fit_transform(X)
explained_variance = pca.explained_variance_ratio_

#Plotting a histogram
x=np.arange(1,17)
h,bins=np.histogram(y,16)
plt.bar(x-0.4,h)
plt.xlabel('Class Labels')
plt.ylabel('Number of Instances')
plt.xticks(x)

np.savetxt("reduced_features_X1.csv",X, fmt='%s', delimiter=",")
np.savetxt("feature_y1.csv",y, fmt='%s', delimiter=",")
```



```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import csv
X=pd.read_csv('reduced_features_X1.csv',header=None)
y=pd.read_csv('feature_y1.csv',header=None)

from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 42)

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 13,n_jobs=-1,weights='distance')
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)


classifier.score(X_train,y_train)
classifier.score(X_test,y_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)


#classification reports
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

reader=csv.reader(open("feature_y1.csv","r"),delimiter=",")
y=list(reader)
y=np.array(y)
y=y.astype(np.int)
y=y.ravel()


from sklearn.manifold.t_sne import TSNE
X_Train_embedded = TSNE(n_components=2).fit_transform(X)
print (X_Train_embedded.shape)
model = classifier.fit(X,y)
```

RESULTS

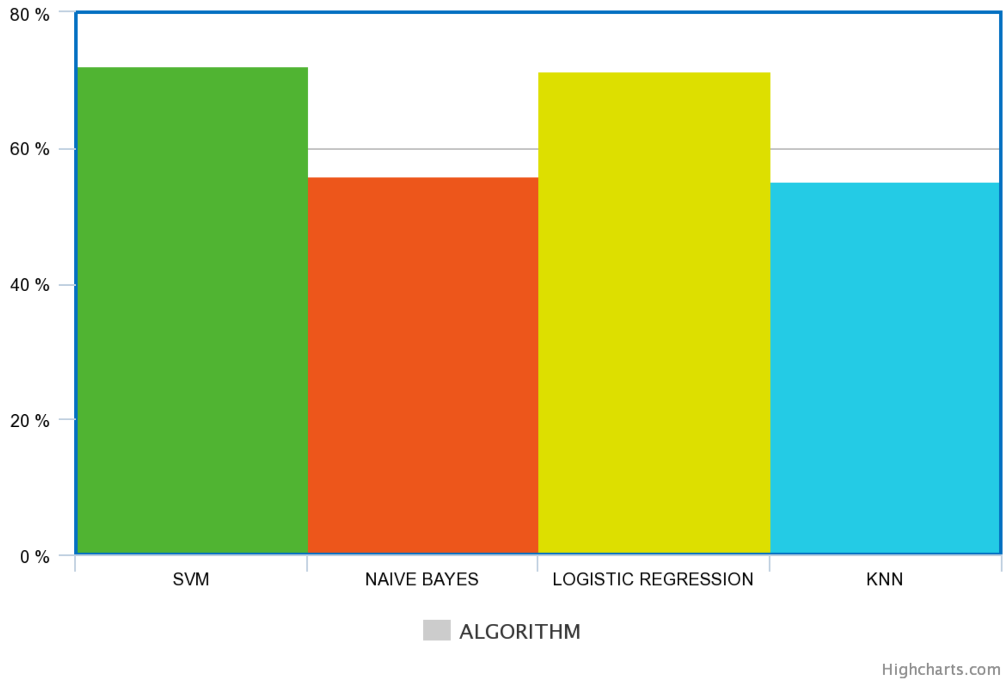


FIGURE 8: COMPARISON GRAPH BETWEEN DIFFERENT ALGORITHMS

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.

We obtained best result in SVM with 72.05 accuracy, other algorithms Logistic regression gives accuracy 71.42, KNN gives accuracy 55.14 and naïve Bayes gives accuracy 55.88.

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 = 88.9% and Testing Data=72%). 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 96 % for training set and 73 % for testing set.

CONCLUSION

We used 4 classifiers for the classification of cardiac arrhythmia. These were Naive Bayes Algorithms, Support Vector Machine, Logistic Regression and KNN classifier.

When the dataset was cross-validated and tested, the maximum accuracy was found to be obtained by Support Vector Machine Classifier. The accuracy obtained was 72.05%.

Thus in our approach, we have used the Support Vector Machine Classifier to obtain the best possible results for classifying arrhythmia.

GAPS IDENTIFIED

A number of combinations of algorithms can be implemented in the hierarchical scheme. KNN algorithm is a little slow because of distance, and SVM is sometimes becomes incorporates when datasets is very large. We would try to improve it.

REFERENCES

[1] "UCI machine learning repository: Arrhythmia data set," 1998. [Online].

Available: https://archive.ics.uci.edu/ml/datasets/Arrhythmia. Accessed:Feb. 10, 2017.

[2] "Heart attack kills one person every 33 seconds in India - Times of India", The Times of India, 2017. [Online]. Available: http://timesofindia.indiatimes.com/life-style/health-fitness/healthnews/ Heart-attack-kills-one-person-every-33-seconds-in-

India/articleshow/52339891.cms. [Accessed: 09- Mar- 2017].

[3] S. Xue, X. Chen, Z. Fang, and S. Xia, "An ECG arrhythmia classification and heart rate variability analysis system based on android platform," 2015 2nd International Symposium on Future Information Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC) and Communication Technologies for Ubiquitous HealthCare (Ubi- HealthTech), May 2015.

[4] Nir Kalkstein, Yaron Kinar, Michael Na'aman, Nir Neumark, and Pini Akiva, "Using Machine Learning to Detect Problems in ECG Data Collection," in Computing in Cardiology, IEEE, 2011.

[5] O. Valenzuela, F. Rojas, L. J. Herrera, F. Ortuno, H. Pomares, and I. Rojas, "Comparison of different computational intelligent classifier to autonomously detect cardiac pathologies diagnosed by ECG," 2013 13th International Conference on Intelligent Systems Design and

Applications, Dec. 2013.

[6] Desai, Usha et al. "Machine Intelligent Diagnosis Of ECG For Arrhythmia Classification Using DWT, ICA And SVM Techniques". 2015 Annual IEEE India Conference (INDICON) (2015): n. pag. Web.4 Sept. 2016.

[7] Deselaers, Thomas, Lexi Pimenidis, and Hermann Ney. "Bag-of- visual words models for adult image classification and filtering." *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE,

2008.

[8] **https://ieeexplore.ieee.org/document/8282537**