MACHINE LEARNING PROJECT

AIR INSTRUMENT

Playing Music With Gestures

Team Name: Grey Matter

Team Members:

Aayush Batra - 13ME33006
Amol Sakhale - 13EE35005
Anem Phani Vinusha - 13AG36002
Rupesh Arora - 13AG36012
Sravya Dumpala - 13EE35010

Introduction

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Evolved from the study of pattern recognition and computational learning theory. Machine learning is being widely used in the field medical diagnosis, marketing, photography, music, finance, big data, advertizing etc. Used for improving performance of data based system and also predict. Fundamental machine learning techniques can be used to make sense of human gesture, musical audio, and other real-time data. Air Instrument is one such unique mechanism involving supervised machine learning techniques to create music by gestures.

A robust and accurate position detection and motion recognition system that could be used for playing music using hand gestures mimicking musical instrument. Involving acquiring real time data from 3- axial accelerometer and gyroscope of smart phone and integrating it with wifi enabled bridge which filters and classifies using KNN classifier connecting it to a media file creating music. The classification of the data uses machine learning techniques making it an accurate system and robust.

K Nearest Neighbours

K nearest neighbour is the technique that is used for in classification of the mobile sensor data. KNN is a non-parametric lazy learning algorithm it does not make any assumptions on the underlying data distribution. Most of the practical data does not obey the typical theoretical assumptions made. It is a lazy algorithm it means that it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal.

Lack of generalization KNN keeps all the training data. KNN makes decision based on the entire training data set. There is a minimal training phase but a costly testing phase. More memory is needed as we need to store all training data.

KNN assumes that the data is in a *feature space*, the data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance. This need not necessarily be Euclidean distance although it is the one commonly used.

Each of the training data consists of a set of vectors and class label associated with each vector. In the simplest case, it will be either + or – (for positive or negative classes). But KNN, can work equally well with arbitrary number of classes. We are also given a single number "k". This number decides how many neighbors influence the classification.

Training Data Collection and Real Time Data Acquisition

Real time data has been acquired from accelerometer and gyroscope sensors of smart phone connected through Matlab Mobile application with sampling time as 1 second.

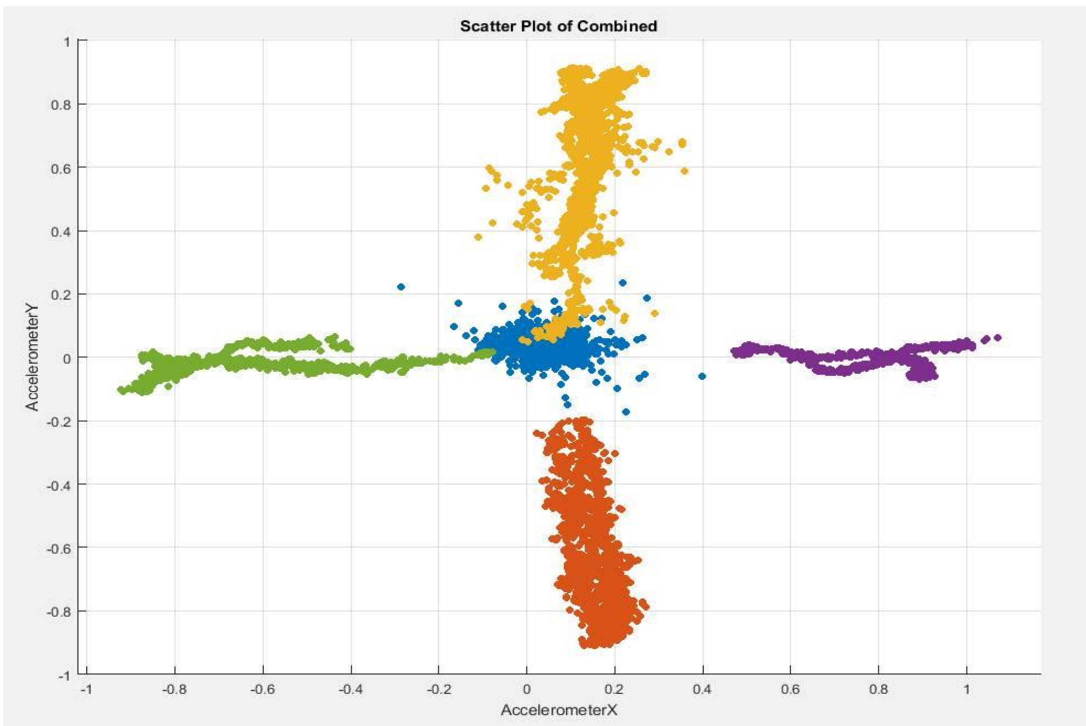
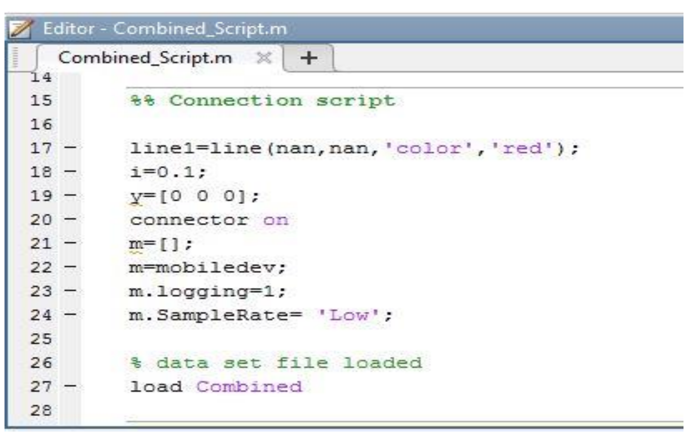


Fig.1: Complete Data Scatter Plot




```
%% Connection script

line1=line(nan,nan,'color','red');
i=0.1;
y=[0 0 0];
connector on
m=[];
m=mobiledev;
m.logging=1;
m.SampleRate= 'Low';

% data set file loaded
load Combined
```

Fig.2: Real time data acquisition code

Training of classifier

The Android sensor framework lets you access many types of sensors which includes accelerometer and gyroscope which has been integrated with mobile Matlab application and Matlab in the personal computer bridged through wifi connection. Data has been collected at sampling frequency of 50Hz for different gestures used for training the model . The Data has been classified in 5 classes with total 14239 data instances with 6 features.

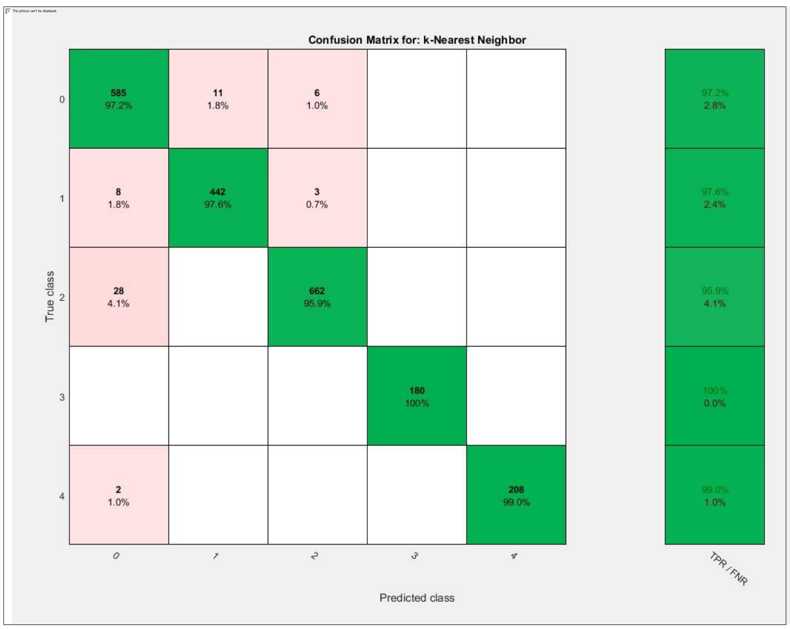
```
%% Classifier and Sound Script

% features extracted
x=Combined(1:14239,2:7);
% Labels extracted
y=Combined(1:14239,8);
Md1 = fitcknn(x,y);
Md1.NumNeighbors=5; % set the value of k to be 5
rng(10);

[y,Fs] = audioread(char(listing(j)));
player = audioplayer(y,Fs);
play(player);
volume = 0.4;

while 1
a=m.Acceleration/9.81;
w=m.AngularVelocity;
f=[a w];
value = predict(Md1,f);
x1=get(line1,'xdata');
y1=get(line1,'ydata');
x1=[x1 i];
if (i==0.1)
    value = 0;
end
y1=[y1 value];
set(line1, 'xData',x1,'yData',y1);
i=i+2;
pause(1.5);
```

Results:

Prediction from classifiers

Comparison of Classifiers:

1. Coarse KNN classifier confusion matrix.

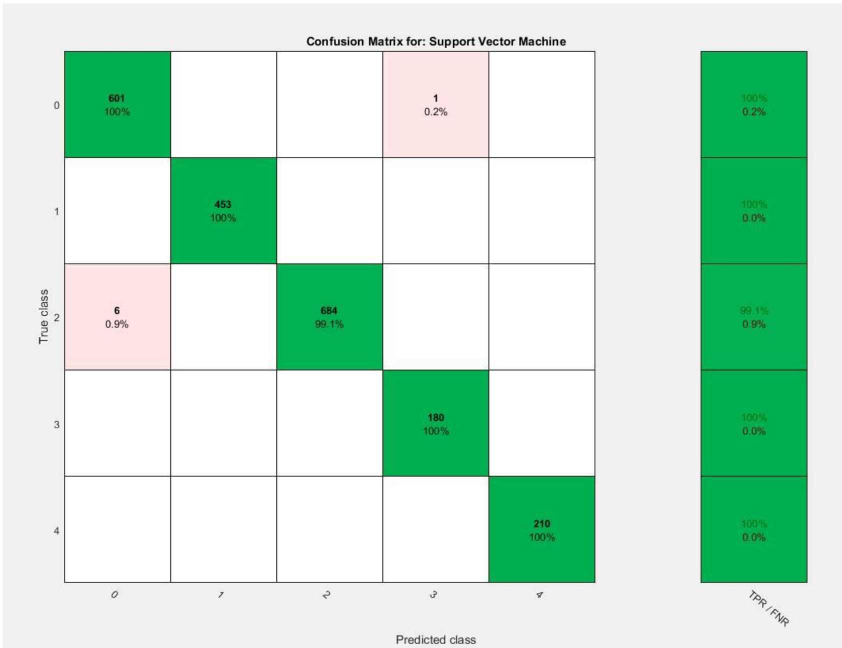


2. Cubic SVM Classifier confusion matrix

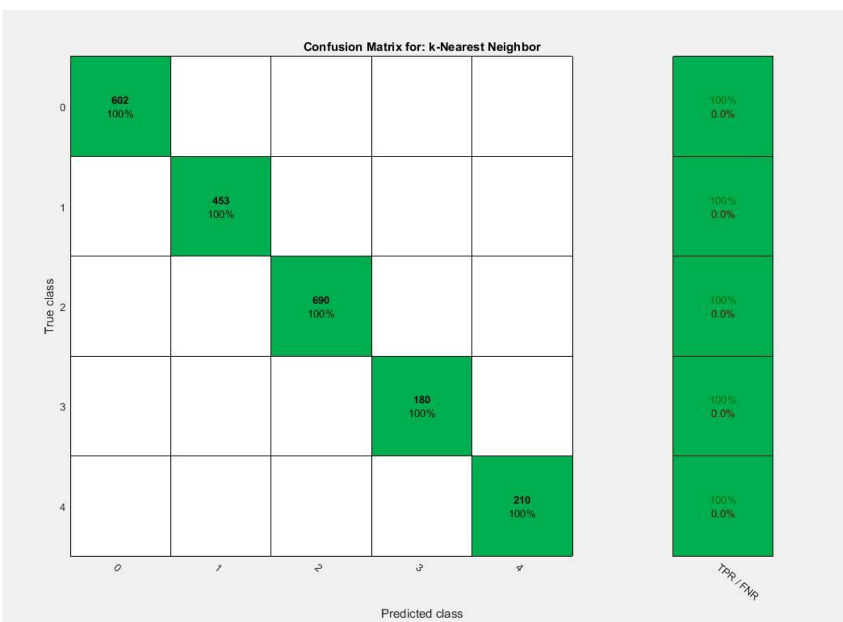
3. Linear SVM classifier confusion matrix

4. Medium KNN classifier confusion matrix

5. Quadratic SVM classifier confusion matrix



6. Weighted KNN Classifier confusion matrix



As per our analysis, Weighted KNN is giving maximum accuracy out of all classifiers we have used.

SCATTER PLOT REPRESENTATION OF PREDICTION BY KNN