Report:

Identification of P300 signals after visual stimuli:

As per [1] the P300 are positive peaks in EEG signals due to visual or auditory stimuli. Thus they are event related peaks in the EEG. Identification of P300 helps in decision-making.
There are issues of muscle artifacts that generate noise in the P300 signals but can be identified by filtering them out. They are usually high frequency signals and P300 lies between 0.1 and 15 Hz. Thus a band pass filter can be used in order to separate out the frequencies and just keep the useful frequencies of P300 [2]. [3] Shows that down-sampling is also done on the database.

Thus the P300 signals can be isolated before further analysis. Then feature extraction based on wavelet transform is applied. It was noted from [4] that the important features of P300 lie between 1-8 Hz and thus wavelet transform was used to extract that.

Machine Learning Techniques:

I have tried implementing two techniques: SVM and Decision Trees.

Reason for application of DT over SVM for this task:

1. Data is usually large thus DT performs better compared to SVM in larger data with lesser dimensions
2. Model complexity in SVM classification is much higher compared to DT
3. DT can be easy to interpret thus help more in understanding the signals by the tree graphs and helps in analysis
4. Training time for SVM is higher than DT and thus for larger dataset the training time increases

An analysis had been done in [5] where they show the accuracy reached by DT is higher to that of SVM. K-NN also shows good efficiency but I am applying DT here.
Another research [6] based on EEG data shows the performance of DT compared to SVM is much higher.

Database:

Visual P300 speller database [7]

In this database, 10 subjects are considered with signals taken by 9 channels and are classified as '1' if target is flashing and '0' if it is not flashing.

Results:

I obtained results after the processing of the data as well as before that.

The processed data shows SVM is slightly better than DT in terms of accuracy but these was all based out of 10 channels. If channels are lesser, the accuracy of DT is higher than SVM.
It is notable that the database for subject #2 is smaller in size compared to the others and hence DT performs weaker than SVM for that particular subject. Thus DT would show better results if the size of the database increases.

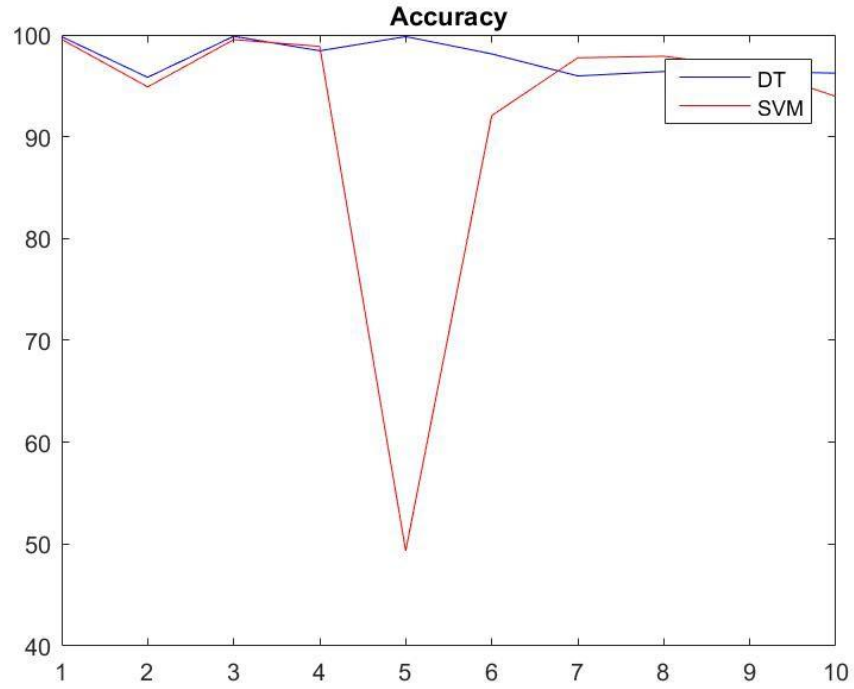
Also, if the dimensions decrease, DT would perform better than SVM.
In terms of time, DT is way faster than SVM. Thus for real time analysis, DT can be used for testing. I also tried with LDA but DT performed better than that.

Description of techniques:

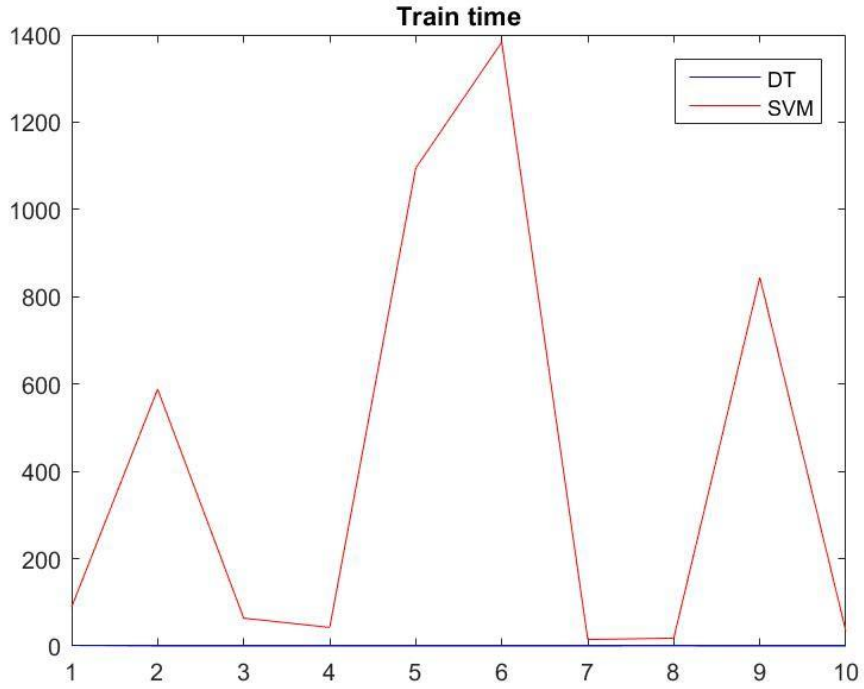
1. No-Features-Extracted – Muscle artifacts are eliminated but features are not extracted
2. Features-Extracted – Muscle artifacts are eliminated, features are detected by wavelet transform using 'haar' technique

No-Features-Extract:

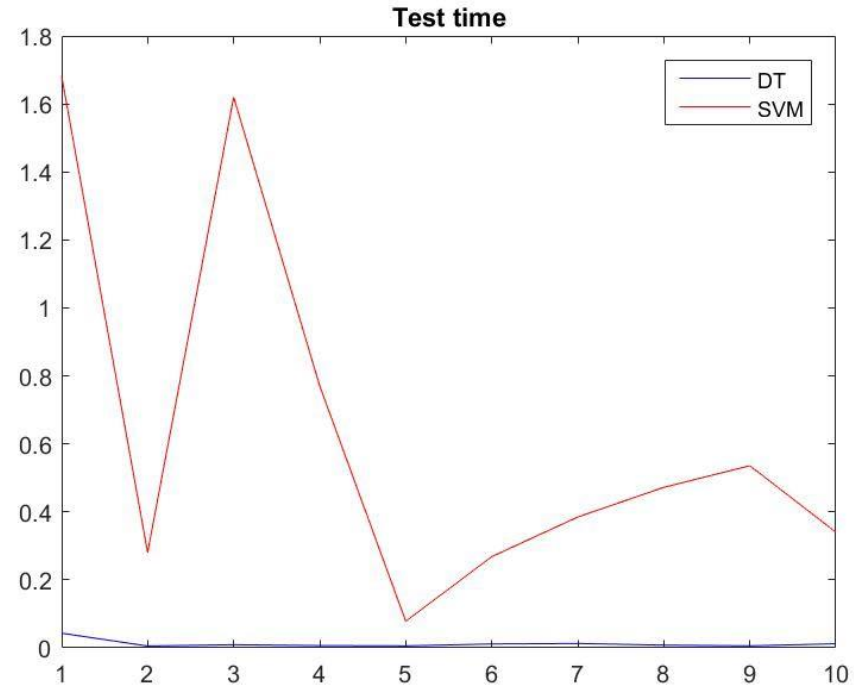
1. Accuracy:



2. Training time:

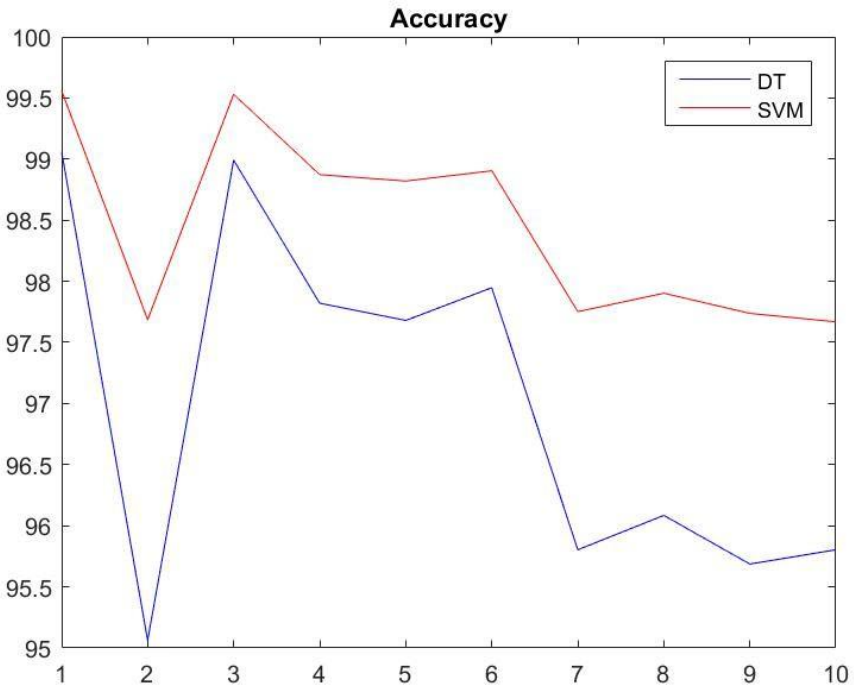


3. Testing time:

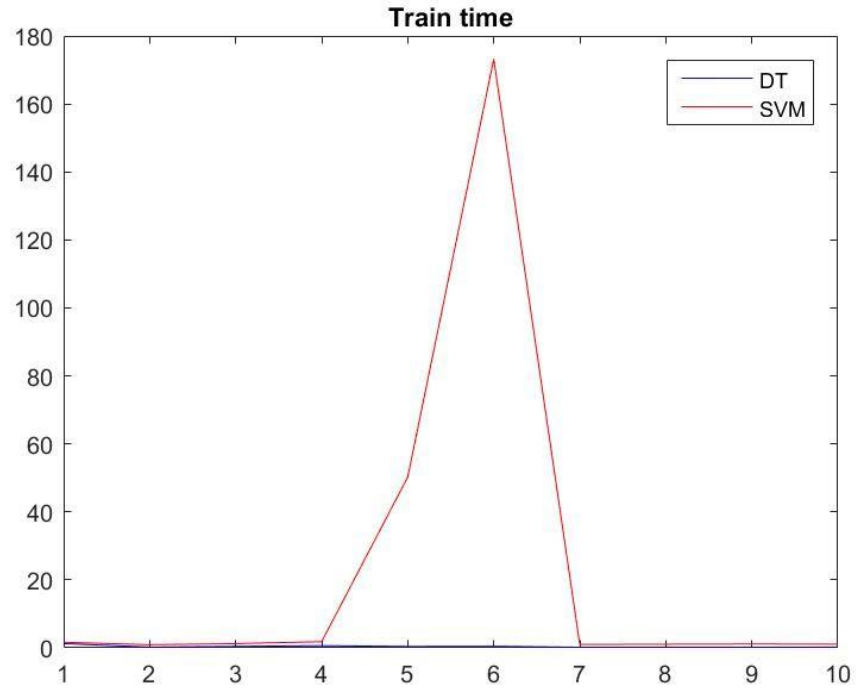


Features-Extracted:

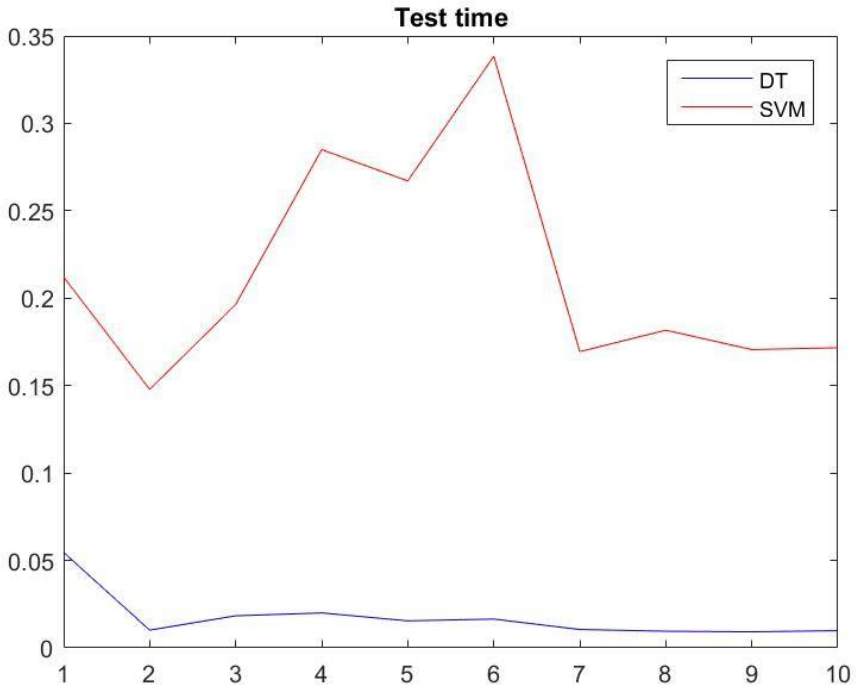
1. Accuracy:



2. Training time:



3. Testing data:



Roadmap:

1. Learning in detail about P300 and the uses of the same.
2. Learning the different ways to obtain P300 from EEG signals. This is important as they are obtained in certain range only and their features have to be extracted. Moreover the noise signals and muscle artifacts have to be considered and removed eventually.
3. Comparing different algorithms and using them for analysis. I chose k-NN instead of DT but it gave worse results compared to SVM thus I chose DT. The reasons for choosing DT is as described above.
4. Problems I encountered and how I overcame:
    a. Obtaining information about extracting P300 from EEG signals but then on searching more I could get a proper reference which showed the frequency range between which P300 can be obtained
    b. I tried with k-NN earlier but yielded worse results than SVM thus I chose to deal with DT
    c. Selecting a proper technique for feature extraction. I finally chose a common method of haar-wavelets but specific transform could be used to obtain better results.

Citations:

[1] http://www.mdpi.com/1424-8220/12/2/1211
[2] https://hal.inria.fr/hal-00756669/document
[3] http://www.sciencedirect.com/science/article/pii/S0304394009008192
[4] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3660708/
[5] http://www4.ncsu.edu/~arezaei2/paper/JCIT4-184028_Camera%20Ready.pdf
[6] http://www.mdpi.com/1424-8220/16/3/336
[7] http://bnci-horizon-2020.eu/database/data-sets

The above work is original to the best of my knowledge. The citations are mentioned. The code is also original and has been made after following the procedure.

APPENDIX – MATLAB Code

```
clear all
clc

fnames = dir('*.mat');
n_files = length(fnames);
sizesofDB = [];

%Filtering between 0.1 and 15 Hz
b = fir1(12,[0.0002,0.03]);
Hd = b'*b;

disp('Subject number:');
for i = 1:10

    disp(i);

    sub = struct2cell(load(fnames(i).name));

    sub_train = sub{:}.train(1:9,:)';
    sub_train_labels = sub{:}.train(11,:)';
    sub_test = sub{:}.test(1:9,:)';
    sub_test_labels = sub{:}.test(11,:)';

    sizesofDB = [sizesofDB; length(sub_train),
length(sub_test)];

    %Filter data
    sub_train = filter2(Hd, sub_train);
    sub_test = filter2(Hd, sub_test);

    %Wavelet analysis
    [c,l] = wavedec2(sub_train,3,'haar');
    sub_train = wrcoef2('d',c,l,'haar',3); clear c
l;
    [c,l] = wavedec2(sub_test,3,'haar');
    sub_test = wrcoef2('d',c,l,'haar',3); clear c
l;

    %Decision Trees:
    tic;
```

```
    tree = fitctree(sub_train, sub_train_labels);
time_tree_train(i) = toc; tic;
    labels_predict_tree = predict(tree,sub_test);
time_tree_test(i) = toc;
    temp_tree = (sub_test_labels +
labels_predict_tree);
    accuracy_tree(i) =
(length(temp_tree(temp_tree==2)) +
length(temp_tree(temp_tree==0)))/length(temp_tree)*
100;

    %SVM:
    tic;
    svm = fitcsvm(sub_train, sub_train_labels);
time_svm_train(i) = toc; tic;
    labels_predict_svm = predict(svm, sub_test);
time_svm_test(i) = toc;
    temp_svm = (sub_test_labels +
labels_predict_svm);
    accuracy_svm(i) =
(length(temp_svm(temp_svm==2)) +
length(temp_svm(temp_svm==0)))/length(temp_svm)*100
;

end

figure(1);
plot(1:10, accuracy_tree, 'b', 1:10, accuracy_svm,
'r');
legend('DT', 'SVM'); title('Accuracy');

figure(2);
plot(1:10, time_tree_train, 'b', 1:10,
time_svm_train, 'r');
legend('DT', 'SVM'); title('Train time');

figure(3);
plot(1:10, time_tree_test, 'b', 1:10,
time_svm_test, 'r');
legend('DT', 'SVM'); title('Test time');
```