

# Pattern Recognition

# Daily Sports and Activities

---

## Group-15

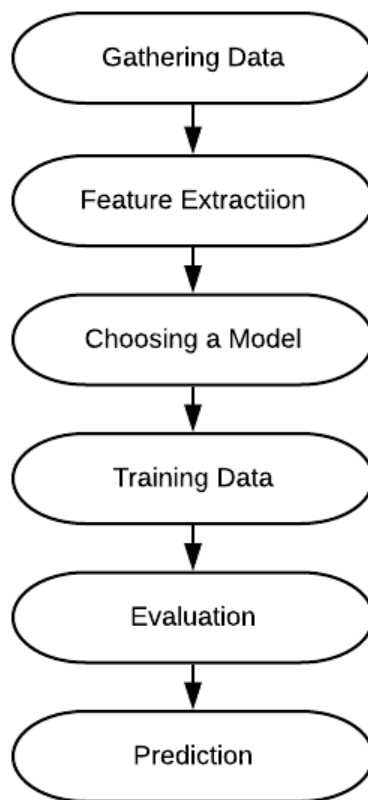
E. Pruthvik Reddy	S20170020203
E. Pavan Kumar	S20170020205
Shubham raj Gupta	S20170020238

---

## Problem Statement

To train a classifier in order to predict which activities a user is engaging in, based on sensor data collected from devices attached to body parts (limbs and torso).

## Classification Architecture



- 
- We first gather data. Sometimes the data we collect needs other forms of adjusting and manipulation. Things like de-duping, normalization, error correction, and more. Hence, we need to consider all those factors and prepare the data.
  - From the Multiple Models available we need to pick a model which best suits our data.
  - we then use our data to incrementally improve our model's ability to predict and classify the data.
  - The training process involves initializing some random values for  $W$  and  $b$  and attempting to predict the output with those values. We must adjust the values in  $W$  and  $b$  such that we will have more correct predictions. This process then repeats till the Cost Function Converges to a global Minima.
  - Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen and Finally Predict the output for a given problem.

## Feature Extracted

- 8 users participate in 19 activities. Each of the 5 devices (4 limbs and 1 torso) have 9 sensors (x, y, z accelerometers, x, y, z gyroscopes, and x, y, z magnetometers). The data is collected in 5 second segments with a frequency of 25 Hz for a total of 5 minutes for each activity for each user.
- Hence in each text file (of one segment), there are 5 units x 9 sensors = 45 columns and 5 sec x 25 Hz = 125 rows. Additional 15 columns of data are added which gives the magnitude of each sensor (magnitude of data of particular sensor). Hence overall, we obtain 125 rows and 60 columns for each segment.

## Mean, Variance, Skewness, and Kurtosis

The distribution of each signal is approximately Normal. This means that we can take the first four statistical moments for each 5 second segment. By including the four moments, we are helping our models better learn the characteristics of each unique activity.

## Autocorrelation

First Ten Values of the Autocorrelation are taken.

## Maximum five peaks of the Discrete Fourier Transform

---

After taking the DFT of each 5-s signal, the maximum five Fourier peaks are selected so that a total of 300 Fourier peaks are obtained for each segment.

Hence, for each column, Mean, Variance, Kurtosis, Skewness, Autocorrelation (First 10 values) and First Five Peaks of Discrete Fourier Transform are calculated using Fast Fourier Transformation Algorithm. This gives 19 features for each column and hence for 60 columns we obtain, 1140 features. (60x19).

## Classification Implementation

### Support Vector Machine

#### I. Quadratic SVM

If the feature vectors in the original feature space are not linearly separable, SVMs pre-process and represent them in a higher-dimensional space where they can become linearly separable. The Support Vector Machine model performed substantially better than Logistic Regression. The model is classifying activities at near 99% accuracy.

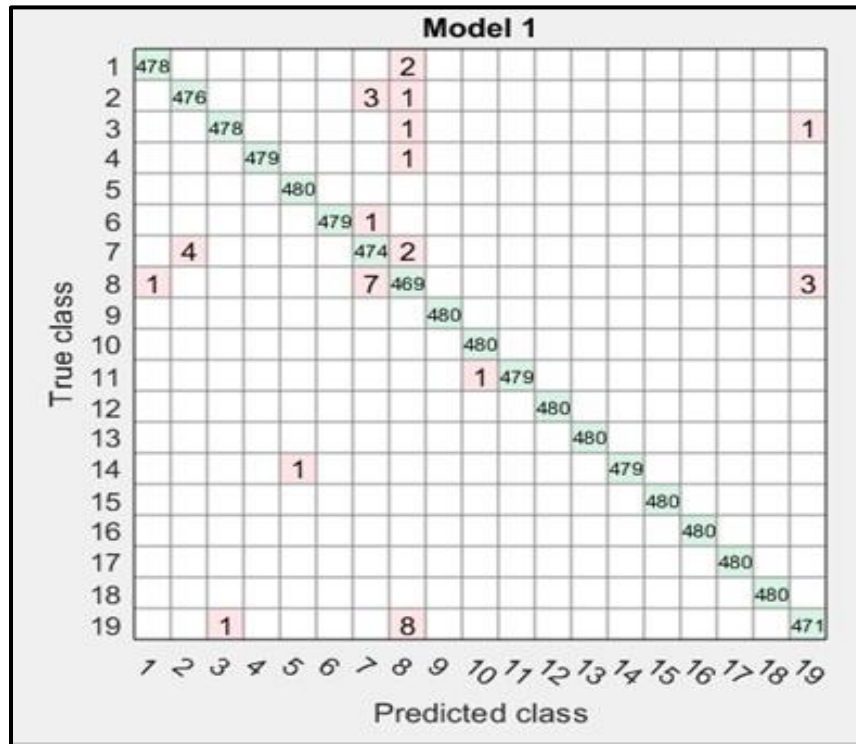
Support Vector Machine Model gives a better accuracy when we apply feature Normalization. With Feature Normalization (Normalizing all features in the scale between 0-1), we get an accuracy of 99.7%. On Applying PCA (reducing the dimension to 30 from 1140), the accuracy is dropped to 99.6%. The drop-in accuracy is insignificant when the training time is considered, which reduces immensely on applying PCA.

Without Feature Normalization, we get an accuracy of 99.7 % when we train the entire features. But the accuracy drops to 99% after applying PCA.

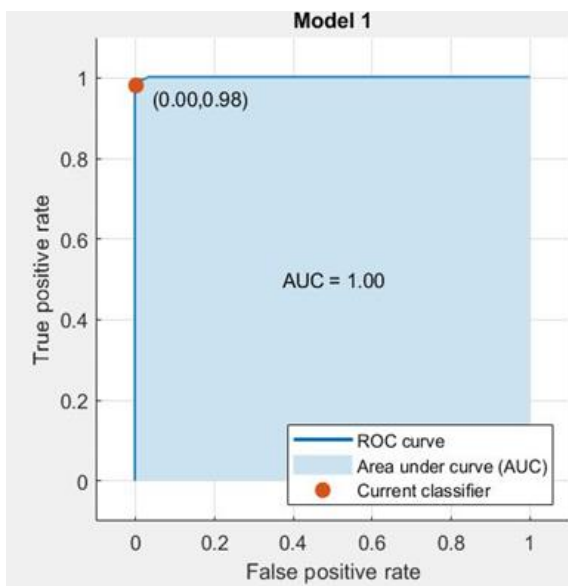
Hence, for Support Vector Machine, the ideal way is to Normalize the Features and then apply PCA (reducing it to 30 features from 1140) as it gives the ideal output in terms of both Accuracy and Training Time.

- **Accuracy: 99.6%**
- **F1-Score=0.9992**

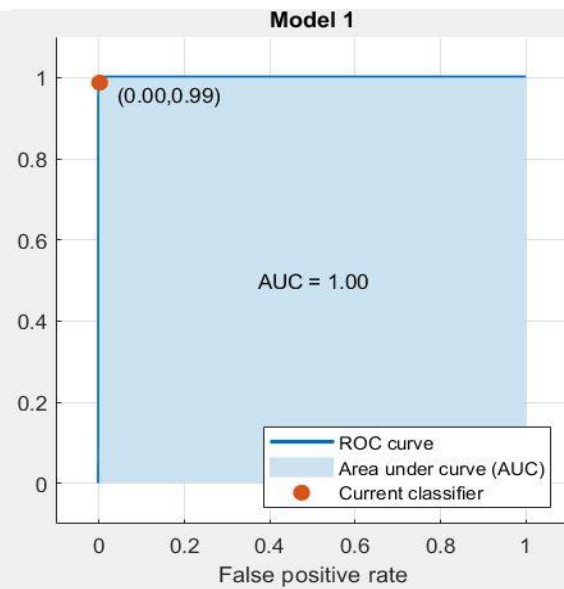
## CONFUSION MATRIX FOR QUADRATIC SVM



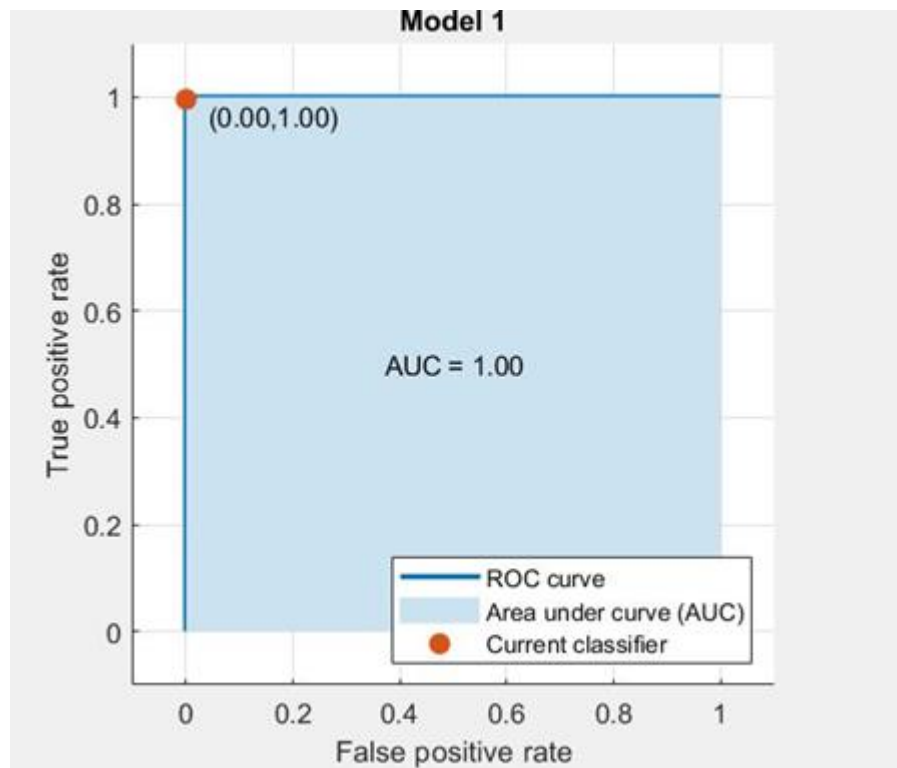
**ROC Curve for classes 8,19:**



**ROC Curve For classes 2,7:**



ROC curves for classes 1,3,4,5,6,9,10,11,12,13,14,15,16,17,18:



## II. Artificial Neural Networks

Multi-layer ANNs consist of an input layer, one or more hidden layers to extract progressively more meaningful features, and a single output layer, each composed of a number of units called neurons. The model of each neuron includes a smooth nonlinearity, called the activation function.

For ANN, Mean Normalization of the features give less accuracy compared to Non-Scaled data and hence we consider the original Scaling by own (Normalization or Not proved Insignificant for ANN)

---

5 hidden Layers	88.87%
10 hidden Layers	99.89%
20 hidden Layers	99.97%
30 hidden Layers	99.97%
40 hidden Layers	99.98%
100 hidden Layers	99.95%

An inordinately large number of neurons in the hidden layers can increase the time it takes to train the network. The amount of training time can increase to the point that it is impossible to adequately train the neural network.

The time taken for training increases with increase in the number of hidden Layers. Though the accuracy increases, it becomes insignificant when time taken to train is taken into account. So, 20 hidden layers are considered as it gives the best output in terms of Run-time and accuracy.

**Accuracy=99.97%;**

**Precision=0.9996;**

**Recall=0.9996;**

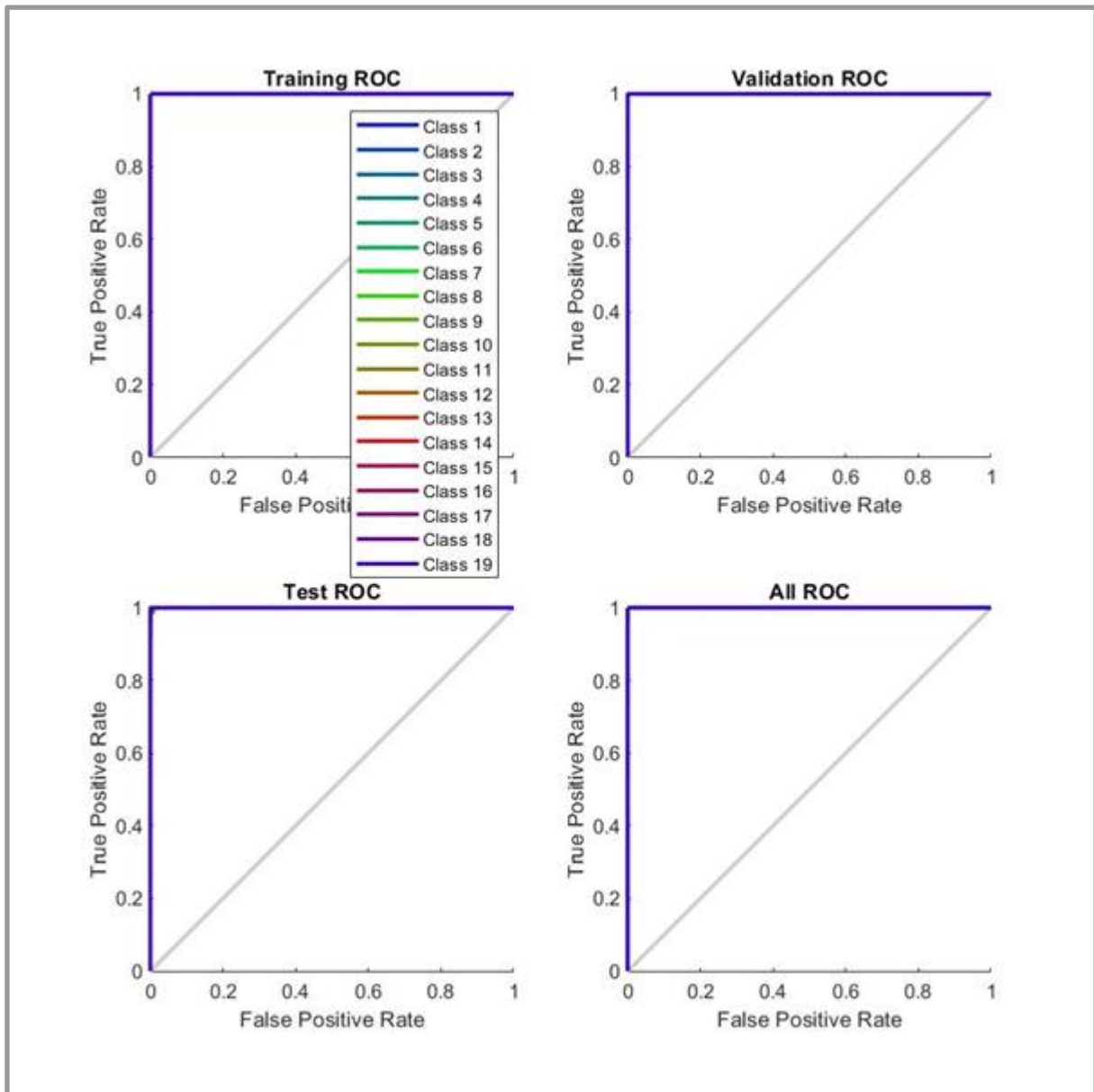
**F-1 Score=0.9996;**

## CONFUSION MATRIX FOR 20 HIDDEN LAYERS

Confusion Matrix																					
Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		
	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.4%	
	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	478 5.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.3%	0 0.0%	
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	479 5.3%	
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.6% 0.4%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.8% 0.2%	100% 0.0%	100% 0.0%
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
			Target Class																		



## ROC curves of ANN



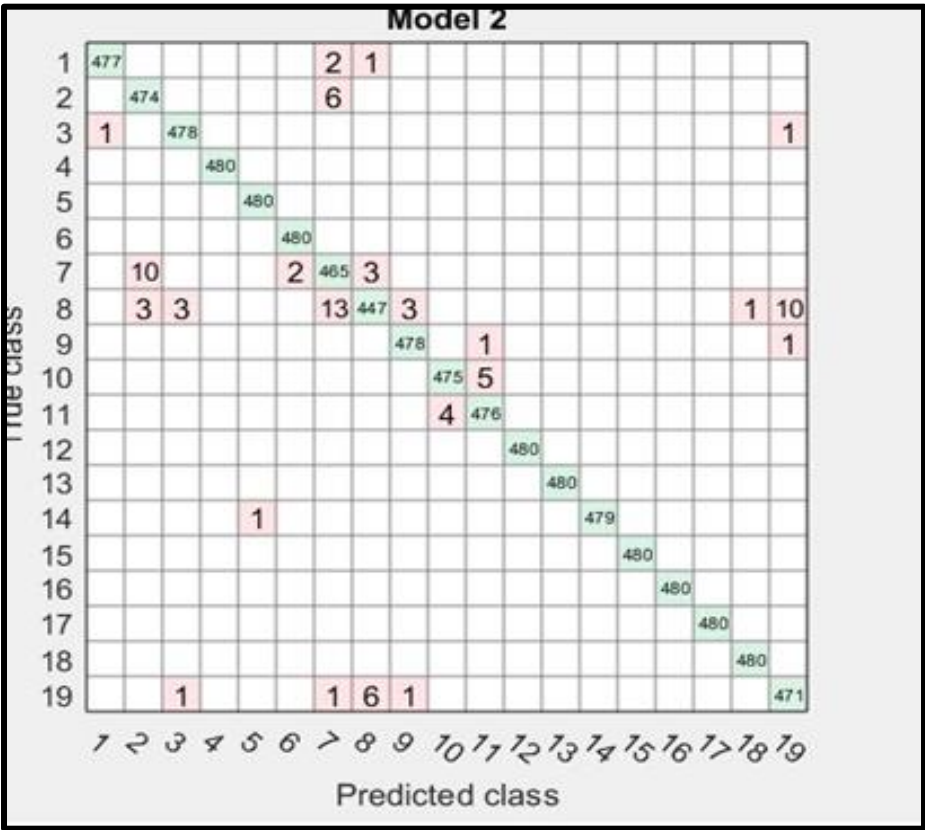
### III. KNN-Mean Normalization

We take the Mean-Normalized Feature Vector here as it gives the best accuracy. We also perform PCA reducing dimensional space to 30 from 1140.

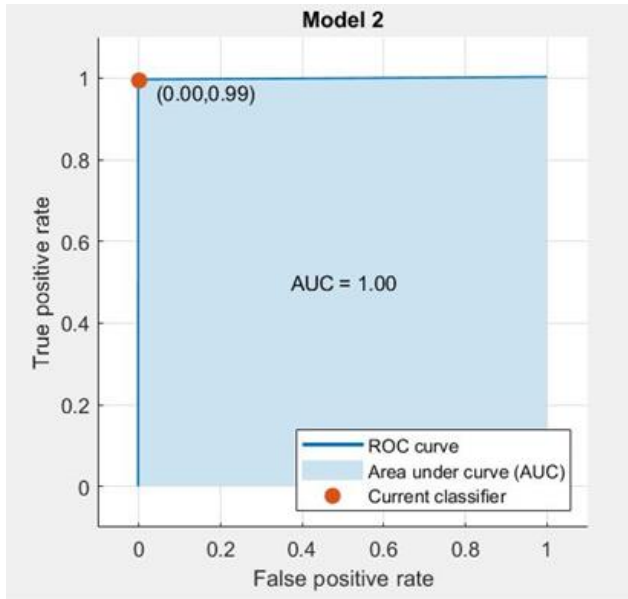
Accuracy=99.1%

F1-Score=1;

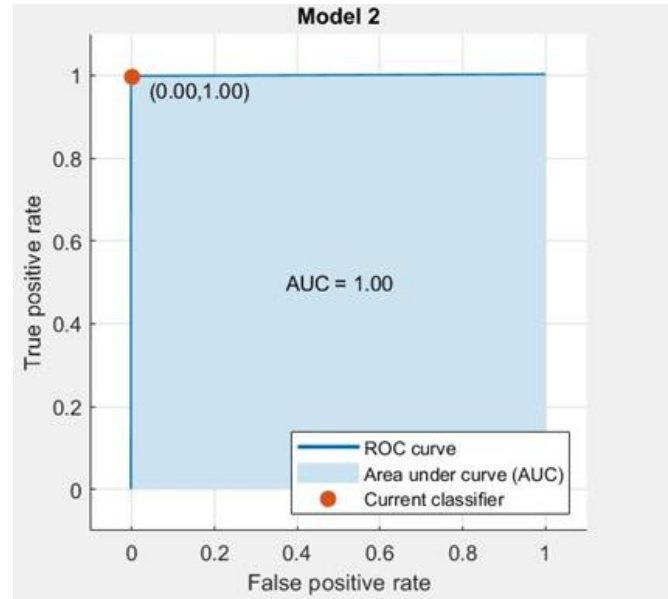
Confusion Matrix for KNN-Mean Normalization



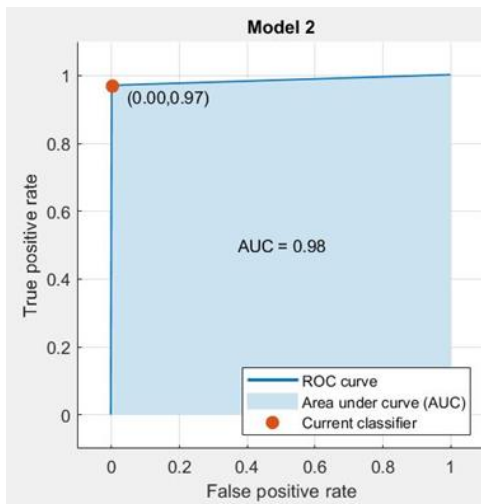
ROC Curves for classes 1,2,10,11



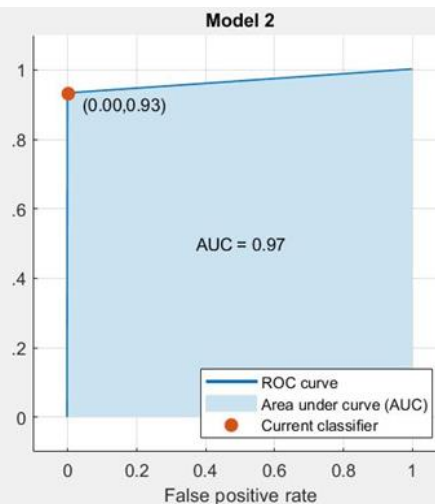
ROC Curves for Classes 3,4,5,6,9,12,13,14,15,16,17,18



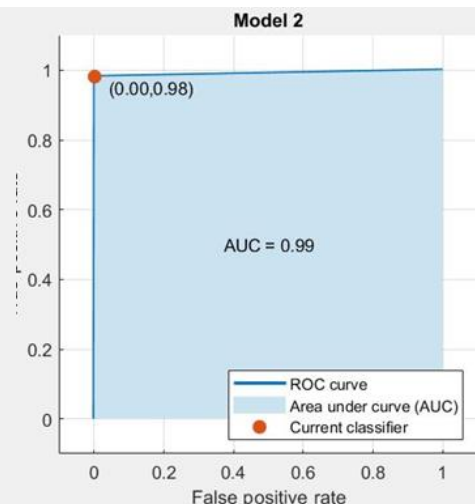
ROC curves for class 7:



ROC curve for Class 8:



ROC Curve for Class 19:

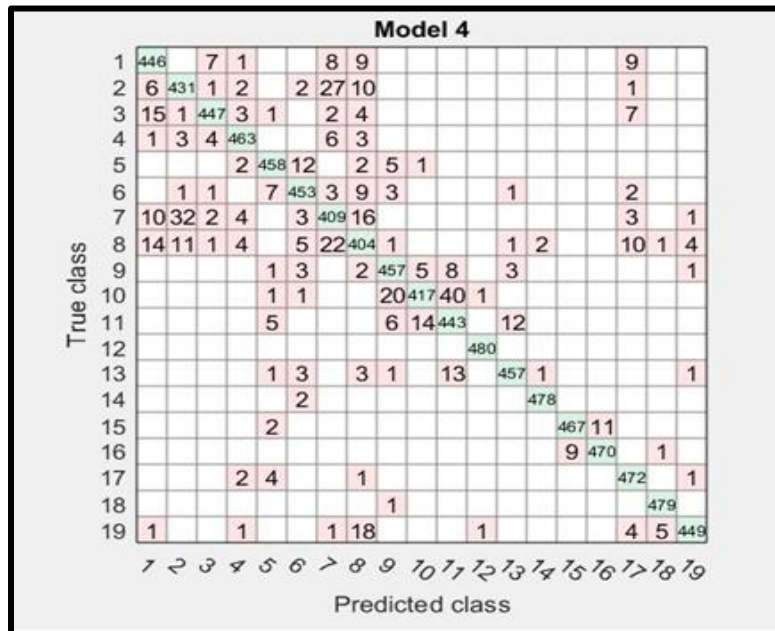


## IV. Decision Trees (FINE TREE)

We considered the Mean Normalized Feature Space here and Applied PCA, reducing Dimension from 1140 to 30.

**Accuracy:94.1%, F1-score=0.9658;**

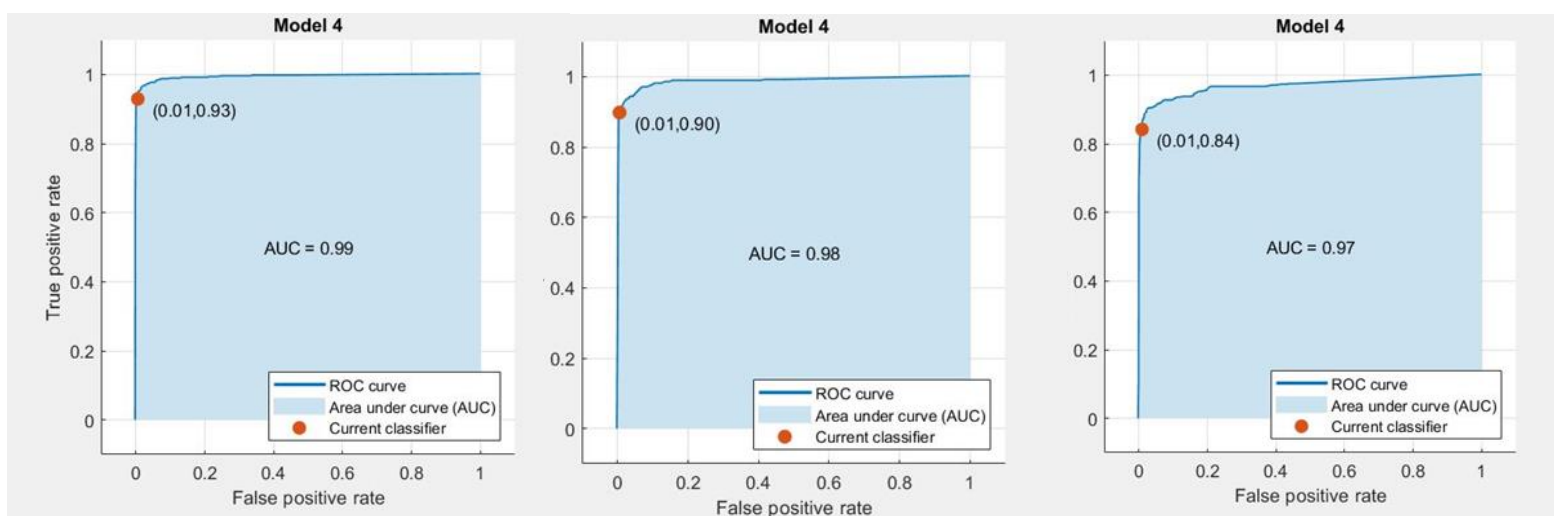
Confusion Matrix for Fine Tree:



ROC Curve for Class-1

ROC curve for Class 2

ROC curve for Class -8

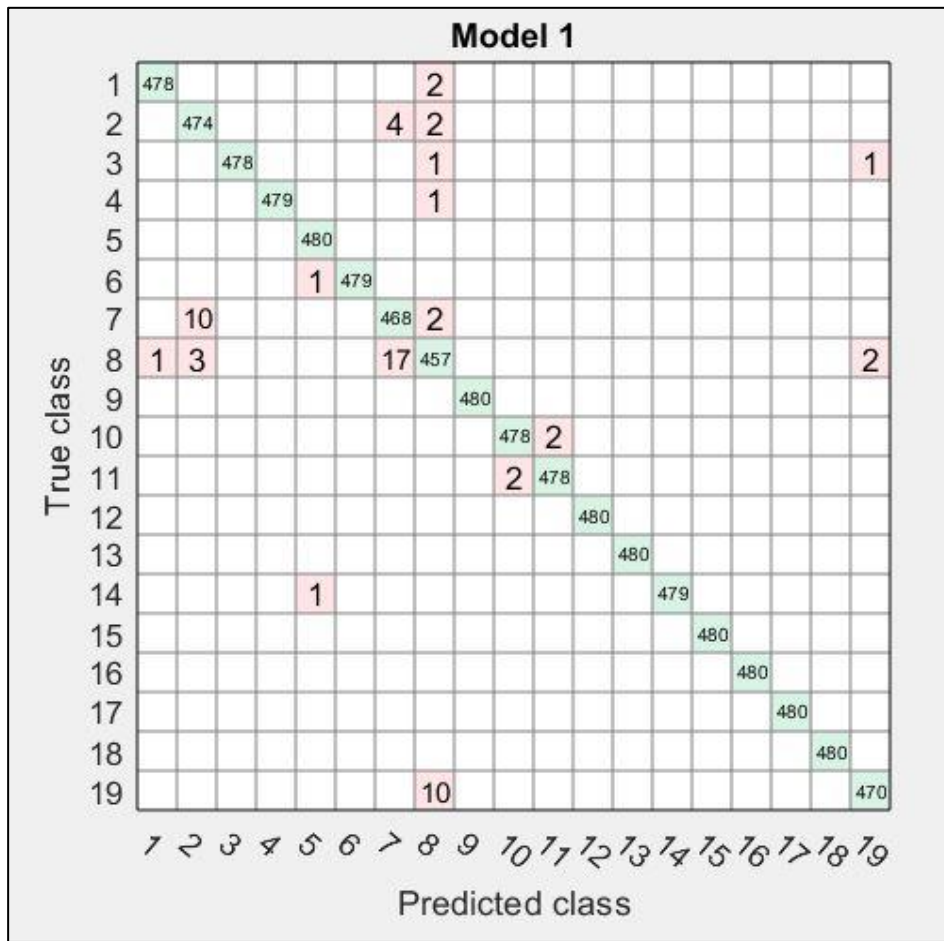


## V. Linear SVM

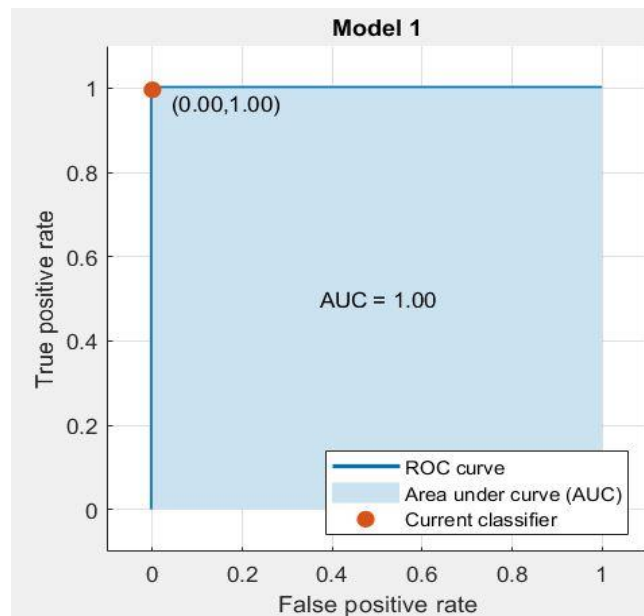
Mean Normalization is applied to the feature Space and then fed to Classification Learner with 30 Principal Components.

- Accuracy: 99.3%;
- F1-Score: 0.9962;

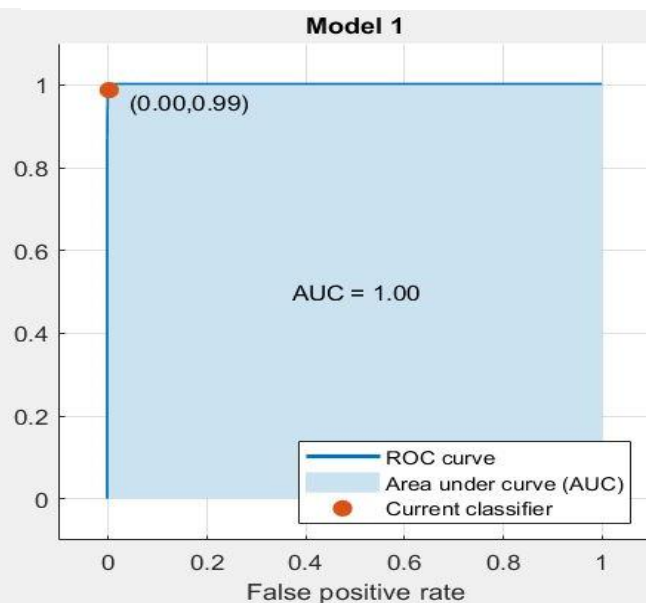
**Confusion Matrix for Linear SVM**



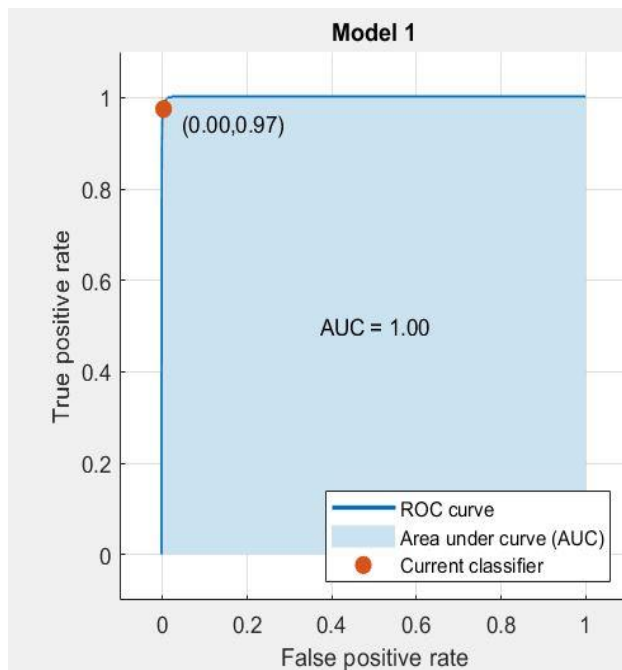
### ROC Curve for Class- 1,3,4,5,6,9,10,11,12,13,14,15,16,17



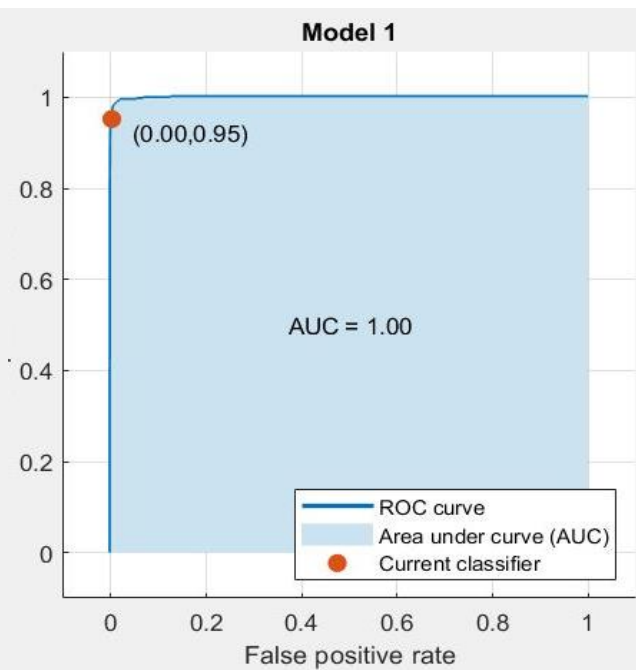
### ROC Curve for Class - 2



### ROC Curve for Class -07



### ROC Curve for Class -08

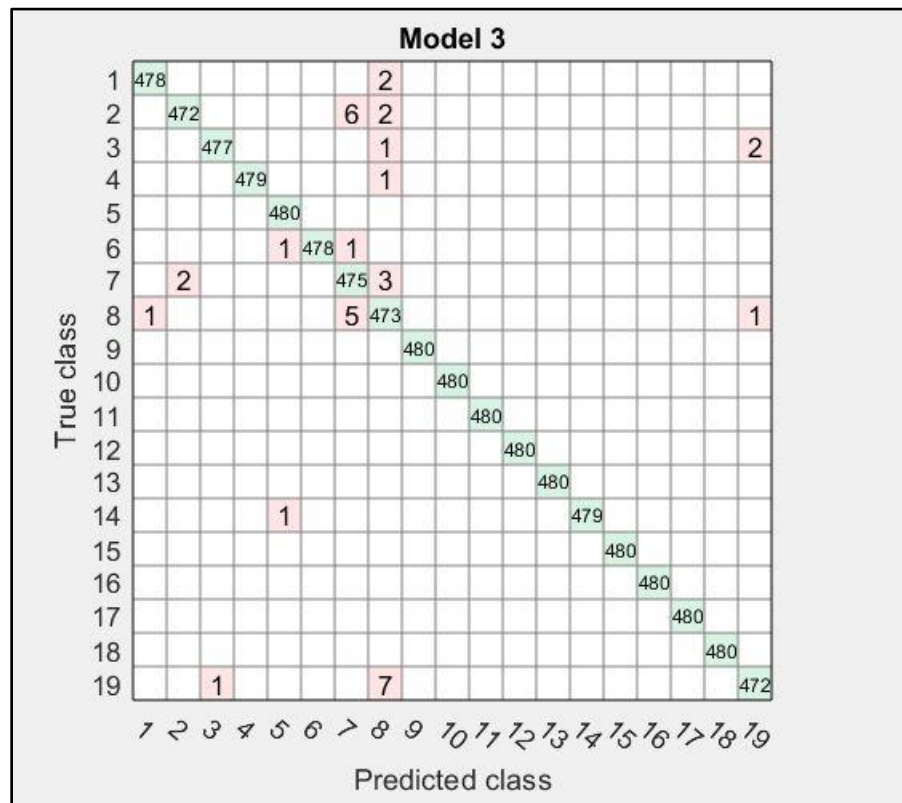


## VI. Cubic SVM

Mean Normalization is applied to the feature Space and then fed to Classification Learner with 30 Principal Components.

- Accuracy: 99.6%;
- F1-Score: 0.9996;

**Confusion Matrix for Cubic SVM:**



## Comparing Classifiers

- For all the Classifiers (except ANN), Mean Normalization is applied to the feature space as it gives a better accuracy than without applying Mean Normalization.

- Artificial Neural Networks gives the best output when both accuracy and Run-Time are considered. Though Quadratic SVM with Mean Normalization gives accuracy on the similar lines, the time taken for it to train is high compared to ANN. KNN with PCA gives 99.1% and without PCA gives 94%. Cubic SVM gives 99.0 with PCA and 99.6% without PCA.
- The Accuracy when PCA enabled is slightly lower than when it is not enabled. But due to very high dimensions, the training time is much more without PCA. Hence, though it reduces Accuracy to a certain extent, we consider the Algorithms with PCA enabled to reduce the training time.
- If we consider the F1-score, KNN gives the highest and best possible F1-score of 1. The F1-score of ANN and SVM are also close to 1, but considering Training time and Accuracy, we would rank the algorithms as:

*ANN > KNN > Cubic SVM > Quadratic SVM > Linear SVM > Fine Tree*

- Though Cubic and Quadratic SVM's are giving accuracy, the F1-score of a cubic SVM model is high compared to quadratic SVM and hence, Cubic SVM is considered the better among the two. Linear SVM also gives a high accuracy of 99.3% but its quite low compared to Quadratic and Cubic SVM.

ALGORITHM	Accuracy	F1-score
ANN (20 Hidden Layers)	99.97%	0.9996
Quadratic SVM (Mean Normalization and with PCA)	99.6%	0.9992
Cubic SVM (Mean Normalization and with PCA)	99.6%	0.9996
Linear SVM (Mean Normalization with PCA)	99.3%	0.9962
KNN (Mean Normalization with PCA)	99.1%	1



---

Fine Tree (Mean Normalization with PCA)	94.1%	0.9658
---	-------	--------

## Conclusion

Signal Processing and time series data can lead to engineering features and building machine learning models that predict which activity users are engaged in with 99% accuracy.

The model was able to learn which signals correspond to activities like walking or jumping for users.

We implemented and compared a number of different algorithms and ranked them in order of accuracy and training time.

Another aspect of activity recognition and classification that can be noticed is the normalization between the way different individuals perform the same activities. Each person does a particular activity differently due to differences in body size, style, and timing.

When Mean Normalization is done (scaling all features between 0 and 1) we see a better accuracy in SVM. However, it doesn't have a significant effect in case of ANN.