**Capstone Project**
Machine Learning Engineer
Nano degree

SARTHAK DARGAN
November 15th 2017

# Human Activity Recognizer and Tracker

## I. Definition

❖ **Project Overview :**

- The average office worker sits about 10 hours a day. There are all those hours in front of the computer, going through emails, making calls or writing proposals and eating lunch. And then there is hours sitting in front of the TV or surfing the web at home.

- Medical researchers have long warned that prolonged sitting is dangerous, associated with a significantly higher risk of heart disease, diabetes, obesity, cancer and depression, as well as muscle and joint problems.

- The aim of this project is to build a use case that take sensor information from our mobile phones and use that information to recognized and classify human activity like sitting running etc. and to analyse that information to give productive notifications, suggestions and recommendations

- Research Paper - http://ieeexplore.ieee.org/document/5581050/

- http://www.sciencedirect.com/science/article/pii/S1877050914008643/

❖ **Problem Statement :**

- The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

- The goal is to predict the likelihood that a human activity is from a certain class from the provided classes, thus making it a multi-class classification problem in machine learning terms and thereby analysing it.

- The goal is to train our model to classify human activity into these six classes

which could be incorporated in mobile applications for constructive feedback, notifications and analysis day routine.

- To solve this problem we would go with these major steps :-

1. Download the dataset and segment in training and testing datasets.
2. Study the datasets and configure additional parameters.
3. Try various models and evaluate each of them
4. Find the best model
5. Train the model
6. Visualize model accuracy, f1score, precision, recall.
7. Select the best model out of all.
8. Implement neural network
9. Compared accuracy for both best model and neural network
10. Visualize the comparison.

Six target classes are provided in this dataset:


❖ **Metrics:**

- **Accuracy** - Classification accuracy is the number of correct predictions made as a ratio of all predictions made

- **Precision** - precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances

- **F1_score** - considers both the precision and the recall of the test to compute the score

- **Recall** – recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances

**Why these metrics? -** We took Accuracy, Precision, F1 Score and Recall as metrics for evaluating our model because accuracy would give an estimate of correct activity prediction to all the activity predictions we made so far. Precision would give us an estimate about the positive human activity predicted value i.e. how much our model is giving relevant result. F1 Score gives a clubbed estimate of precision and recall. Recall would provide us the relevant positive activity prediction to the false negative and true positive human activity recognition results.

# II. Analysis

**Source:**https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones

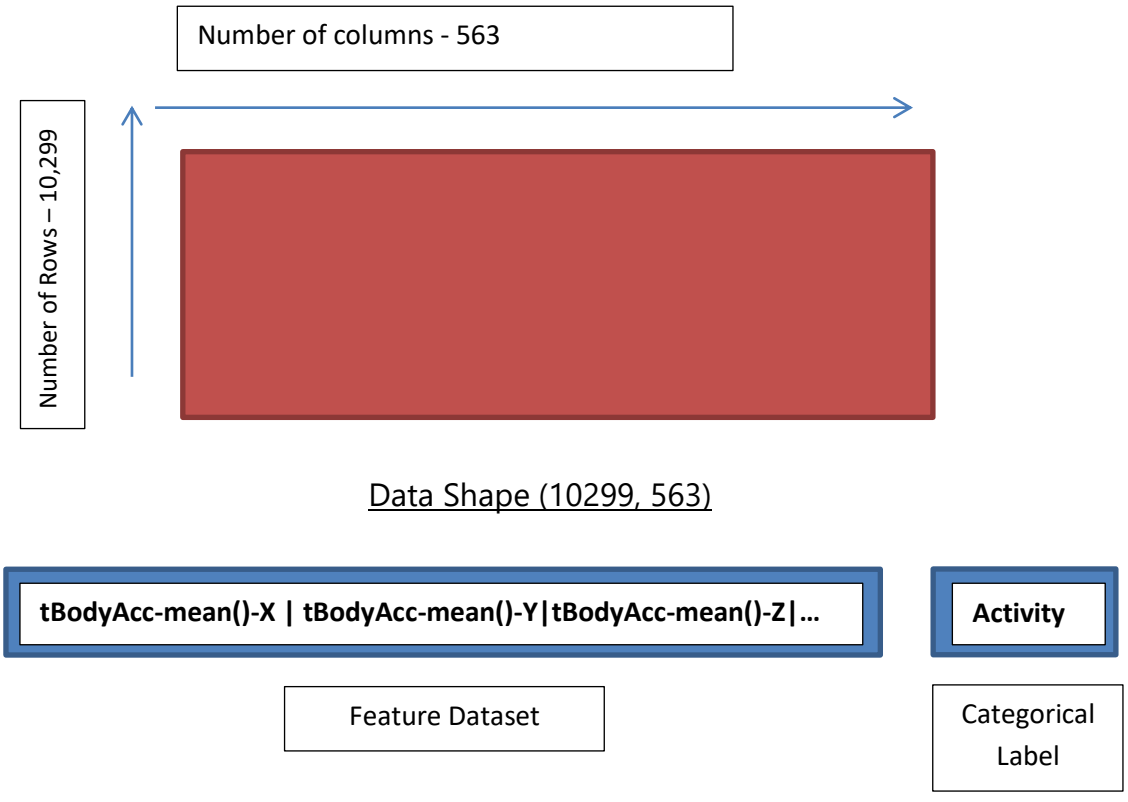
❖ **Data Exploration:**


*Video Recorded – Click on the image to watch video*

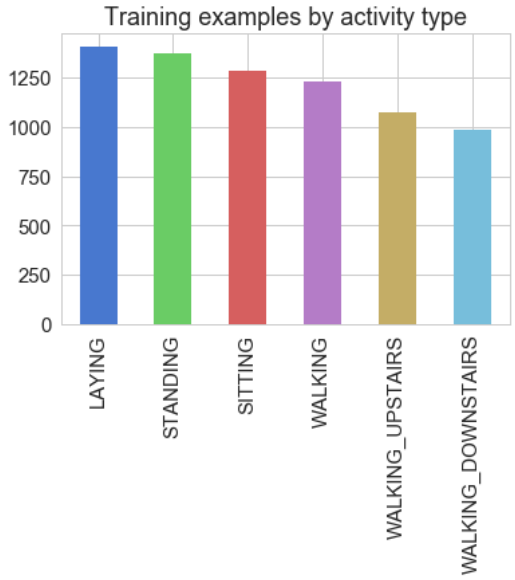
- Experiment for creation of Human Activity Recognition database is collected by 30 volunteers by mounting a Samsung s2 mobile phone with embedded accelerometer and gyroscope on waist.
- Samsung s2 mobile phone configured with 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz.
- Accelerometer and gyroscope sensory information were pre-processed by applying noise filters
- Sensory information sampled in fixed-width sliding windows of 2.56 sec and 50% overlap.
- The sensor signal was separated using a Butterworth low-pass filter into body acceleration and gravity.

- The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

1. Triaxial acceleration from the accelerometer (total acceleration).
2. Triaxial Angular velocity from the gyroscope.
3. A 561-feature vector with time and frequency domain variables.
4. Its activity label.
5. An identifier of the subject who carried out the experiment.



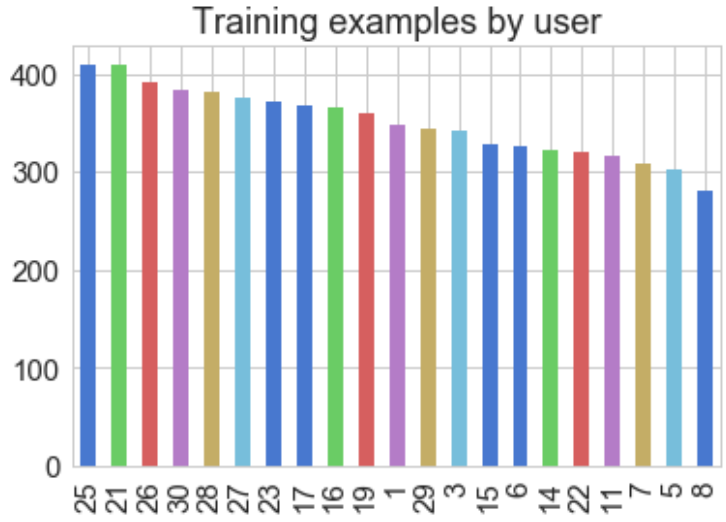

❖ **Exploratory Visualization:**




**Take Away -** It depicts the distribution of the data. For a good model, the dataset should be in regular fashion. Here we can infer that records related to laying is maximum, approximately matching to standing, then we have sitting, then walking ,then walking upstairs and then the least number of records for walking downstairs.

Figure 1

*Figure 1* -Depicts records distribution based on activity type considering training examples.




**Take Away –** As we visualize in in the visuals explained above [records distribution based on activity type] on the same way we need regular records distribution among the users who are taking this experiment so that records should not be biased to one user, it should be uniform. Here we observe that records are somehow uniform.

*Figure 2* -Depicts records distribution based on activity type considering training examples.

❖ **Algorithms and Techniques:**

- To solve this problem we would be working out with several Machine Learning algorithms and techniques.

- We will evaluate each model and the model which gives us the maximum efficiency based on evaluation model we would choose it and optimize it much more.

- We will be considering –

  - Decision Tree Classifier
  - Random Forest Classifier
  - Gaussian NB Classifier
  - K Neighbour Classifier
  - Support Vector Machine
  - Neural Networks

- **Decision Tree Classifier** – Decision Tree Classifier is used to solve supervised machine learning problem where after each level we divide the data according to a parameter. In our Human Activity Recognition we are solving a Supervised Machine Learning problem .**Pros:** 1. Easy to understand 2. Robust against skewed features set **Cons:** 1. do not work best with a lot of un-correlated features 2. Result can show biased traits.

- **Random Forest Classifier** – Random Forest Classifier is an upgraded version of Decision Tree Classifier; here the results accumulated are from multiple trees rather than one so it could be the selected model as it's an upgraded version of Decision Tree .**Pros:** 1. over fitted results are diminished 2. Feature Engineering could be performed **Cons:** 1. Hard to visualize as compared to decision tree 2. Often more biased to the class with more levels.

- **Gaussian NB Classifier** – Gaussian Naive Bayes is an upgraded version of naive bayes which based on fact each feature is independent of other and each feature have same importance; let's figure out how it these specs will effects our problem set **Pros:** 1. Fast 2. Less number of samples could show good results **Cons:** 1. makes a strong assumption 2. We lose information relating feature set.

- **K Neighbour Classifier** – K Neighbour Classifier takes averages of nearest neighbours while classifying a point, let's consider this property to classify human activity. **Pros:** 1. Easy to understand 2. Fewer assumptions **Cons:** 1. more testing time as compared to training 2. Highly sensitive.

- **Support Vector Machine - In page link Click to go !**

- **Neural Networks - In page link Click to go !**

❖ **Benchmark Model:**

- To create a initial benchmark model I would be considering 96% accuracy claimed by **Amit Patney on Kaggle discussion page** as the highest attained accuracy so far.

- He claimed to use SVM with boosting an ensemble method,

- I will be using technique as mentioned earlier and try to reach as close.

# III. Methodology

❖ **Data Pre-processing:**

- In this model luckily we have our feature in numerical format so not much pre-processing steps were required and the output label as categorical data.

- These Steps for followed as pre-processing steps-

    - Data is loaded from csv files.
    - Divided into training and testing dataset.
    - Shuffling of training and testing dataset.
    - Label and One Hot encoded the label resulting into 6 more columns.

- Tried reducing the dimensionality, but it doesn't helped much accuracy from the final model seems to be diminish; so idea of reducing dimensionality using PCA was dropped.

❖ **Implementation**

- After data pre-processing, dividing data into training and testing dataset, we have a question that to solve this problem statement which machine learning technique would be best suited.

- So to get to the answer, we plan to start with a couple of algorithms and see which algorithm gives better result.

- Once we get the algorithm which is best suited for the problem statement we would work to make it more refine and accurate.

- But before initiating with it! We would always be inclined to use a simple model as much as possible under similar accuracy.

- So the algorithms which we planned to test for our model are –

    - Decision Tree Classifier
    - Random Forest Classifier
    - Gaussian NB Classifier
    - K Neighbour Classifier
    - Support Vector Machine
    - Neural Network

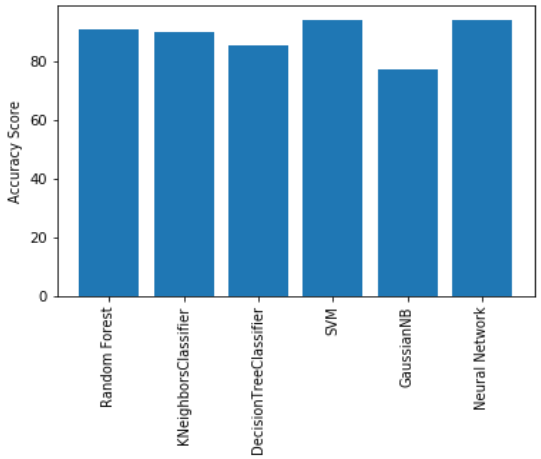
- We initially trained the model on each of classifier except neural networks.

- Results are as follows :

**Cross Validation Score on Training Dataset:**

| Classifier | Cross_val_score[AVG] |
|---|---|
| **Decision Tree Classifier** | 0.9383842987983666 |
| **Random Forest Classifier** | 0.9658606464155355 |
| **Extra Trees Classifier** | **0.968716347332843** |
| **Support Vector Machine** | 0.9405607267376771 |
| **K Neighbors Classifier** | 0.9650445414707481 |

**Metrics on Testing Dataset:**



- We observe Random Forest Classifier, K Neighbours, Extra Trees Classifier shows high scores on training dataset

- And we observe Random Forest Classifier, K Neighbours, Extra Trees Classifier and other classifiers doesn't show much good results on testing datasets.

- We can observe that algorithms shows different characteristics when testing and training datasets is concerned, which depicts there algorithms are over fitting.

- We can observe SVM [Support Vector Machine] shows good accuracy score when test data is concerned.

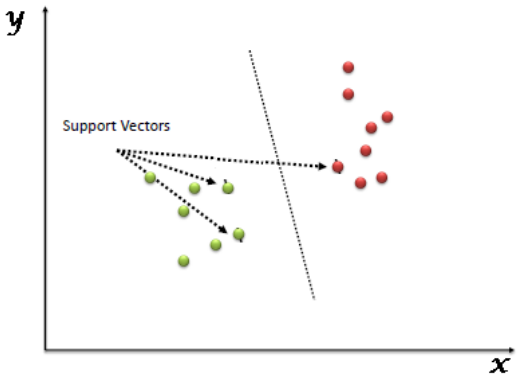
- So we will be opting for SVM as the selected model and we will compare this model for final face off with the Neural Network and we would select the model with better results.

**Source:** https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

**What SVM Classifier Is?**

- Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

- Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line)




Source Analytics Vidhya

**Source:** https://keras.io/

**Keras MultiLayer Neural Network-**

- Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

- Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).

- Supports convolutional networks and recurrent networks, as well as combinations of the two.

- Runs seamlessly on CPU and GPU.

KERAS MODEL STRUCTURE -

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_1 (Dense) | (None, 96) | 53952 |
| dropout_1 (Dropout) | (None, 96) | 0 |
| dense_2 (Dense) | (None, 72) | 6984 |
| dropout_2 (Dropout) | (None, 72) | 0 |
| dense_3 (Dense) | (None, 40) | 2920 |
| dropout_3 (Dropout) | (None, 40) | 0 |
| dense_4 (Dense) | (None, 30) | 1230 |
| dropout_4 (Dropout) | (None, 30) | 0 |
| dense_5 (Dense) | (None, 24) | 744 |
| dense_6 (Dense) | (None, 6) | 150 |

==========
==========
Total params: 65,980
Trainable params: 65,980
Non-trainable params: 0

**Challenging Part:** The most troublesome task was to decide how to initiate the problem; tackle the problem and come to solution i.e. which algorithm to consider and which metrics to consider.

## Refinement of Selected Model [Support Vector Machine] –

We will now work to tune Hyper Parameters; hyper parameters are those parameters which are not learned by the model itself, they are the one which are passed as parameters to the model, for ex C, kernel and gamma.

For refinement process we would be going for **GridSearchCV** on our selected model i.e. **SVM**

**Results Obtained [ Test Data ]:**

- Tuning hyper-parameters for accuracy

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 537 |
| 1 | 0.97 | 0.89 | 0.93 | 491 |
| 2 | 0.91 | 0.98 | 0.94 | 532 |
| 3 | 0.96 | 0.99 | 0.98 | 496 |
| 4 | 0.99 | 0.97 | 0.98 | 420 |
| 5 | 0.97 | 0.96 | 0.97 | 471 |
| avg / total | 0.97 | 0.97 | 0.97 | 2947 |

**Take Away –**

After hyper parameter tuning, now we are achieving **97%** precision, recall and f1 score from **94 %** that we achieved without hyper tuning parameter.

Hence model seems to be **Robust enough** as it is showing spectacular scores for unseen data.

# IV. Results

**Source:**https://en.wikipedia.org/wiki/Precision_and_recall

## ❖ Model Evaluation and Validation:

For initial evaluation of number of models except Neural Network, we could consider following as the parameter on which model evaluation will depend upon –

- **Recall Score**- the proportion of actual positive cases which are correctly identified.
- **Precision** - Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV).
- **F1 Score** - F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

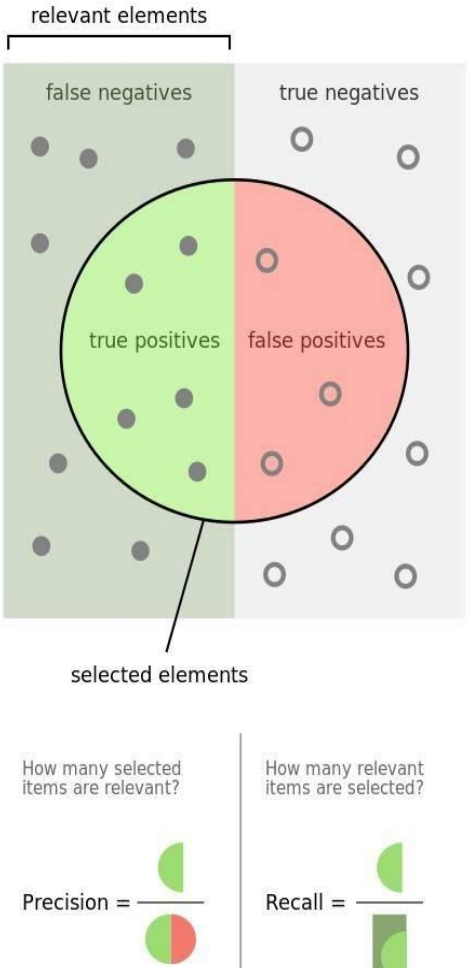



Image Source Wikipedia

DecisionTreeClassifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 537 |
| 1 | 0.83 | 0.76 | 0.80 | 491 |
| 2 | 0.80 | 0.86 | 0.83 | 532 |
| 3 | 0.83 | 0.89 | 0.86 | 496 |
| 4 | 0.85 | 0.83 | 0.84 | 420 |
| 5 | 0.82 | 0.77 | 0.79 | 471 |
| avg / total | 0.86 | 0.86 | 0.86 | 2947 |

Decision Tree Classifier Result

SVC

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 537 |
| 1 | 0.91 | 0.86 | 0.88 | 491 |
| 2 | 0.88 | 0.92 | 0.90 | 532 |
| 3 | 0.94 | 0.99 | 0.97 | 496 |
| 4 | 0.98 | 0.90 | 0.94 | 420 |
| 5 | 0.94 | 0.96 | 0.95 | 471 |
| avg / total | 0.94 | 0.94 | 0.94 | 2947 |

**Support Vector Machine Result**

GaussianNB

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.60 | 0.74 | 537 |
| 1 | 0.58 | 0.75 | 0.65 | 491 |
| 2 | 0.80 | 0.86 | 0.83 | 532 |
| 3 | 0.82 | 0.84 | 0.83 | 496 |
| 4 | 0.83 | 0.61 | 0.70 | 420 |
| 5 | 0.76 | 0.96 | 0.84 | 471 |
| avg / total | 0.79 | 0.77 | 0.77 | 2947 |

**Gaussian NB Result**

RandomForestClassifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 537 |
| 1 | 0.87 | 0.89 | 0.88 | 491 |
| 2 | 0.90 | 0.87 | 0.89 | 532 |
| 3 | 0.85 | 0.96 | 0.90 | 496 |
| 4 | 0.93 | 0.85 | 0.89 | 420 |
| 5 | 0.90 | 0.84 | 0.87 | 471 |
| avg / total | 0.91 | 0.91 | 0.91 | 2947 |

**Random Forest Classifier Result**

KNeighborsClassifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.99 | 1.00 | 537 |
| 1 | 0.91 | 0.79 | 0.85 | 491 |
| 2 | 0.83 | 0.93 | 0.88 | 532 |
| 3 | 0.85 | 0.98 | 0.91 | 496 |
| 4 | 0.94 | 0.79 | 0.86 | 420 |
| 5 | 0.90 | 0.89 | 0.89 | 471 |
| avg / total | 0.90 | 0.90 | 0.90 | 2947 |

**K Neighbours Classifier**

As we can analyse SVM i.e. **Support Vector Machine portrait better results** than any other classifier so we will consider SVM to be the selected model and now we will compare the results with Neural Network.

**Neural Network attains Accuracy Score of 94.12 whereas SVM attain accuracy score of 94.02 so we can go for either of techniques as both shows similar results.**

## Support Vector Machine [ Validation]

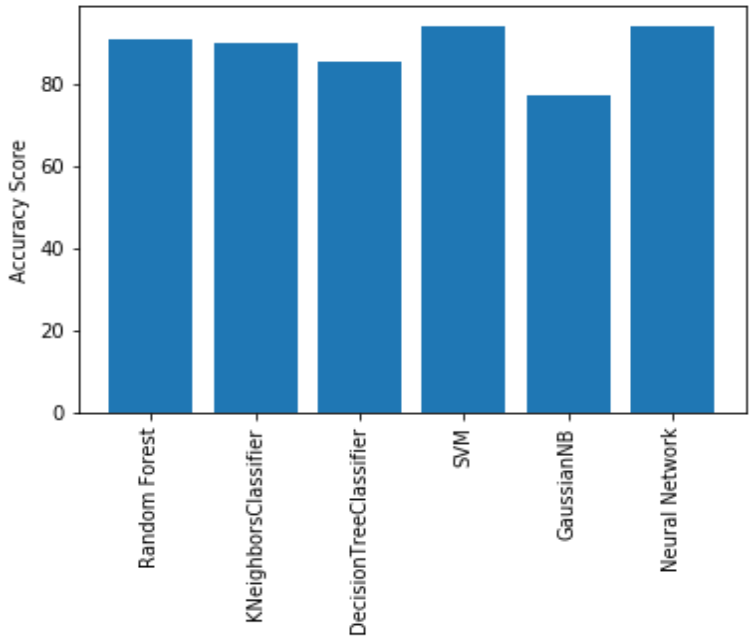
These beautiful insights give us a validated outcome that the choosen model - SVM is the eligible model for our problem domain. It's giving Precision Macro Mean Score - 0.94811323831, Recall Macro Mean Score 0.947051984049 and R2 Score - 0.977392570478; which are good scores and supports our decision while selecting SVM as the final model.

❖ **Justification:**

- Till now highest accuracy claimed is around 96% posted by Amit Patney on Kaggle discussion page.
- We attained accuracy more than 94 %. Close!
- We reached very close to the accuracy till date.
- **As a final model both SVM and Neural Network attained similar accuracy approx. so we can use any model as a solution to the problem statement.**

# V. Conclusion

❖ **Free-Form Visualization:**




**Take Away –**

This visual gives accuracy score comparative analysis of following algorithms Random Forest, K Neighbours Classifier, Decision Tree Classifier, SVM , Gaussian NB, Neural Networks. We found that SVM shows better accuracy score than other and Gaussian NB shows least.

**Accuracy Score Comparision**

❖ **Reflection:**

- This is what we did to reach to our final solution to the problem statement –
  1. Download the dataset and segment in training and testing datasets.
  2. Study the datasets and configure additional parameters.
  3. Try various models and evaluate each of them

4. Find the best model
5. Train the model
6. Visualize model accuracy, f1score, precision, recall.
7. Select the best model out of all.
8. Implement neural network
9. Compared accuracy for both best model and neural network
10. Visualize the comparison.

- For me the most interesting topic in this project was the problem statement and dataset itself. Now with the help of this model I could identify human activity and give insights and further add on.
- Difficult situations in this project were; I got struck at various segments of the project when my code didn't worked and errors were reported. So I need to figure out the problems and rectified the same.
- Yes! According to me it attains a good accuracy score; It works well above what I expected as when I initiated this project.

**Source:**https://deeplearning4j.org/lstm.html

❖ **Improvement:**

- Improvement that I think; could be there by using **Long Short Term Memory network [LSTM]** a **recurrent neural network** on the data to learn and to recognize the type of activity that the user is doing. Many to one RNN architecture taking series of input vectors and output vectors with probabilities to help in classification as one of the 6 classes. LSTM could give good results, as it is well-suited to classify process and predict time series use cases where, state change with time.
- This project could be converted into an Android or IOS app and the result could be used for giving insights, feedback, suggestions, recommendations etc.