CS 205 - Health Analytics

Activity Recognition

Akshay Sharma (504946035)

Nikhil Thakur(804946345)

Rahul Dhavalikar (205024839)

Satya Vasanth Reddy (405029459)

Introduction

The availability of a system capable of automatically classifying the physical activity performed by a human subject is extremely attractive for many applications in the field of healthcare monitoring and in developing advanced human-machine interfaces. By the term physical activity, we mean either static postures, such as standing, sitting, lying, or dynamic motions, such as walking, running In this project we aim to predict the Human Activity from the sensor readings of Sony smartwatch. We are primarily using accelerometer and gyroscope readings for our modelling.

Motivation

The use of on-body wearable sensors is widespread in several academic and industrial domains. There is great interest in their applications in ambulatory monitoring. The first step in predicting the human behaviour and understanding the human body's status is to be able to classify the activity the human wearing the device is performing. Activity monitors are now capable of storing a large amount of raw acceleration data, allowing examination of detailed features from the monitor sensor output. Using machine learning techniques to process these data can detect activity types and patterns of physical activity as well as intensity and energy expenditure. Since the eruption of IoT and widespread popularity of wearable devices the task of classifying human action and there by using this information to draw meaningful conclusions has become trending. Research teams in the industry and academia have come up with various predictive models and achieved considerable success but there is still scope for improvement.

Challenges

There are various challenges in this project. They are described as below,

- We have to be very careful while collecting the data, updating the proper label while performing the appropriate action.

- Some unwanted movement can result in bad data and affect the robustness of our model to a very high extent.
- Proper care has to be taken regarding the settings of the watch including the sampling frequency of the sensors.
- If both the accelerometer and gyroscope data are to be used for analysis, they should be properly combined as the sampling frequency is different for both the sensors.
- To make the model robust, training data should be collected from variety of different people with diverse pattern on performing various activities.
- Feature selection is something that can be very challenging, a lot of time goes in experimenting with various time domain and frequency domain features based on the sensors.
- Also, the computation complexity can increase considerably with incorporation of huge amount of data and complex models like Recurrent Neural Networks(RNN).

Data Collection

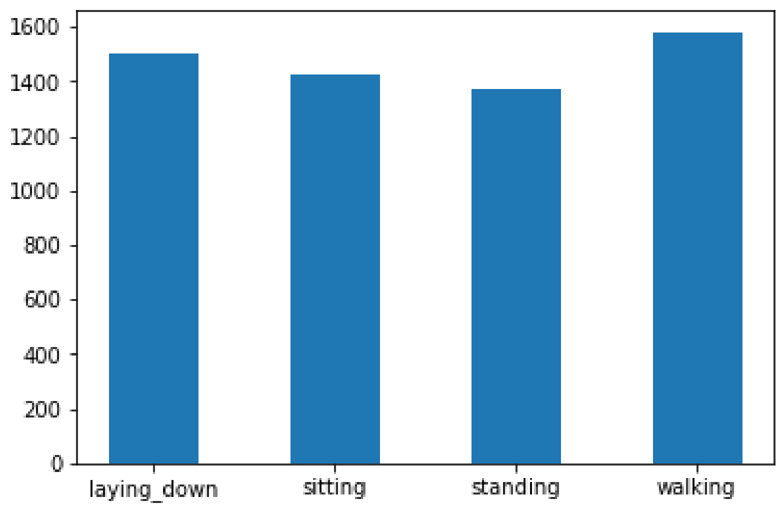
In our problem we have considered following 4 activities as labels,

- Sitting
- Standing
- Laying Down
- Walking

We have collected roughly around 100 minutes of data for each activity from 8 different people. The reason for collecting data form different people as mentioned above is reduce the variance of the model and make it more robust. We have tried to keep the data for all the labels balanced so that the model learns much more information from the training data and doesn't become biased to one particular label and performs poorly on unseen dataset.

The data has to be arranged in windows of some fixed seconds and features are calculated for that window. We have taken the window size as 4 seconds, and we get 250 readings per second. Hence, we get 1000 readings for each window.

One more thing is to have a balanced dataset for each label. This helps to train the model better than skewed datasets. As the data collection phase is also in our hands, we have collected a nearly balanced dataset for each label. The below histogram shows our distribution of training data for all the labels.



Training, Validation & Test Sets

Another important thing is to have data from multiple subjects so that the models are not biased towards one person and can generalize over anyone. We have collected data from multiple people and divided it into training, validation and test sets. We use unique people in each set and do no have the same people under training and testing sets.

We have used training data to train our model. Training data is collected from around 8 people.

We have used the validation set to tune the hyperparameters of our models so as to get optimal results.

While the final testing is done on an entirely different set taken from different people not involved in training and validation to ensure that the models are robust and can be generalized.

Data Preprocessing

Once we have gathered all the data, we need to process it before we can do feature extraction on it. Many a times during data collection it was observed that there was some delay or confusion while updating labels on the watch. We did not want this noise to affect the quality of our data, so to get clean data we have trimmed the first and last 10 seconds of data that we have got during each label update. This ensures that the data we will be training on is clean.

The accelerometer and gyroscope data is being sampled at different times and different frequencies. So the timestamps found in both the data csvs are not always the same. So first we need to correlate the data we have received from both the sensors. For this we have used an intelligent algorithm to assign a gyroscope reading to its corresponding accelerometer reading.

Feature Engineering

Once we have collected We have done some considerable research regarding the time domain and frequency domain features. We tried to select the features common among most of the research papers for our model. We have not selected a lot many features, the reason being, large number of features may cause the model to overfit and also it may be difficult later to get a high level idea of which features are important for the purpose.   Following table shows the features we have used with their description.

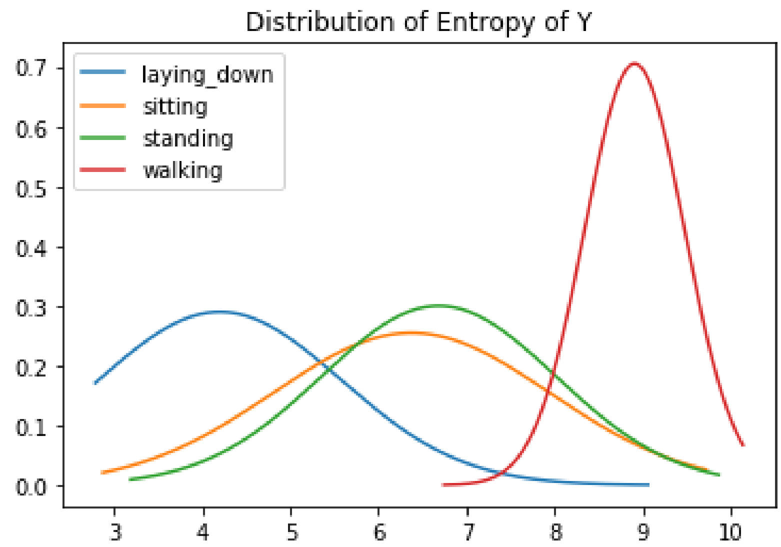
Features	Description
Mean of X,Y and Z	Mean of the Signal values along the 3 axes
Median of X, Y and Z	Median of the Signal values along 3 axes. **Importance**: Mean may not be a representative value because of the presence of outliers or skewed data. In such a case , median works better.
Variance of X, Y and Z Standard Deviation of X,Y and Z	**Importance:** Variance and Standard deviation are measure of spread of distribution around the mean.
Signal Magnitude Area	Normalized integral of values in waveform graph. **Importance**: The classification of rest and movement can be done using Signal magnitude area (SMA) by fixing a threshold
Maximum and Minimum X,Y and Z values	Minimum and Maximum value of signal along the 3 axes

Power of X, Y and Z	Power is sum of squares of magnitudes divided by length of the signal. **Importance:** Indicates the strength of the signal
Skewness of X, Y and Z	**Importance:** Skewness can be quantified to define the extent to which a distribution differs from a normal distribution. If the mean is less than the mode, this situation is called negative skewness.
Kurtosis of X, Y and Z	Kurtosis is a measure of the combined sizes of the two tails. It measures the amount of probability in the tails. The value is often compared to the kurtosis of the normal distribution, which is equal to 3. If the kurtosis is greater than 3, then the dataset has heavier tails than a normal distribution (more in the tails). If the kurtosis is less than 3, then the dataset has lighter tails than a normal distribution (less in the tails).
Entropy of X, Y and Z	Information entropy is defined as the average amount of information produced by a stochastic source of data.
Interquartile range of X, Y and Z	**IQR:** It is equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1. In other words, the IQR is the first quartile subtracted from the third quartile; these quartiles can be clearly seen on a box plot on the data.

Cross-correlation of XY, YZ and XZ	It is a measure of similarity of two series as a function of the displacement of one relative to the other. This is also known as a sliding dot product or sliding inner-product. It is commonly used for searching a long signal for a shorter, known feature.
Mean roll	Rotation around the front-to-back(X) axis is called roll.
Mean pitch	Rotation around the side-to-side(Y) axis is called pitch. **Importance:** Pitch and roll, from the acceleration readings are used to investigate how motion, orientation and rotation changes improve the recognition accuracy.
Zero crossing rate of roll and pitch	The zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.
Normalized FFT coefficients for X, Y and Z.	FFT coefficients are the signal values in the frequency domain.
Energy of X, Y and Z	Energy is the square modulus of the coefficients.

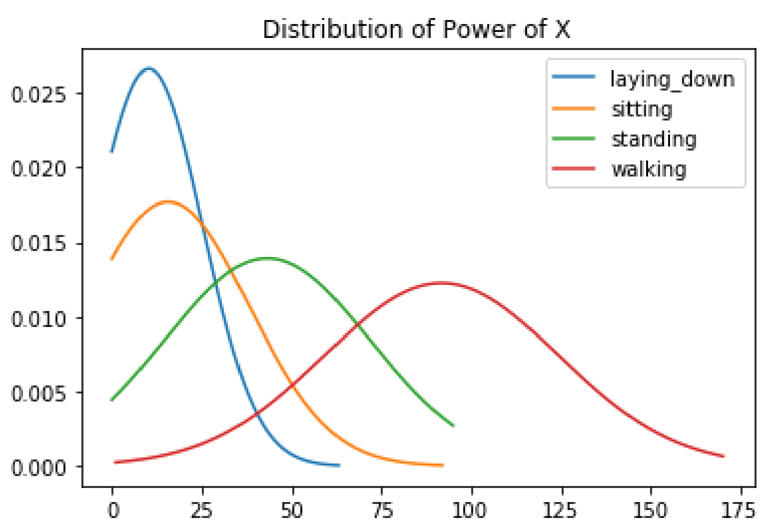
Some Insights

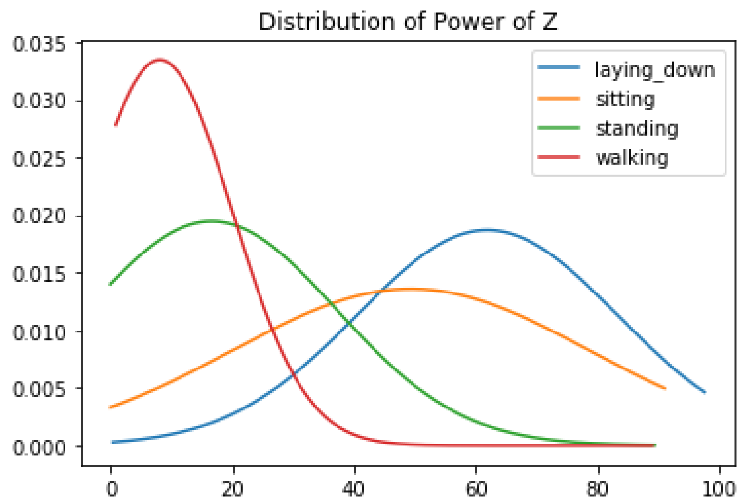
During our research, we found papers which were using a lot of features but none of them actually elaborated on the importance of each feature individually and how it impacts the classification task at hand. So we tried to get some more insights into the features we have extracted and analysed the impact of couple of features on our data. We plotted the distribution of some of the features to analyse their impact on the classification. As you can see in the figure below, we have plotted the plot for Entropy.



Entropy is basically a degree of randomness or chaos. During walking, there is going to be a lot of changes going on as compared to laying down where the movement happening is minimal. In terms of entropy this means the activity of walking will have a high entropy as compared to the activity of laying down. This can be seen cleared in the above graph that how walking and laying down are clearly separated. Also sitting and standing would lie somewhere in between these two ends. Just by looking at the above distribution a human can differentiate between the activities and will also be a good feature for training a machine learning model.

Similar example can be seen for the Power on X and Z axis





Models Experimented With:

The features extracted from the previous step are given as inputs to a classification model along with the input labels collected. The models we have experimented with are listed below.

- **Logistic Regression**

  Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

- **Decision Tree**

  A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

- **Random Forests**

  Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

- **Support Vector Machines(SVM)**

  Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

- **K-Nearest Neighbours (KNN)**

  In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- **Multilayer Perceptron (MLP)**

  A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable

Our novel approaches

Along with the traditional models, we have experimented with a few other methods as well.

1) RNNs and LSTM
2) CNN Image Embedding
3) Model Ensemble

RNNs and LSTM

Recurrent nets are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or numerical times series data emanating from sensors, stock markets and government agencies. Like most neural networks, recurrent nets are old. By the early 1990s, the vanishing gradient problem emerged as a major obstacle to recurrent net performance. LSTMs help to overcome this obstacle and preserve the error that can be back-propagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely.

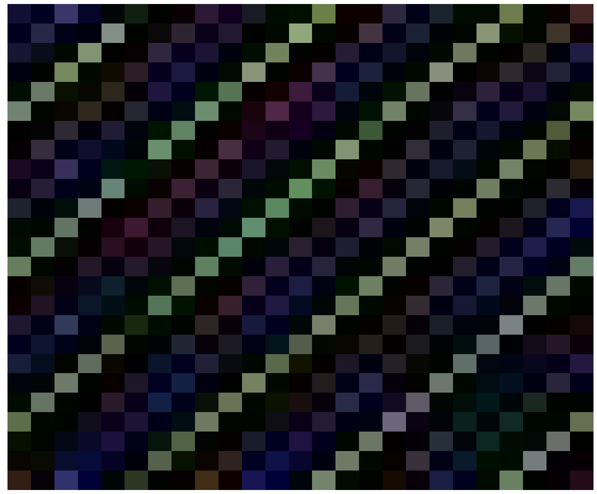
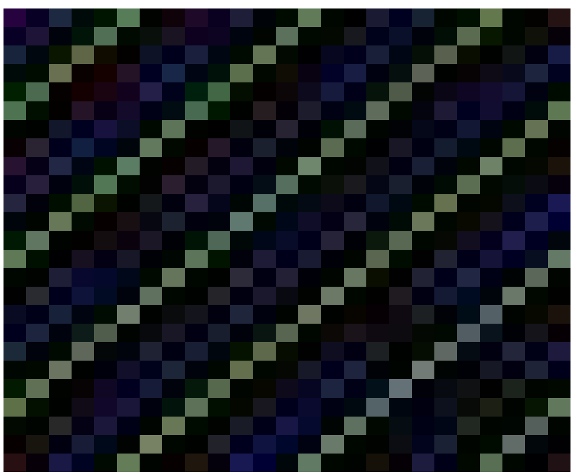
Due to its usage in the analysis of time series data, we tried to apply RNN's to this problem as the data available is a time-series data.

CNN Image Embedding

This is our novel approach. The motivation came from the belief that there must be some pattern in the signals which is different for each activities. But how can we figure out from the humongous data? May be if we see them? Yes. That's our motivation. We wanted to visualize the 49 dimensional data with 10 million entries. That seemed quite challenging but we figured out a way. What if we can represent each data element as a pixel? If yes, we could take every t second window and create an image from it for visualization. But how do we represent 49 dimensions as a pixel of RGB values. The first thing that occurs is to use SVD and reduce the dimension. But that doesn't turn out to be valuable. We need to find out the 3 dimensions where the data from one class or activity varies with another class. SVD simply finds the 3 dimensions that the entire data varies. The appropriate algorithm to use is LDA. It finds the required 3 dimensions and transforms the data into the 3 required dimensions.

Once we get the 3-D data, we transformed it to the range 0-255. We faced another challenge while transforming. Most of the data liked between -3 and 3 where there are outliers till -30 and 30. So if the values are transformed directly to 0 -255, there wouldn't be much difference between the values of different actions. We took data that is only in the range of 1 standard deviation from mean. The data outside the interval is simply replaced with the boundary values and a linear transformation is applied.

After that, we partitioned the data into 625 entries each and arranged them in a in-to-out spiral fashion. This resulted in a 25 X 25 image. We stored the images from the trained data in different folders for each action. Below are the images for the different activities.

Sitting	Walking
	
Laying Down	Standing
	

Model Ensemble

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. These models combine several machine learning techniques into one predictive model in order to decrease variance(bagging), bias (boosting), or improve predictions (stacking). This led us to try the ensemble model to approach to solve this problem. We observed from the performance of the previous classifiers that walking is classified with a high performance and hence we tried an approach similar to ensemble of classifiers. We use a combination of 2 classifiers.

- The first classifier is basically a binary classifier that classifies the input to walking and not walking.
- The second classifier is a multi-class classifier that takes the input and classifies it as **standing, sitting and laying down** in case input is classified as non-walking.

Commonly, the classifier used are Decision trees but experiments can be done with other classifiers too. This approach can be extended to even more classifiers for more activities.

Results and Comparison

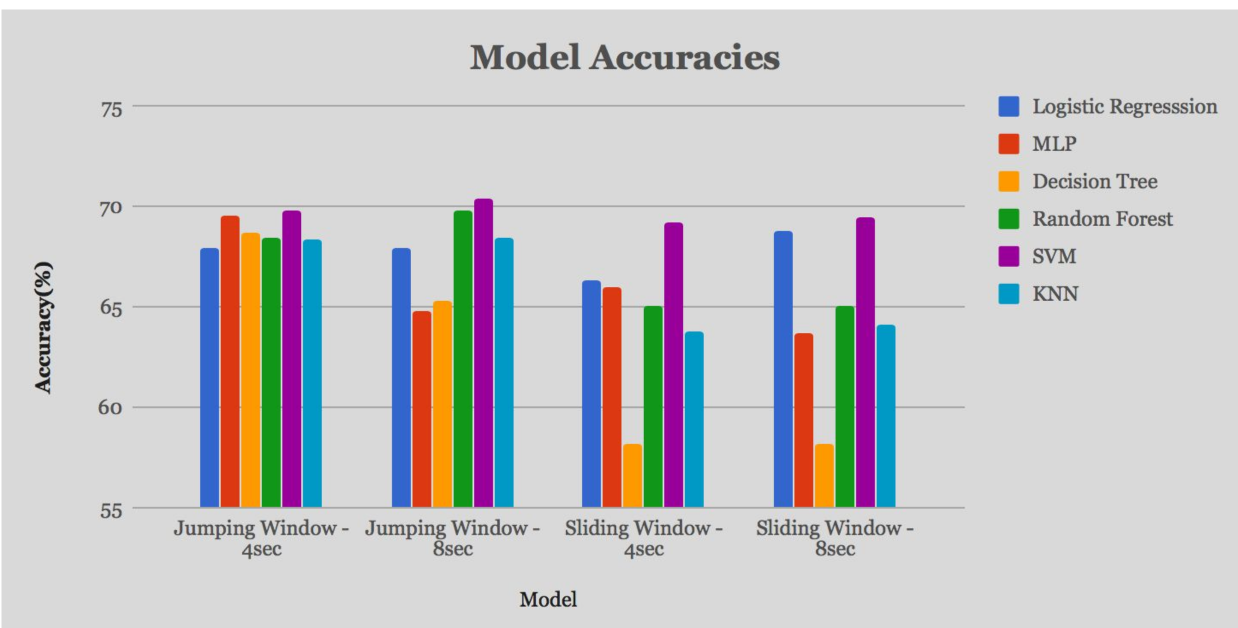
We have performed quite a lot experiments with the how the windows are formed and the models used.

The 2 types of window formulation we have used are,

- **Sliding Window**: In this method, the windows are slided on the data with some percentage of overlap. We have used 50% overlap in out modelling.
- **Jumping Window:** There's no overlap in this method, just start the next window after the end of the previous window.

While on the window duration, we have experimented with using a 4sec window and a 8 sec window.
So we have basically tried with combinations of the above 4 things and compared the results using the different models we have talked about.

Following are the results for the various combinations we tried out



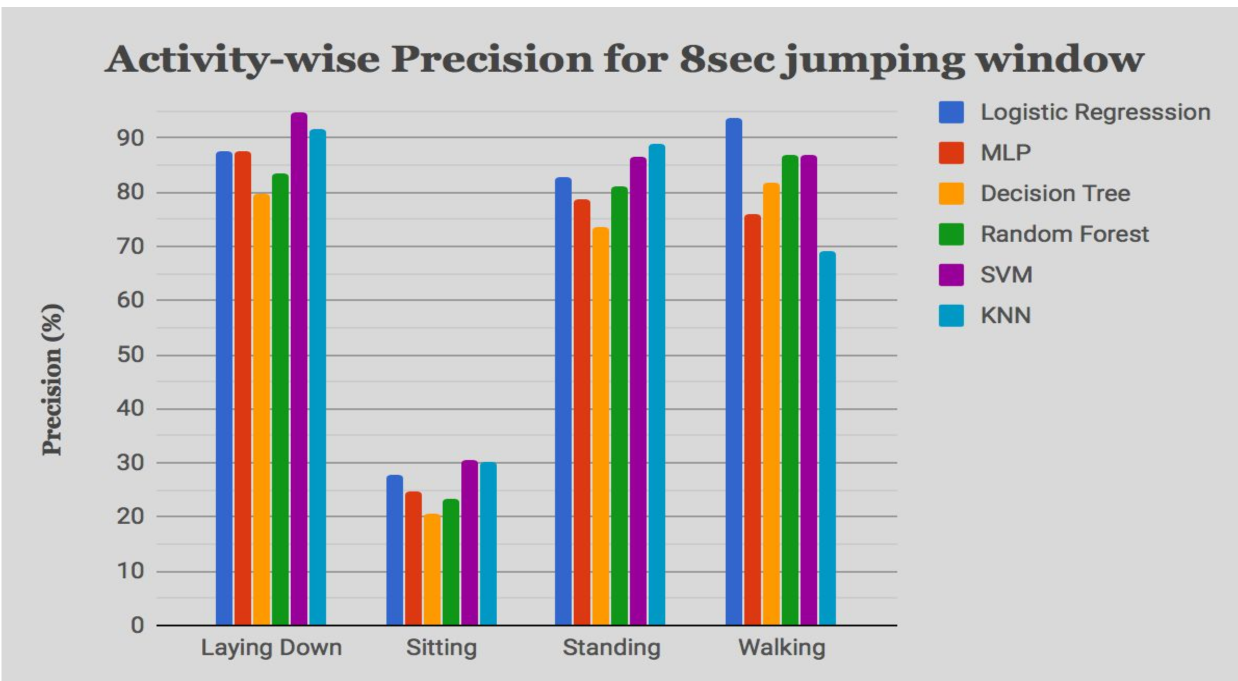
Observations:

- We see that nearly all the models in a particular scenario perform equally well. This means that our features are robust and are not biased towards a particular model.

- In nearly all the scenarios, the SVM model is performing the slightly better than the others.
- The jumping window methods are performing slightly better over the sliding window methods.
- Similarly the 8sec window durations are performing better than the 4sec windows. This was expected as we have a better understanding of an activity in a larger window. However the performance increase is not too huge and in applications where lesser latency is required, shorted window duration of 4sec can also be used without much loss in performance.
- Among all the models and methods, the SVM model with a 8sec jumping window method works the best with an accuracy of 70.37% on the testing set.

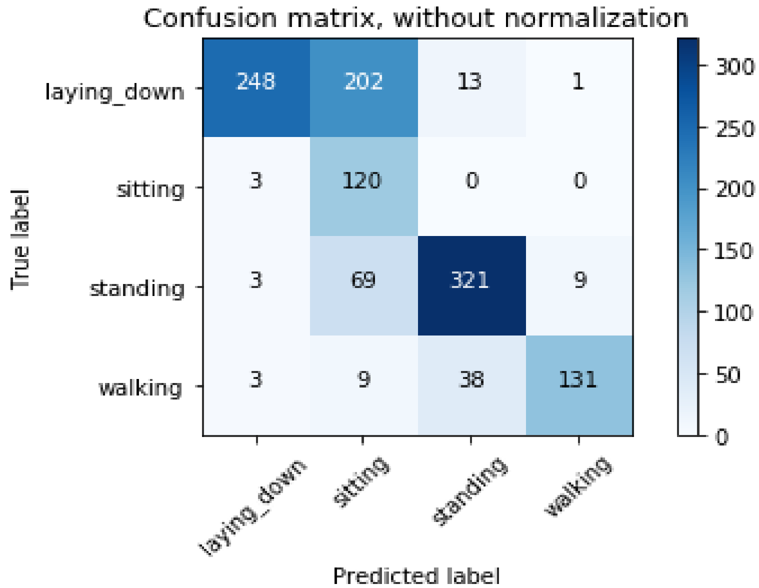
Activity wise results:

We have also calculated an activity wise distribution of precision of each model.



Observations:

- We see that the 3 activites walking, standing and laying down have a good precision score while the sitting activity has a relatively lower score. The models are performing very good overall but the overall accuracy is being lowered due to one of the activities.
- Some more insights can be seen from the confusion matrix where the sitting activity is not always predicted correctly.



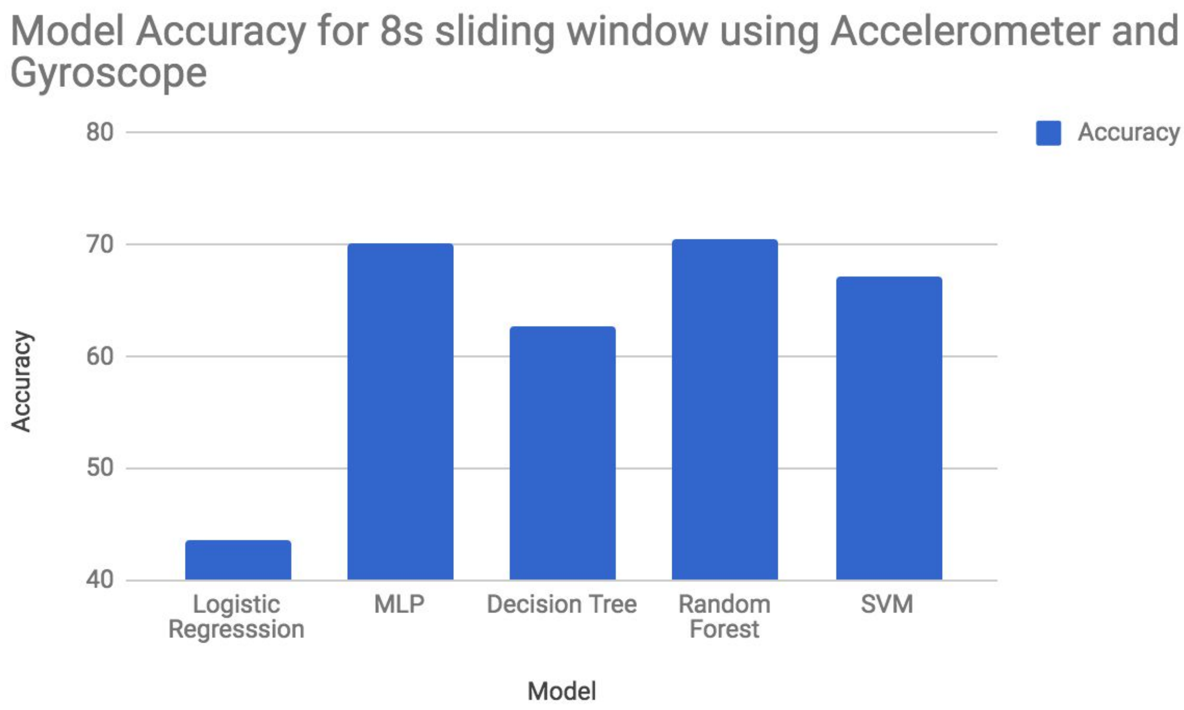
Accelerometer + Gyroscope

We have also experimented with using both the accelerometer and gyroscope data. We first however have to club the data correctly so as to have correct mapping between accelerometer and gyroscope.

We have used similar data preprocessing, feature extraction methods and traditional modelling methods as described above.

We have experimented with the 8sec sliding window method for this task.

The results for this are compiled below.

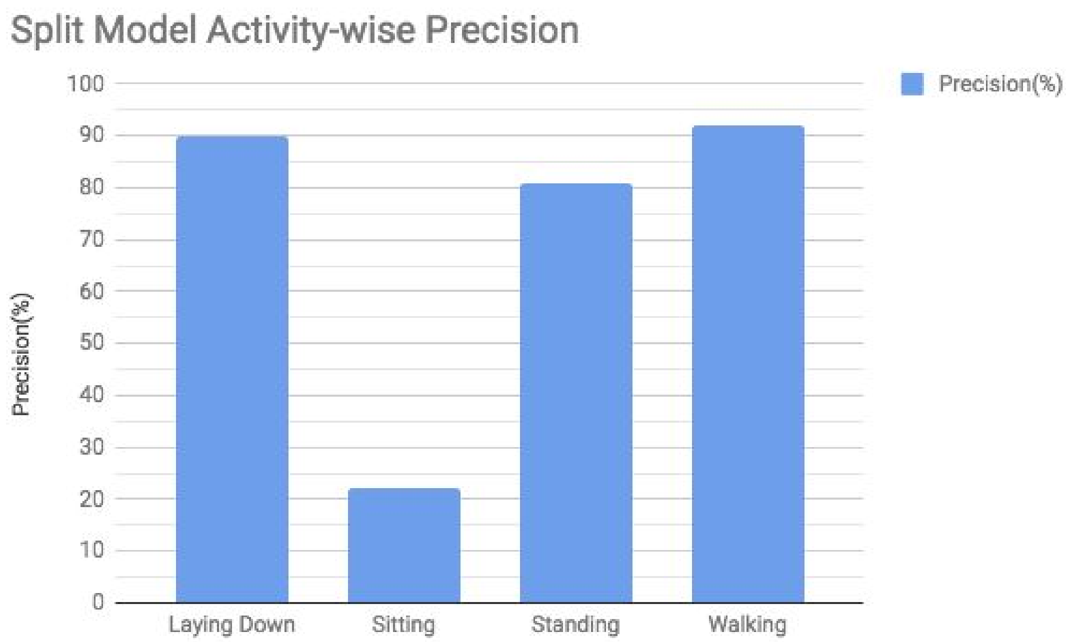


Observations:

- We get similar results for the models with the MLP model getting the best results this time.
- We do not get any huge performance boost by adding gyroscope data as compared to the accelerometer data.

Split Model Activity wise precision:

- We used this method of modeling to first differentiate between walking and non-walking activities.
- We have calculated the class wise precision for this task to compare it with our traditional models.
- On observing the graphs below, we see that we get similar results with high precisions on standing, walking and laying down activities and a lower precision for the sitting activity by using this method of modelling as well.



Conclusion

We have successfully built a model to identify the activity of a person using the data gathered from smart watch.

- Data collection should be done from different sources so that the models do not overfit a particular person and can work on anyone while testing
- Accelerometer and gyroscope data can be used for recognition of activities with high accuracies.
- Feature extraction can be done in the time as well as frequency domains to get relevant features.
- The window duration and overlap between the windows plays an important role.
- With a good feature set, the classification models perform equally well, with SVM model performing the best.

References

[1] Mannini, Andrea, et al. "Activity recognition using a single accelerometer placed at the wrist or ankle." Medicine and science in sports and exercise 45.11 (2013): 2193.

[2] Liu, Shaopeng, Robert Gao, and Patty Freedson. "Computational methods for estimating energy expenditure in human physical activities." Medicine and science in sports and exercise 44.11 (2012): 2138.

[3] Rosenberger, Mary E., et al. "Estimating activity and sedentary behavior from an accelerometer on the hip or wrist." Medicine and science in sports and exercise 45.5 (2013): 964.

[4] Tapia, Emmanuel Munguia, Stephen S. Intille, and Kent Larson. "Activity recognition in the home using simple and ubiquitous sensors." International conference on pervasive computing. Springer, Berlin, Heidelberg, 2004.

[5] Mannini, Andrea, and Angelo Maria Sabatini. "Machine learning methods for classifying human physical activity from on-body accelerometers." Sensors 10.2 (2010): 1154-1175.

[6] Ferscha, Alois, et al. "Orientation sensing for gesture-based interaction with smart artifacts." Computer communications 28.13 (2005): 1552-1563.

[7] Mannini, Andrea, et al. "Activity Recognition in Youth Using Single Accelerometer Placed at Wrist or Ankle." Medicine and science in sports and exercise 49.4 (2017): 801-812.