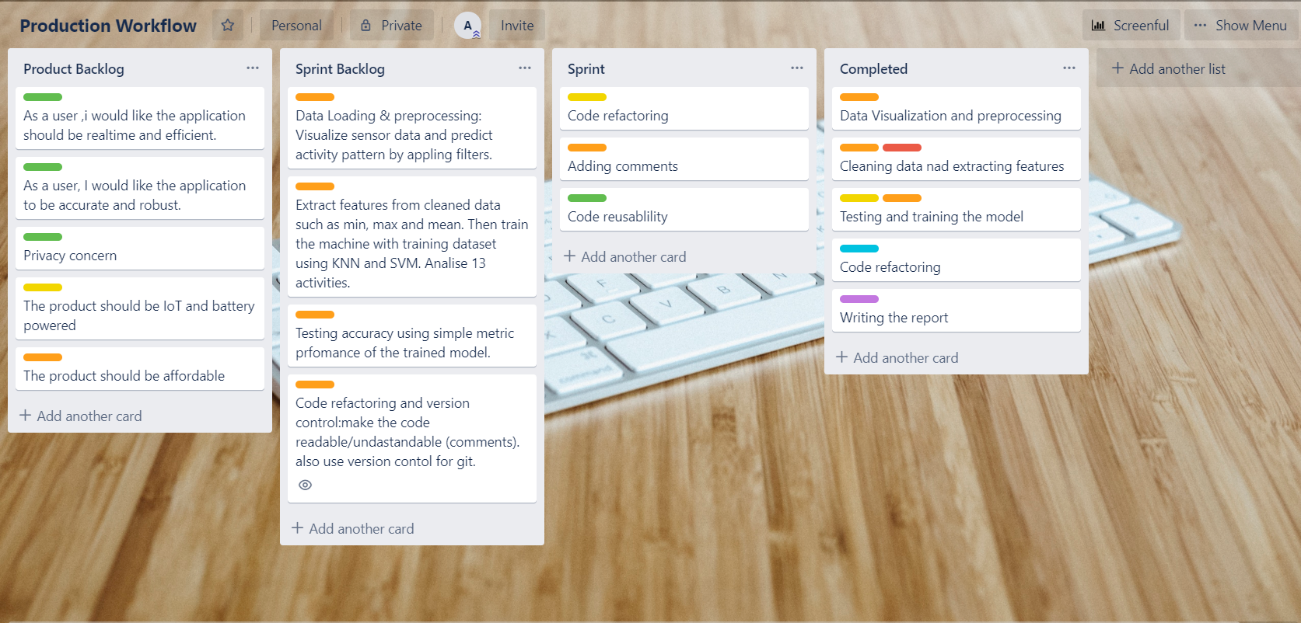
**Detecting Human behaviour/habits using sensors and machine learning**

**Introduction:**

With the growing number of elderly people leaving alone, it is important to know their wellbeing without interfering into their daily life’s. This is now possible thanks to machine learning and internet of thing (IoT), where an IoT device can be placed on a target person to collect data for predicting abnormal behaviour to alert authorities for immediate medical attention. Further, this technology has been adapted by large corporation like Apple in their wearable electronics such as smart watch, that can monitor heart rate, location, hard impacts and alert emergency services. Furthermore, such IoT implementation can be helpful to detect wear level on large infrastructure such as bridge, buildings, etc. to avoid catastrophic failure using sensor nodes in a mesh network. Also, this technique can be cost elective to determine when the infrastructure needs maintenance done by visualizing the data and looking for abnormalities in reading. In this day and age, it is difficult for a medical practitioner to keep track of their patients and check if they following the recovery guideline by the doctor. Such technology can help to determine how often the patient has been physically working out or detect bad behaver patterns such as amount of sugary drink the person has been drinking. Also, such device can be helpful with human health research, by collecting information from various people across the world comparing to find similar traits that can result in fast diagnosis and recovery avoiding certain death. Besides that, machine learning can save countless number of human life’s by detecting driver fatigue and awareness using Electroencephalogram (EEG) and cameras for facial analysis. The goal of this project is to make IoT application to recognise human activity by practising software development life cycle. There is data for 19 participants and we need to clean the data and extract features such as activities using complex matrix.

****

**Design Approach:**

The development approach for this project was done using agile software development cycle, that has been proven to be important to bring products to the market quickly. Also, this platform allows end-users to collaborate with developers in a team environment by refine and improve the features of the products. This method can also be beneficial because it can track the daily activity and amount of work that has gone into the project hence get paid accordingly. Furthermore, with the features such as burndown charts, we can get approximate time frame that the product will complete and delivered to the client. This project consisted of four sprints; each sprint had to be completed within seven days.

**Sprint 1: (week 1)**

The first sprint consisted of loading all the 19 datasets into the project sequentially. The 19-dataset represent 19 individual and each individual had 13 activities. The accelerometer and gyroscope data for wrist sensor was used for this project to predict human activity. Both accelerometer and gyro data were plotted and visualized before and after filtering to make sure our filters worked properly and the data is getting cleaned for feature extraction. The visualization and filtering program looped through all 13 activities, however since there were many graphs for each activity, I will only include the graphs for sitting activity only.

**Sprint 2: (week 2)**

For second sprint, we had to take the cleaned data and extract features such as min, max mean, time domain, frequency domain and amplitude. Also, we had to test how different features extraction impact performance of the system because some application such as self-driving cars need real-time feedback to make decisions. Furthermore, the testing and training dataset was extracted for machine learning model such as KNN and SVM used to recognise human activity. New feature set was added to increase the accuracy such as standard deviation, mean and variance.

**Sprint 3: (week 3)**

Sprint 3 contained the testing approach for the model. A confusion matrix was used to analyse each activity and performance of the model. Also, the increase the accuracy we replaced the standard scaler to MaxAbsScaler. Furthermore, we checked the accuracy in KNN and SVC model and the results were accurate to 84%.

**Sprint 4: (week 4)**

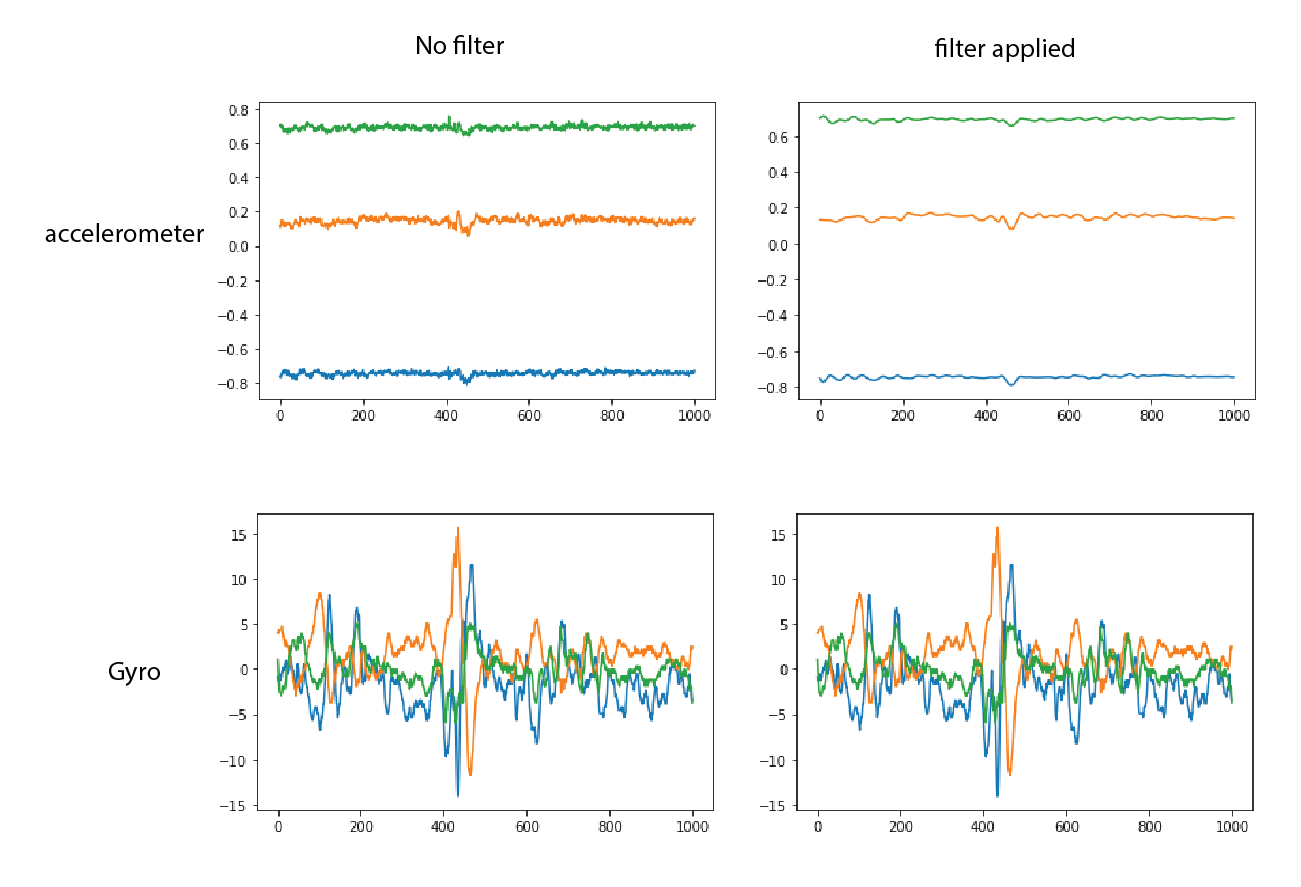
The last sprint was to refactor the code because it was smelling. Some of the techniques used to improve code refactoring was adding comments, making the code reusable, avoid code duplication, the code should be expandable, avoid long method etc. further, all the work on this project should be tracked by GitHub version control.

**Sprint Burndown Chart**

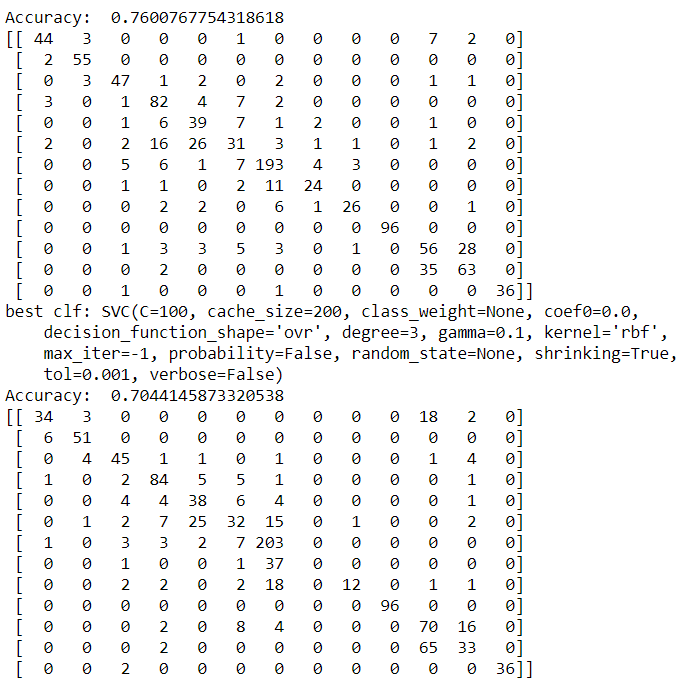
The weekly burndown chart consisted of activities that had to be completed every week to ensure the project finished in time and well organised. The total working hours for this project was 112 hours and each day required at least 4 hours of work to complete. The graph also displayed the amount of work and effort remaining for this project.

**Implementation:**

The filtering of data was done using lowpass filter to remove larger noise from frequency to time domain and vice versa. The feature extraction consisted of three elements min, max and mean to create training and testing data. Training model such as KNN and SVC was used to generate complex matrix to evaluate how accurate the data was for each activity.

**Results:**

The graphs above demonstrate the visualization of sitting activity before applying the filter and after. This process was done for all 13 activities to ensure the data was cleaned and ready for feature extraction. The x-axis of the graph was scaled to thousand samples for viewing between 7000 and 8000 samples. The gyro data was not affected too much with this filter, however small noise was removed.

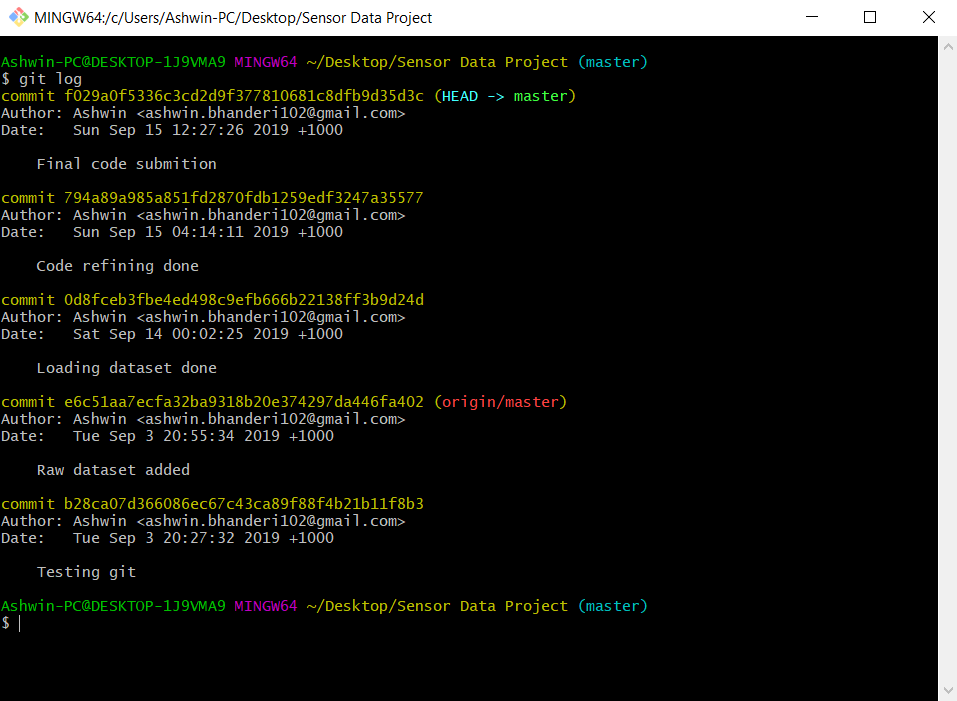
. 

The complex matrix accuracy depended on the number of neighbours near the diagonal line. Also, the standard scaler in the training example was replaced with MaxAbsScaler to increase the accuracy. The number of sample features in feature extraction method can also increase the accuracy by implementing other variable such as variance, standard deviation and median. However, in my program I was getting errors if I added more than 3 feature samples. Overall, I was able to increase the accuracy by a small margin due to errors by adding more sample features.

**Challenges:**

This project had a lot of challenges such as understanding the code and modifying it. Also running the code on a laptop computer was time consuming because of making small changes and testing how it affects the program. The sample code provided was smelling and written in bad programming practise. Some of the syntax errors was hard to fix because python is fairly a new language (4th generation) and there is not much support on the internet. Likewise, using Jupiter at the beginning was tricky because the program needed to be run in sequentially manner otherwise the output might defer widely. While refactoring the code for this task, some function could not be implemented because loops were used to move through each activity and dataset. Refining the code for feature extract function had two similar for loops, hence I created a function to call and put the loop inside but I was getting errors.

**Version Control:**



For this project, GitHub version control was used to keep track of the changes and iteration of the project. This enables us to see the history of the project and how it has been progressing. Moreover, it allows us to see how active the project is. Also, it can track small changes inside a file such, addition or deletion of code. This can also be beneficial if the project is carried out in a team environment to know who made the changes and why. Furthermore, this method can allow different versions of the code to be developed alongside and then merged to the main repository.

**Conclusion:**

Overall in this project, we were able recognise user activities using machine learning by loading and filtering the data to remove noise. The cleaned data was used to extract features such as min, max and median for KNN and SVC classifiers for training and testing the machine learning models. Also, complex matrix was used to get the accuracy for each activity by the model. The project life cycle was based on agile software development to ensure the completion of the project was on time. The sample code provided was smelling and had to be refactored. All the project files were being tracked by GitHub for version control.

**Reference:**

Link to GitHub repo:

<https://github.com/ASHHBOT/Sensor-Data-Project>