A REPORT ON INSIGHT GENERATION THROUGH EXPLORATORY DATA ANALYSIS (EDA) TECHNIQUES AND MACHINE LEARNING

BY

OYEDEJI, OLAYEMI DANIEL

STUDENT ID: ST20206551

MODULE CODE: CIS 7031

PROGRAM: M.Sc. DATA SCIENCE

COURSE TITLE: PROGRAMMING FOR DATA ANALYSIS

Table of Contents

[1.0 ABSTRACT 4](#_Toc89192644)

[1.2 METHODOLOGY USED 4](#_Toc89192645)

[1.3 IMPLICATION OF THE STUDY ON PHONE COMPANIES 4](#_Toc89192646)

[1.4 OUTCOME/FINDINGS FROM RESEARCH 4](#_Toc89192647)

[2.0 INTRODUCTION 5](#_Toc89192648)

[2.1 AIMS /OBJECTIVES OF THIS STUDY 6](#_Toc89192649)

[3.0 METHODOLOGY USED 7](#_Toc89192650)

[3.1 DATA COLLECTION 7](#_Toc89192651)

[3.1.1 Matlab Generation 7](#_Toc89192652)

[3.1.2 PROCESSING OF THE DATA 8](#_Toc89192653)

[3.1.3 Formulation of the problem 11](#_Toc89192654)

[3.1.4 Exploratory Data analysis 12](#_Toc89192655)

[3.1.5 PLOTS TO ESTABLISH RELATIONSHIPS 14](#_Toc89192656)

[4.0 ANALYSIS OF THE MACHINE LEARNING MODEL 20](#_Toc89192657)

[4.1.1 Selection of models and training 20](#_Toc89192658)

[4.1.2 Evaluation of models 21](#_Toc89192659)

[4.1.3 Procedures for model evaluation 21](#_Toc89192660)

[4.1.7 Hyperparameters tuning 22](#_Toc89192661)

[4.1.8 Final Predicted model 22](#_Toc89192662)

[4.2 CRITICAL REVIEWS OF TOOLS AND TECHNIQUES USED 22](#_Toc89192663)

[4.2.1 Activation Function 22](#_Toc89192664)

[4.2.2 Metrics 22](#_Toc89192665)

[4.2.3 Loss Function 23](#_Toc89192666)

[4.2.4 Optimizers 23](#_Toc89192667)

[4.2.5 Kernel Initializer 23](#_Toc89192668)

[5.0 Analysis and Interpretation of Results 24](#_Toc89192669)

[5.1.1 Split data into training and testing set 24](#_Toc89192670)

[5.1.2 Scaling the data 25](#_Toc89192671)

[5.1.3 Implementing the machine learning models 25](#_Toc89192672)

[5.1.4 Model result comparison 27](#_Toc89192673)

[5.1.5 Hyperparameters tuning 27](#_Toc89192674)

[5.1.6 Evaluating the models beyond accuracy 28](#_Toc89192675)

[5.1.7 Classification Report 29](#_Toc89192676)

[6.0 Artificial Neural Network 30](#_Toc89192677)

[6.1 Encode categorical data 30](#_Toc89192678)

[6.1.1 Selection of X and y values 31](#_Toc89192679)

[6.1.2 Splitting data set into training and test set 31](#_Toc89192680)

[6.1.3 Encoding the classes for output data 32](#_Toc89192681)

[6.2 Create a Neural Network 32](#_Toc89192682)

[6.2.1 Determine the size of the units 32](#_Toc89192683)

[6.2.2 Neural Network Build 32](#_Toc89192684)

[6.3 Compiling the Neural network 33](#_Toc89192685)

[6.3.3 Neural network evaluation 34](#_Toc89192686)

[6.4 Test Model on Unseen dataset 36](#_Toc89192687)

[6.4.1 Evaluation of result on Unseen Test value 37](#_Toc89192688)

[7.0 EXPERIENCE DURING THE REPORT 37](#_Toc89192689)

[REFERENCES 38](#_Toc89192690)

# 

# 1.0 ABSTRACT

The purpose of this report is to help distinguish actions being performed by users of the sensory-enabled smartphone. Users can be performing different actions at some given time-varying from walking, jogging, and running to sleeping. Data generated from these actions can be manipulated and analyzed to predict the exact action being carried out at a point in time.

## 1.2 METHODOLOGY USED

MathWorks was used to collect and clean the data collected from the actions carried out. Deep Learning and machine learning algorithms were used for the analysis and visualizations with the help of python.

## 1.3 IMPLICATION OF THE STUDY ON PHONE COMPANIES

Phone companies can deduct from the analysis of the study which actions users are carrying out at a specific time which would help the companies to innovate and create improved versions of the phones based on their actions. Flaws with the designs of the phones can also be deduced from the report.

## 1.4 OUTCOME/FINDINGS FROM RESEARCH

Human classification problems can be solved by the use of machine learning and neural networks. The activity carried out by an individual at a particular time can be predicted by sensors on a mobile phone.

1.5 LIMITATIONS ON RESEARCH

The number of activities performed in the report is limited to three which include; walking, jumping, and swinging. This would bring about a gap when the model cannot interpret other activities introduced by the data.

# 2.0 INTRODUCTION

Big data has become the order of the day in the world right now. IoT has become a word that’s very popular in household settings. With the emergence of Artificial Intelligence, data plays a huge role in helping it works. The effectiveness of artificial intelligence is highly dependent on the enormous amount of data collected. Machine learning with the help of artificial intelligence helps use data to predict and provide the best solutions; and in terms of this report, it helps with human action classification.

Data is now being collected by different devices across different industries as explained in IoT. Technology such as GPS trackers, RFID chips, and security cameras makes use of big data to obtain information about the positions of items and usage of appliances across homes. Digital communication has made the relationship between these devices very easy and manageable, to say the least.

WHY SENSORY DATA COLLECTION THROUGH MOBILE DEVICES

The data recorded and collected in this report was generated through MatLab by using sensors in phones. The data was collected while performing different activities. The data is simply collected to aid with the human activity classification problem. Through machine learning algorithms, we can predict which activity is being carried out by a person at a particular time.

MACHINE LEARNING/EXPLORATORY DATA ANALYSIS

Testing and training data sets to predict actions can be done by machine learning algorithms and neural networks. Machine learning is under the umbrella of artificial intelligence which makes a system develop and identify patterns in data and make informed decisions with little or no human intervention.

Exploratory data analysis is used to present data in graphs and other plot types to create relationships and understand the trends being faced by the data. This analysis can help evaluate the products of a company and how it fares in the market against other products.

## 2.1 AIMS /OBJECTIVES OF THIS STUDY

The primary goal of this report is to generate insight from data generated from Matlab through sensors, develop models, create relationships and predict the best algorithms/learning models to imbibe.

Objectives of the study:

- Generate and analyze the created data

- Exploratory data analysis of the data and interpretation

- Develop a machine learning predictive model

- Develop a Neural network predictive model

- Critical evaluation of the models used.

# 3.0 METHODOLOGY USED

## 3.1 DATA COLLECTION

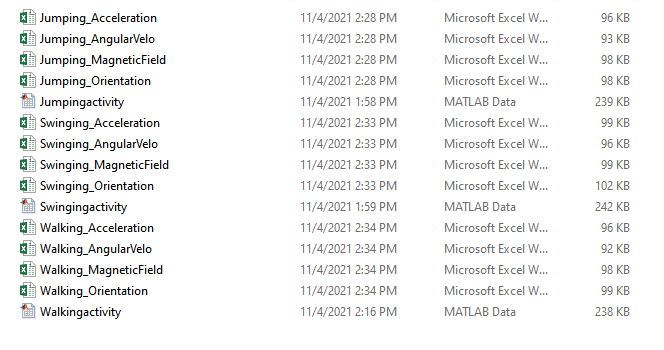
### 3.1.1 Matlab Generation

The data used in this report was generated with the use of Matlab mobile. Navigating the application and putting on the sensors (Orientation, Acceleration, Angular velocity, Magnetic field); and setting the sample rate to 20 Hz to perform three different activities. The activities selected for this report include walking, jumping, and swinging.

The three activities were carried out with MatLab and the outcome consisted of records of the sensors across the X, Y, and Z-axis for each activity. Each sensor activity had its report saved in four files. The Reports were originally generated as MAT files and were converted to .xlsx files with the MatLab application.

Below is a screenshot of the generated data:

FIGURE 1. GENERATED DATA



### 3.1.2 PROCESSING OF THE DATA

Processing the data is a very important step in the general fine-tuning of the data to get it prepared for any machine learning module or neural network.

-Exploration of data

This step involves the loading of the data into the python IDE used as is recommended by the report. Our reports are in .xlsx format as stated earlier so the pd.read\_excel ( ) function is used to load the various records generated by the activities performed. Another important aspect of the exploration is knowing the structure of the data generated which is performed by using the .info () function. This function is used to check the data type of the records. The .isnull ( ).sum ( ) function is used to get a sum of null values present in every column. The .head ( ) function is to show the data in a preview from the top



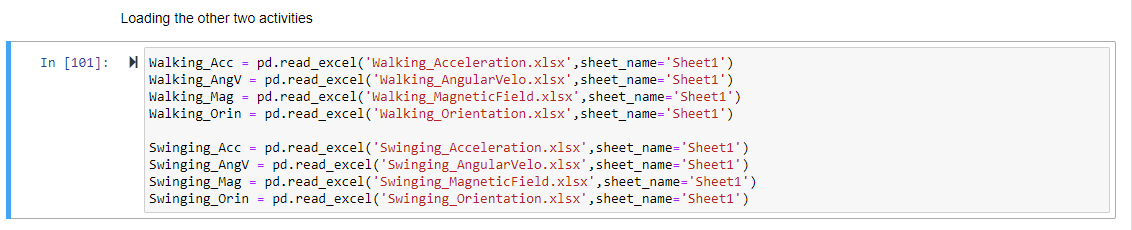


FIGURE 2. LOADING GENERATED DATA

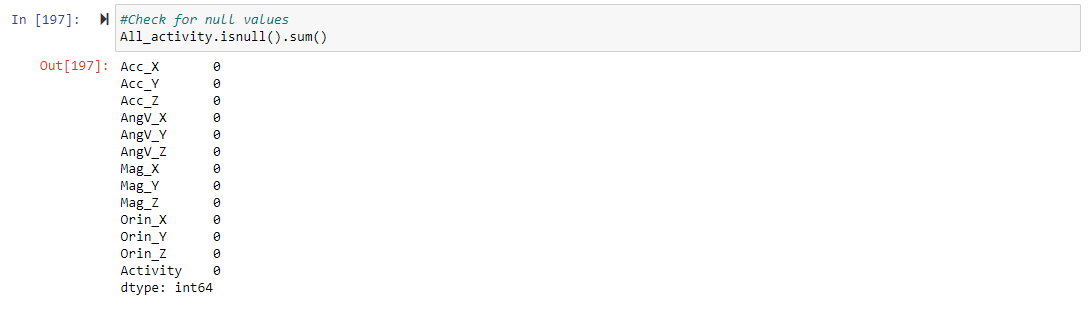


FIGURE 3. CHECKING FOR NULL VALUES

In my generated data, after combining all activities into a single CSV. The number of null values is zero.

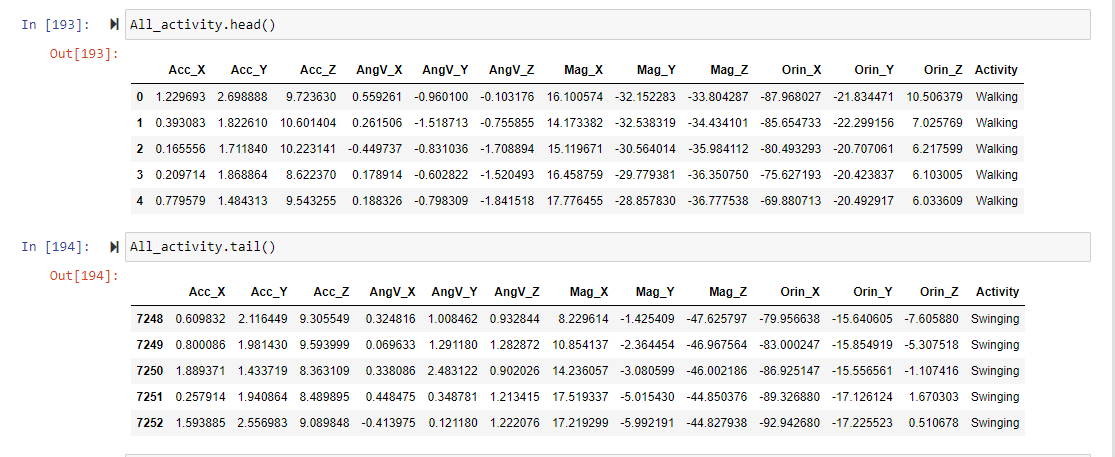


FIGURE 4. VIEWING THE DATA GENERATED

DATA CLEANING

If there are any null values detected in the datasets, we would need to drop the row with null values. Missing or null values can also be manipulated by putting the median, mean, or the value before it in place of it.

We do not have any missing values in my dataset so this step is not to be taken on my dataset.

SELECTION OF FEATURES FOR MACHINE LEARNING

This involves the selection of columns that would influence the final output of an independent variable. In this dataset, all the columns are used apart from Timestamp. All the columns influence the output apart from the Timestamp column. None of the data in the columns from the feature column can be missing or there would be a prediction error.

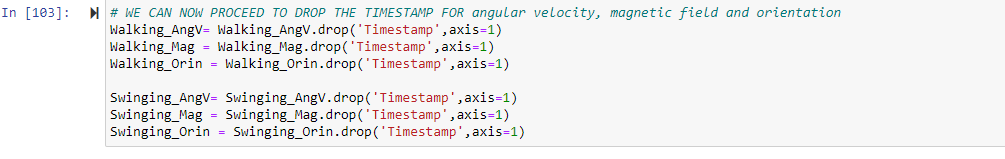


FIGURE 5. DROPPING TIMESTAMP

COMBINATION OF ALL FOUR FRAMES

All four different activities were combined to create a single dataset to make analysis easier and the exploration of data easier to carry out. Pandas create and give avenues for data frames to be combined.

The function .Concat ( ) is used to combine data frames.

For this report, the combination of the four frames was done individually before combining all activities.

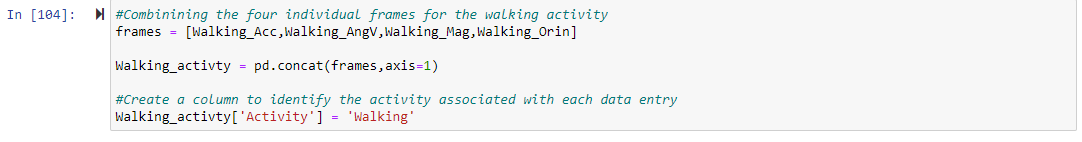
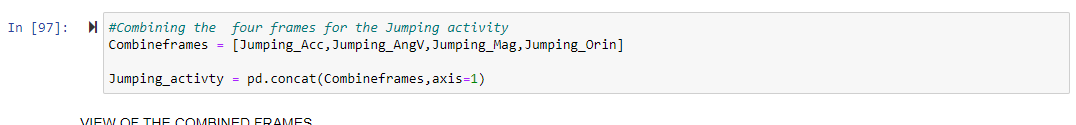
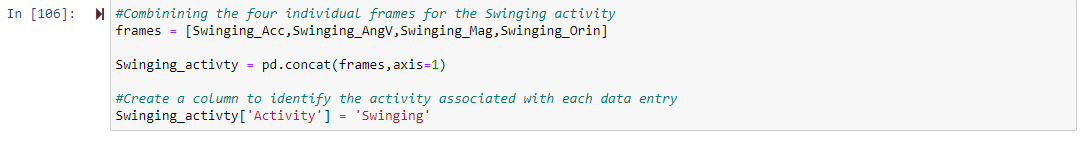


FIGURE 6. COMBINING THE FOUR FRAMES OF THE THREE ACTIVITIES

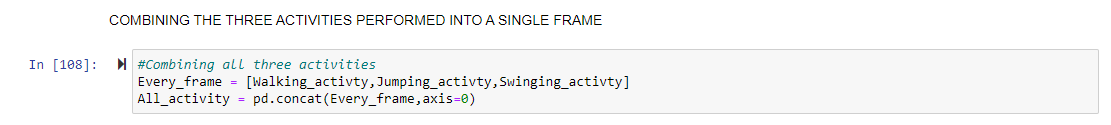


FIGURE 7. COMBINING ALL THE THREE ACTIVITIES INTO A SINGLE FRAME

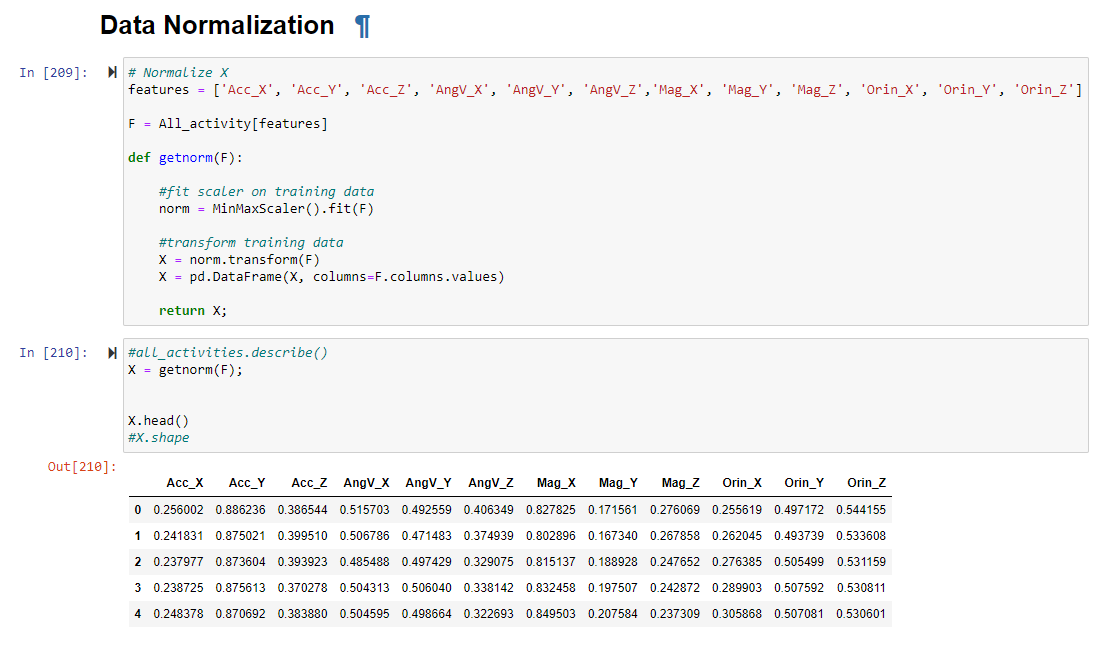
NORMALIZATION OF DATA

This involves the conversion of all entry data from data sets to a standard scale. In machine learning, in default, the features with higher/greater values would dominantly influence the learning process of the models deployed.

Reason for normalizing data:

* Some feature values differ in large scales from others
* Larger features would influence predictions

The function get norm ( ) is used to normalize datasets.



### 3.1.3 Formulation of the problem

In other to create and evaluate a proper model the machine learning, it is important to have a problem definition. In this report, our factors and variables are very important in defining the problem.

Goals of this report

* Make predictions to determine which activity is being carried out by a user at a particular time
* Investigate which parameters are the best loans for the prediction model
* The Y variable (Dependent Variable) in this report are the activities carried out Jumping, swinging, walking.

### 3.1.4 Exploratory Data analysis

Exploratory data analysis in context refers to all investigations and critical thinking process that is performed on data at different levels of data processing (Patil, 2018). These investigations on data are to rediscover and create patterns, discover anomalies, and checkmate assumptions. Some other importance is to test hypotheses and majorly summary statistics and graphical representations.

Steps to having Good Exploratory Data Analysis:

* Understand the data
* Gather insights from the data

In this report, I made use of python libraries to achieve the E.D.A. I imported necessary libraries like pandas, NumPy, matplotlib, and seaborn.

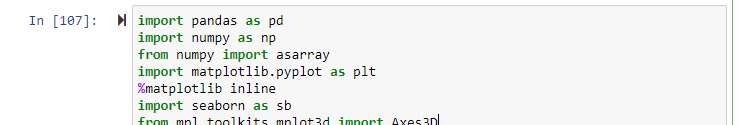


FIGURE 9. IMPORTING EDA LIBRARIES

-Checking for null values and shape

Making sure the data has the right shape and is in the right format to be explored is the first step to having a successful analysis

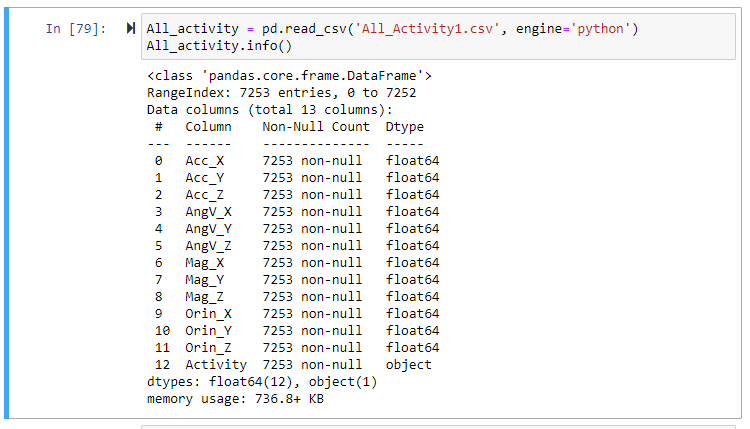
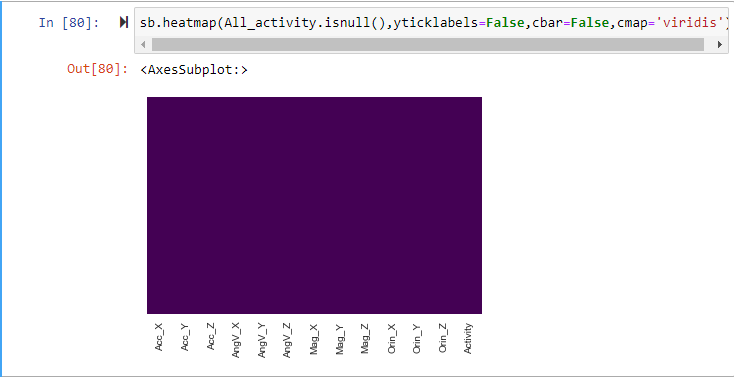


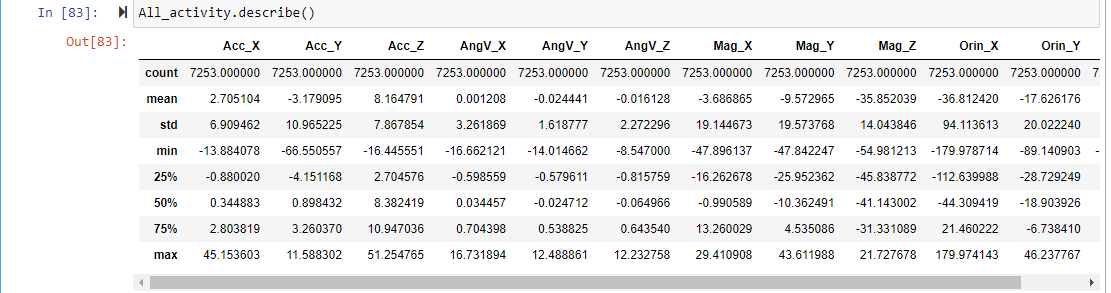
FIGURE 10.CHECKING THE SHAPE OF THE DATASETS

* Visualizing a heat map to check for null values



* Description of the data set

This shows various values of the data sets which include; Count, mean, std, min, 25th percentile, 75th percentile, and the maximum values

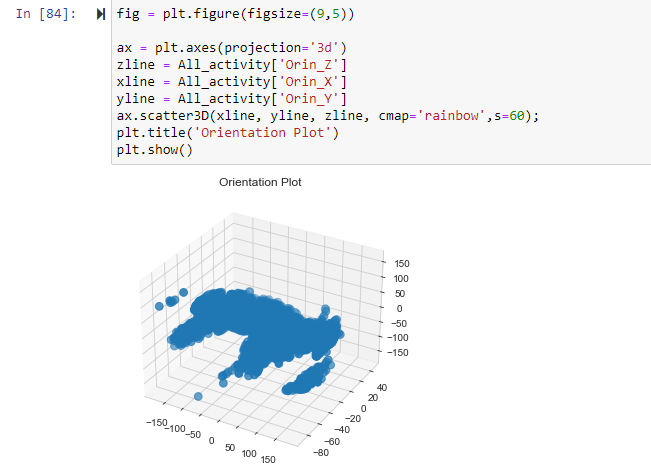


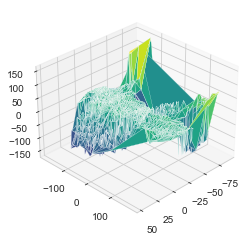
### 3.1.5 PLOTS TO ESTABLISH RELATIONSHIPS

This report is on human activity classification and as well as 3D Data. Figure plots were used to create relationships between sensors during different activities.

ORIENTATION PLOT

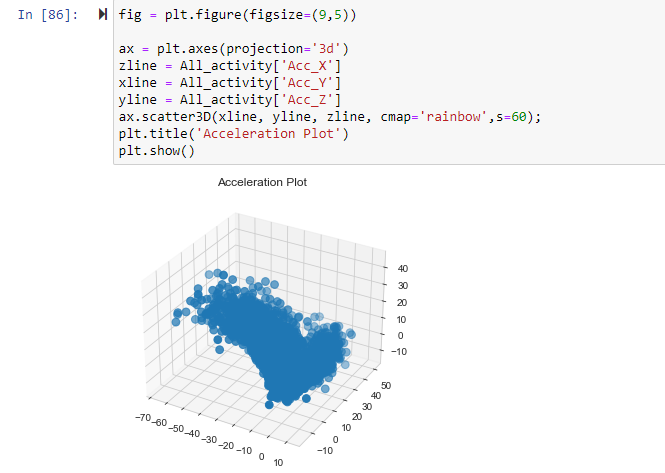
This plot shows the X, Y, Z-axis of the orientation sensor during all the activities and their correlation





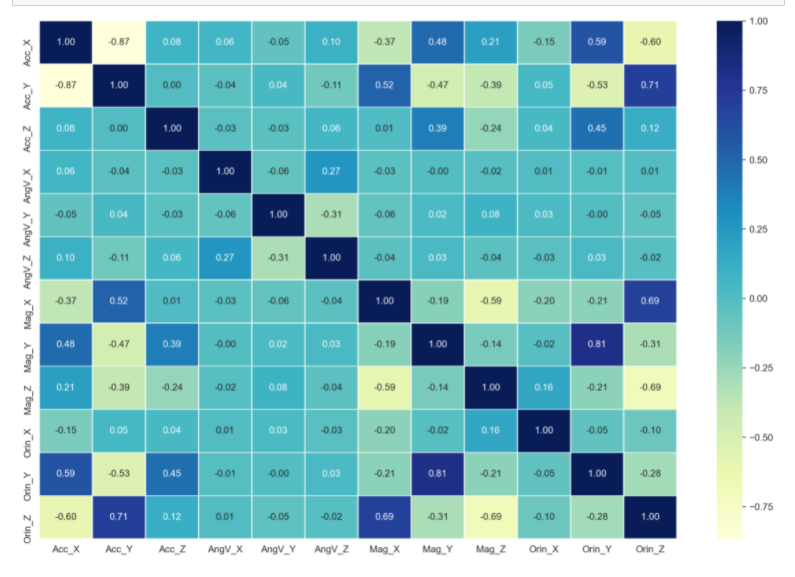
Acceleration plot

This plot shows the X, Y, Z-axis of the acceleration sensor during all the activities and their correlation



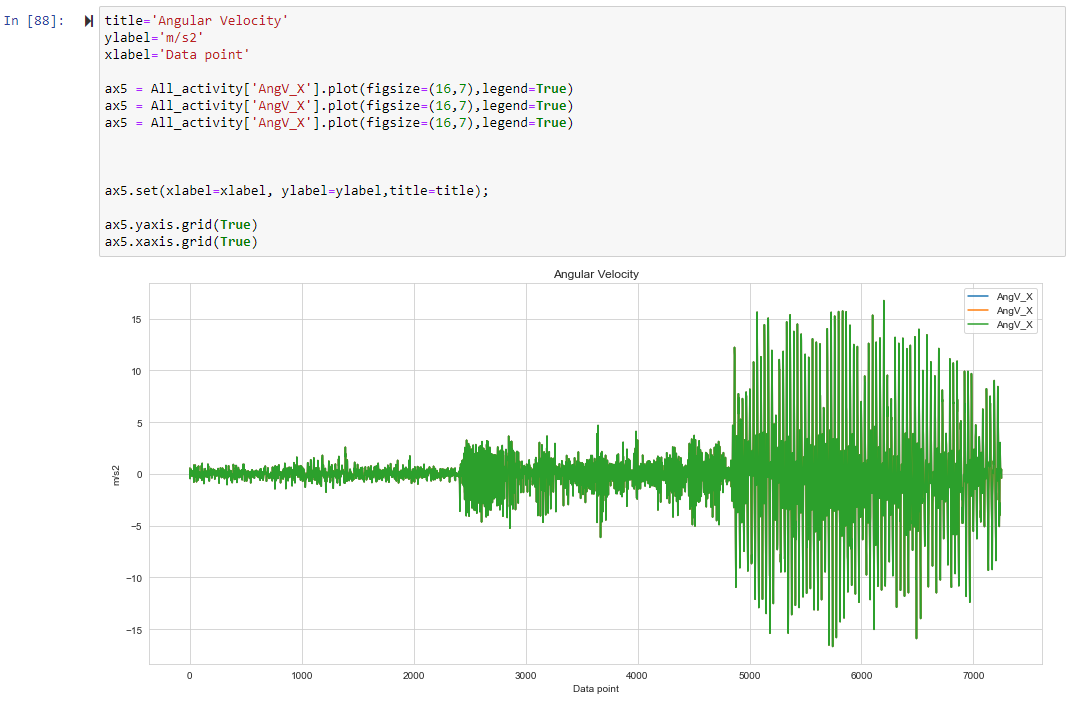
INTERPRETATION: We can infer that the Acc\_x, Acc\_y, and Acc\_Z are based on which activity is being carried out. From this graph, we can infer that the activity being carried out far left from data point 0 to about 2300 is WALKING, from data point 2300 to 4800 is JUMPING and the last activity Swinging is from data point 4800 to 7523. With these, we can tell the difference in acceleration values with all activities in place.

CORRELATION MATRIX



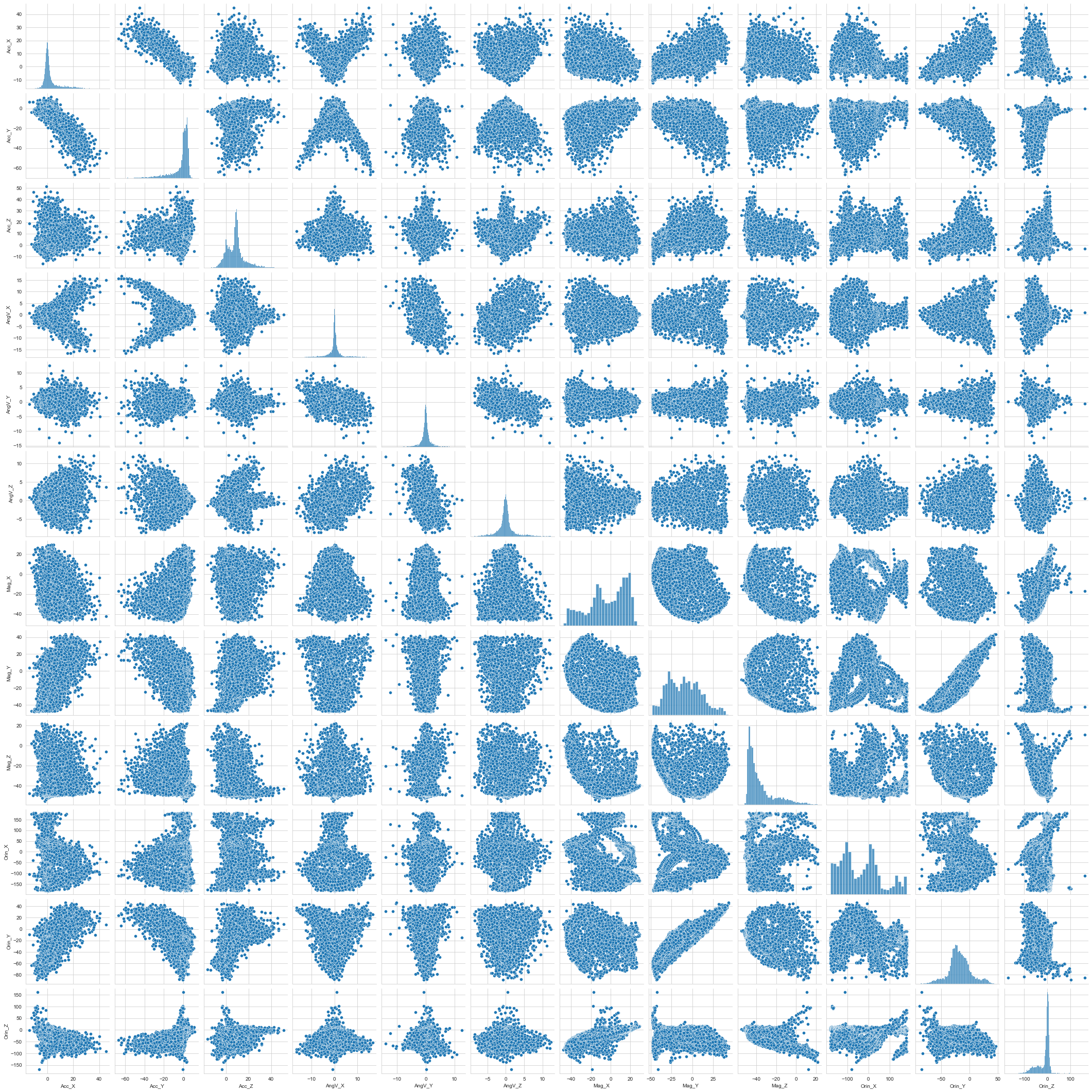
The correlation matrix shows the relationship between columns, the stronger the color, the greater the correlation magnitude of the columns.

ANGULAR VELOCITY PLOT



INTERPRETATION: We can infer that Angv\_x, Angv\_y, and Angv\_Z are based on which activity is being carried out. From this graph, we can infer that the activity being carried out far left from data point 0 to about 2300 is WALKING, from data point 2300 to 4800 is JUMPING and the last activity SWINGING is from data point 4800 to 7523. With these, we can tell the difference in Angular velocity values with all activities in place.

PAIR PLOT



A pair plot distribution allows us to see the distribution of single variables and relationships between other variables (Koehrsen, 2018). Pair plots are majorly used to identify trends in datasets.

# 4.0 ANALYSIS OF THE MACHINE LEARNING MODEL

## 4.1.1 Selection of models and training

The selection of a machine learning model is highly dependent on the given problem. Our report is a human classification which makes the problem a classification problem. We want to be able to predict which exact action is being carried out by an individual at a given point in time. We must understand the quality of the data we are provided to select the best analysis model.

Some necessary checks before selecting a machine learning model:

* Size /Dimension of the data collected: Machine learning models are always much more successful when the size of the data collected is large and can easily be characterized.
* Understanding the parameters used in the model: A lot of entries go into enacting machine learning models. We must understand the meaning of the parameters used in the models. We must know when to tweak certain parameters to get the best results.
* The quality of the dataset: If the collection of data was not done in the best way, there is likely to be difficulty in implementing the right model.
* Selection of the right features: Understanding the concept of features when creating the model is key. Selecting the wrong features would affect the predictability of any model.

In this report, the problem is a classification problem so I made use of classification models; Logistics Regression, K-nearest neighbor a, and random forest classifiers. I also used an artificial neural network to run predictions.

We cannot use linear regression models because they would return continuous values.

Why Logistics Regression?

The logistics regression model is a mathematical technique from the field of statistics to help in machine learning (Raj, 2020). This model was developed in such a way to help prediction from one or more independent variables. The model helps to find the best fitting model to describe the relationship between independent and dependent variables.

Why KNN?

K-Neighbors model is a supervised machine learning algorithm. This model can solve both classification and regression problems (Harrison, 2018). This model depends on the data it is fed to make decisions and predictions. KNN has a discrete value for its output; so is the solution to the prediction model we are creating.

Why Random Forest?

Random forest is a classification algorithm that is coined from a decision tree algorithm. This model is used to solve classification problems. The random forest defeats the limitations of a decision tree algorithm because it is more advanced. Increasing the number of decisive trees increases the validation and precision of the outcome.

## 4.1.2 Evaluation of models

This involves the choice made between models, parameters tuning, and features tuning. It is a procedure where an evaluation metric is used to measure and qualify model performance. It is how well a model can generalize and predict the right results.

We also explored the fitting of the model to see if it’s overfitting or Underfitting:

-Overfitting means Validation loss > training loss

-Underfitting means Validation loss<training loss

-Okay is when Training loss=Validation

### 4.1.3 Procedures for model evaluation

-Splitting data into Train and testing sets

This would help the algorithm to not generalize predictions that are assumed.

-K-fold Cross-validation

Create ‘k’s train/test splits and get an average of the results and estimate a better out-of-sample performance (Ritchie, 2021).

### 4.1.7 Hyperparameters tuning

This involves the selection of the best values of parameters to give an optimal model structure. Parameters involved in machine learning such as activation units, rate, epochs, hidden layers, and hidden units are best utilized to give an optimal model structure.

### 4.1.8 Final Predicted model

This is the final model selected to be used after optimal parameters have been used to have the best results.

## 4.2 CRITICAL REVIEWS OF TOOLS AND TECHNIQUES USED

### 4.2.1 Activation Function

An activation function is used to introduce additional steps at each layer in machine learning. Without activation functions, neutrons would only perform linear transformations on the input layers using weights and biases. There are various types of activation functions. Some are Binary step, linear, sigmoid, tanh, ReLU, Leaky ReLU, Parameterized ReLU, Swish, and softmax. In this report which is a classifier problem, sigmoid functions and their combinations are the best fit. ReLU functions are general activation functions that can work for most problem types. The activation function is a major building block for neural networks.

### 4.2.2 Metrics

Classification problems are about predicting class labels from input data. In a multiclass classification, more than two outcomes can be achieved. There are different ways to measure the performance of a classification which include accuracy, log-loss, and AUC-ROC. The accuracy metrics were used for our classification report. Accuracy measures how often the classification model predicts the right value. It is the ratio of the correct prediction to the total number of predictions. Accuracy is mostly useful when the target class in a dataset is well and thoroughly balanced. Some other metrics can be used in place to get for unbalanced datasets.

### 4.2.3 Loss Function

The loss function of a model is used to majorly measure if an algorithm is doing a good job (Pere, 2020). The loss function is used to measure the algorithm’s current predicted output to its expected output. The loss function is mainly used as a feedback mechanism for the algorithm, it helps with the learning of the model. It is used to evaluate how well an algorithm models the data. The configuration of output layers must be in alliance with the chosen loss function. The loss function chosen is also dependent on the modeling problem to be solved; either classification or regression.

This report makes use of the categorical loss cross-entropy function, this function is one of the most suitable functions for a multi-class classification problem. The outer activation layer of the neural network works great with this loss function. The disadvantage of using categorical entropy is that the variables it is fed need to be encoded, and this would affect larger datasets.

### 4.2.4 Optimizers

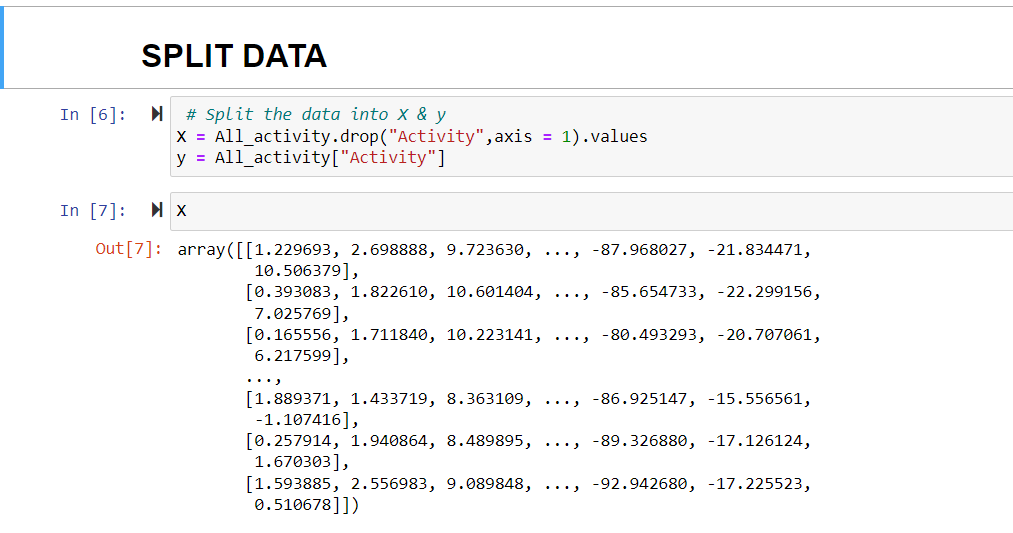
Optimizers are algorithms used to train models to reduce their error rate (Doshi, 2019). They are used to edit and manipulate attributes such as learning rate and weight to reduce loss in a model. The change and editing of attributes depending on the optimizer being used in the model. This is because optimizers are responsible for the reduction in loss and the provision of more accurate model predictions. There are various examples of optimizers namely Gradient Descent, stochastic gradient descent, mini-batch gradient descent, momentum, Nesterov accelerated gradient, adagrad, Adam, and adadelta. In this report, stochastic gradient descent (SGD) was used as an optimizer that frequently updates model parameters and requires less memory to store loss functions. Some drawbacks include a high variance in model parameters and getting a convergence can only be by slowly reducing the value of the learning rate.

### 4.2.5 Kernel Initializer

The kernel initializer is used to define the weights of a Keras layer. It is generated by statistical distribution and functions. Kernel initializers include Xavier and he\_uniform.

# 5.0 Analysis and Interpretation of Results

The machine learning implementation started with the splitting of data into training and testing sets. The X value is all the values on the datasets under each column apart from the timestamp column. The y value is the set of values that is to be predicted as “the activity”. The y value is the column “activity”.

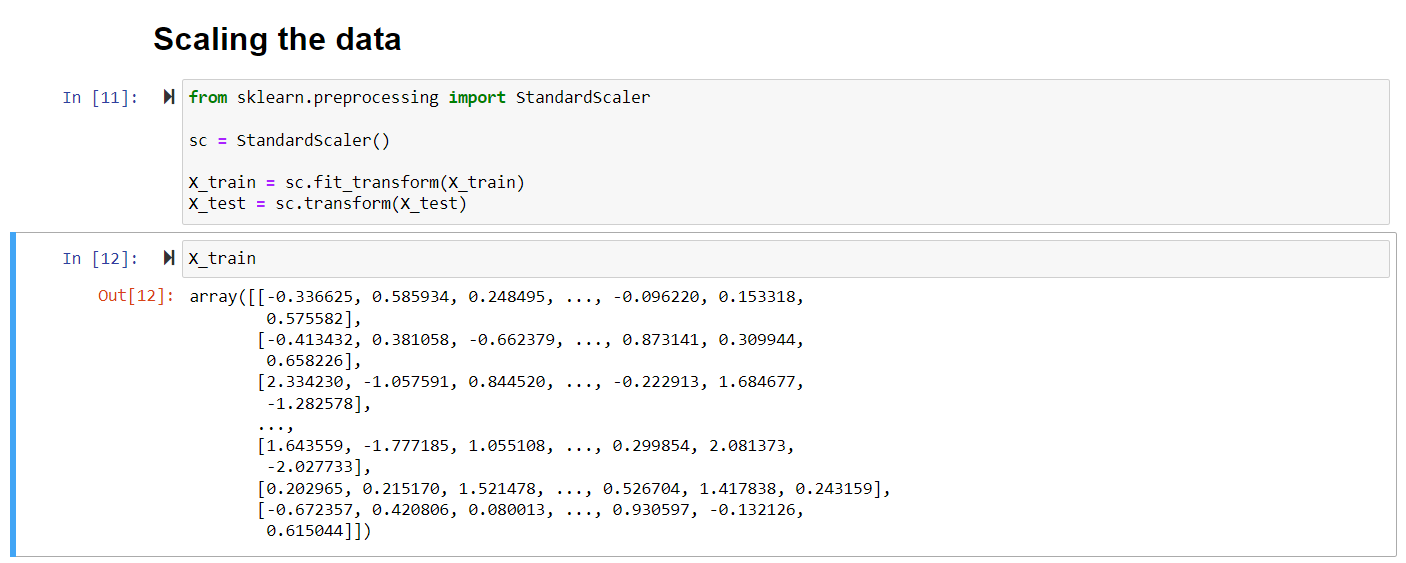


## 5.1.1 Split data into training and testing set

Dividing data into training and testing sets is a proper standard in data models. It is important to split data into train and test sets with the training set as the larger set and the testing set as the lesser set. Analysis services help to keep the data in check and make sure the samples of data for both training and testing set are similar. After a model has been processed with training sets, it would be tested with the test data set. The data in the testing set is made to be similar to the training set so it is easy for the model to predict and determine if the model is effective.

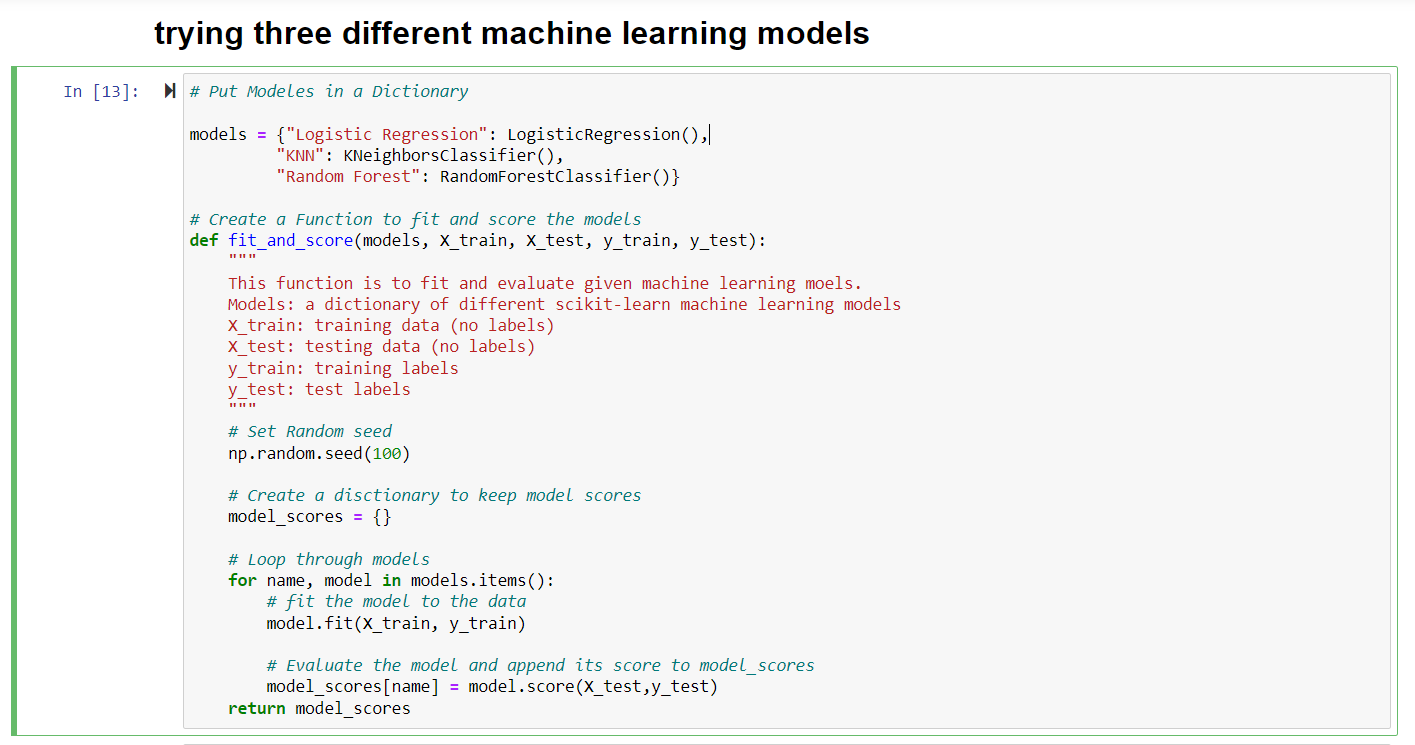
### 5.1.2 Scaling the data

Data scaling is very useful to the general shaping of our datasets. Input variables may have variable scales from different units. A good rule to imbibe would be to standardize input values with a zero mean and a standard deviation of one (Brownlee, 2020).

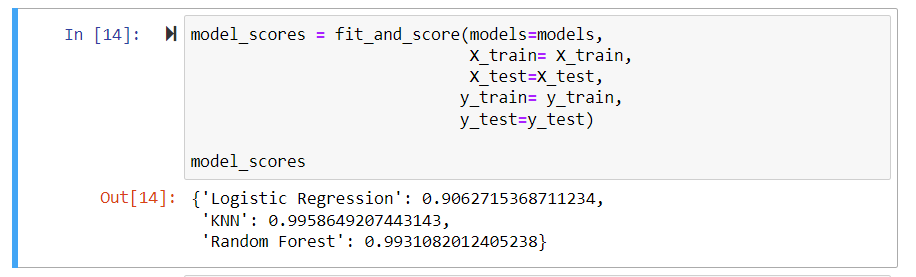


### 5.1.3 Implementing the machine learning models

In this report, three machine learning algorithms were implemented namely KNN, Logistic Regression, and Random Forest classifier models. The models were called in the same dictionary for implementation. A function fit\_and\_score was used to take in the parameters used to implement the models. The random seed function was set at 100, which means the initialization of random number generation was set at 100. The model scores were stored in a dictionary model\_scores.



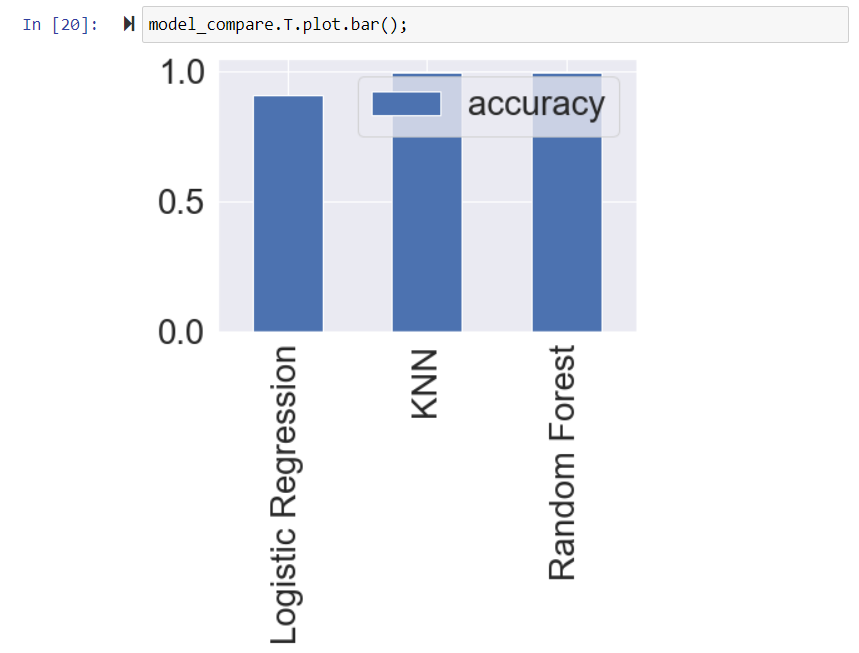
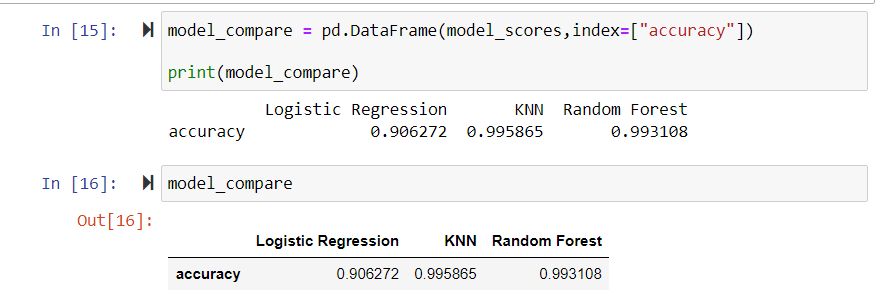
Result and analysis



From this output, Logistic regression has an accuracy of 91, KNN has an accuracy of 99 and Random forest has an accuracy of 99 percent.

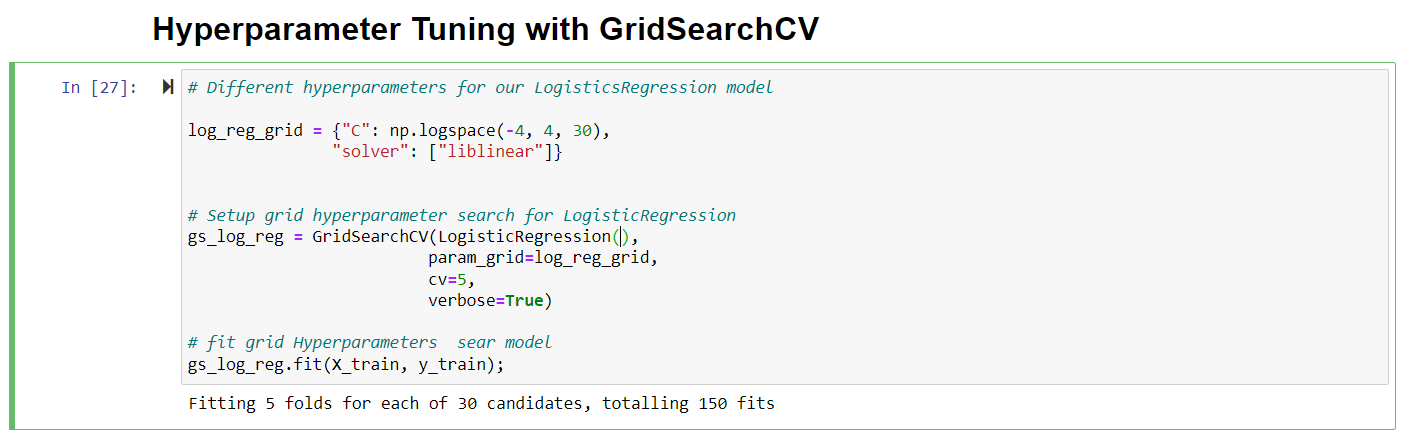
### 5.1.4 Model result comparison

The results from the models deployed above need to be compared, to know which of the models best fit.

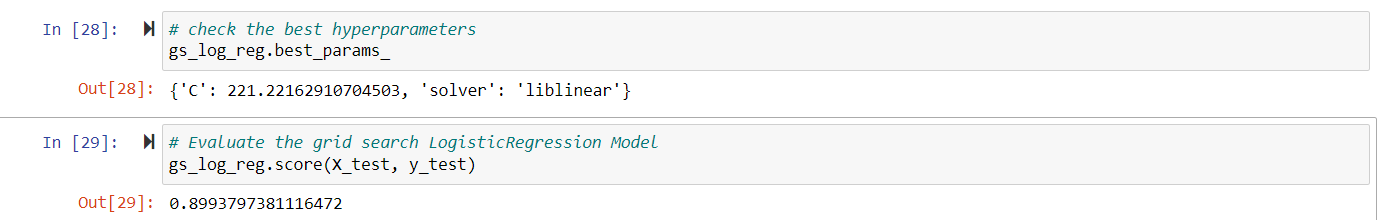


### 5.1.5 Hyperparameters tuning

As earlier explained in this report, Hyperparameters tuning is used to determine the best parameters to be used in other to get the best accuracy for the models. We would use gridsearchcv.



Interpretation: The tuning fits 5 folds to 30 candidates which produce about 150 fits.

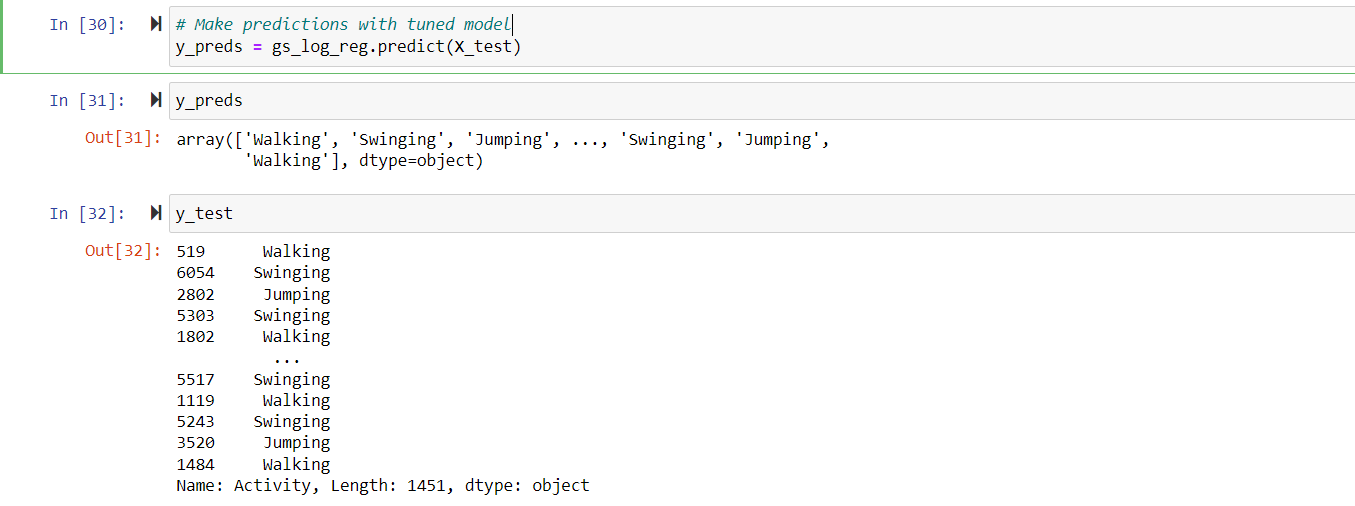


The best hyperparameter value is to be determined and the C statistic value which is equal to the area under the ROC curve is 221.2216. It is a measure of fit for the logistic regression model.

The hyperparameter value was evaluated and produced an accuracy of 89.9 percent.

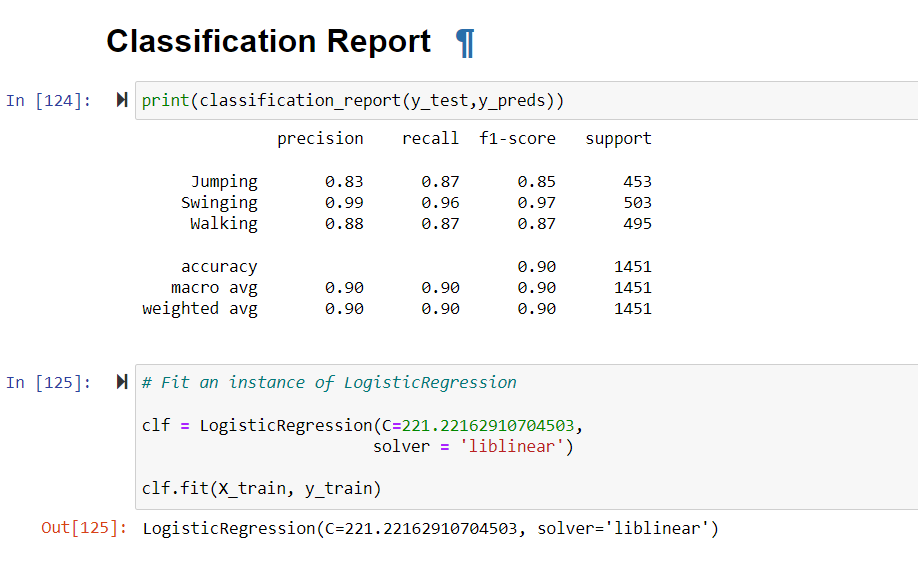
## 5.1.6 Evaluating the models beyond accuracy

This is the use of the recently generated tuned model to make predictions on the model.



A series of predictions is gotten from the tuned model.

### 5.1.7 Classification Report



Interpretation: The precision value for the activity jumping is 83 percent, the recall value which means the actual prediction correction is 87 percent, and the balanced score (f1-score) is 85 percent for jumping.

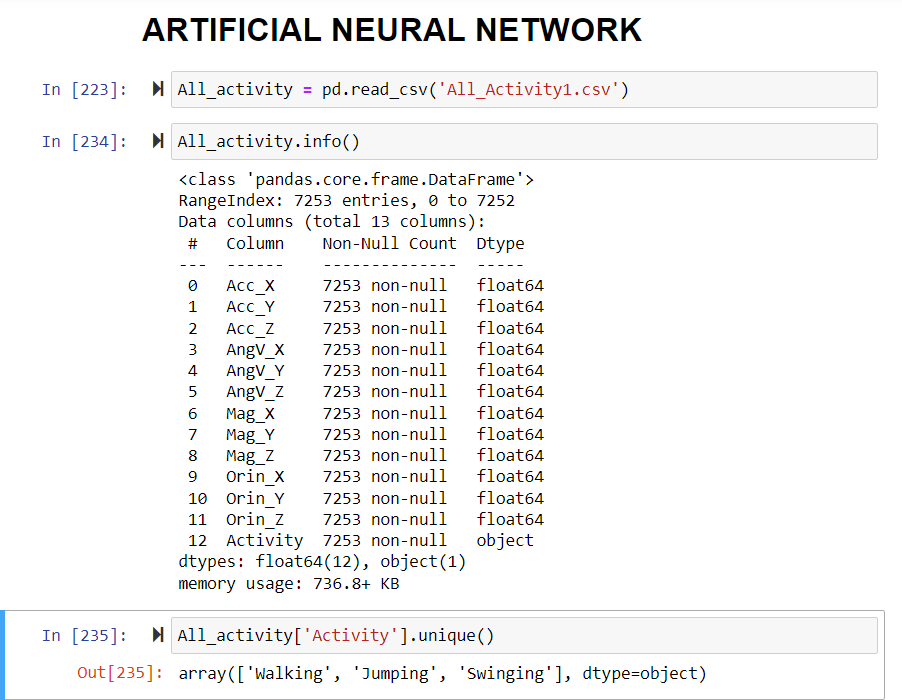
The precision value for the activity swinging is 99 percent, the recall value which means the actual prediction correction is 96 percent and the balanced score (f1-score) is 97 percent for swinging.

The precision value for the activity walking is 88 percent, the recall value which means the actual prediction correction is 87 percent and the balanced score (f1-score) is 87 percent for walking.

The average value of the scores for all activities is 90 percent.

# 6.0 Artificial Neural Network

The data saved in a CSV file is loaded into the notebook.

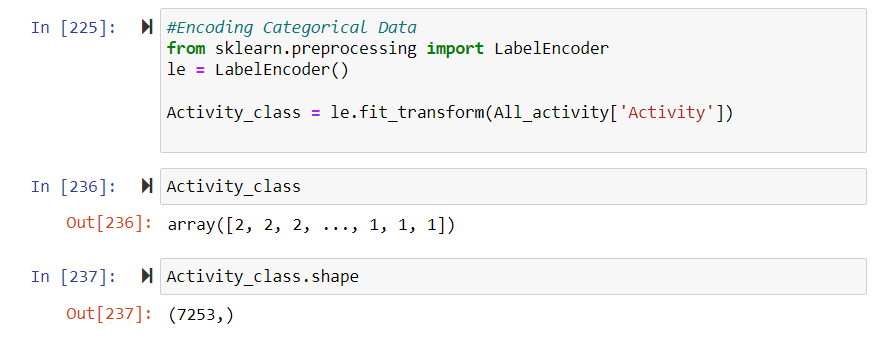


The structure of the data set is checked to avoid null values.

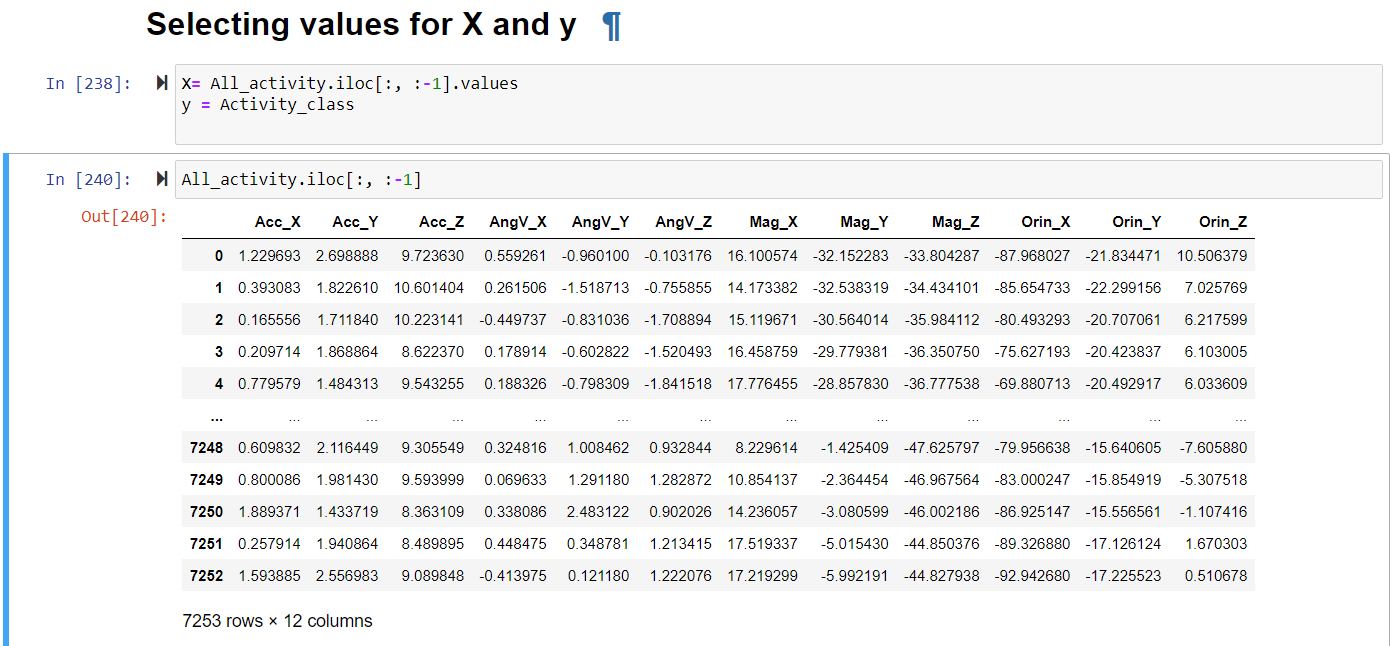
## 6.1 Encode categorical data

Autoencoders are used to efficiently compress and encode data and also learn how to reconstruct data to its encoded representation and a representation close to its original input (Badr, 2019).

Autoencoders are majorly used to reduce data dimensions by getting rid of the noise in the data.



### 6.1.1 Selection of X and y values



The X values are all activities while the y value is the activity column to be predicted.

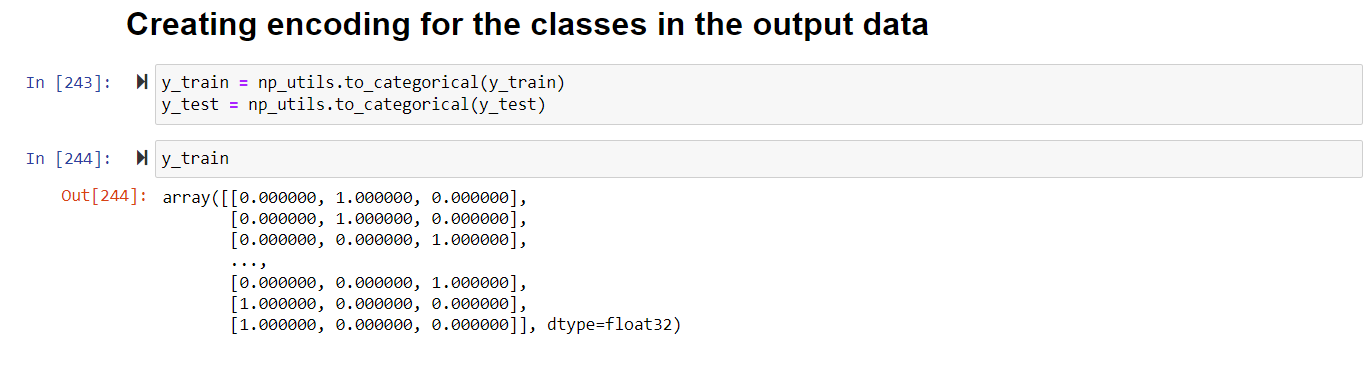
### 6.1.2 Splitting data set into training and test set

The data is split into training and testing sets and a standard scaler is used to scale the data to size.



### 6.1.3 Encoding the classes for output data

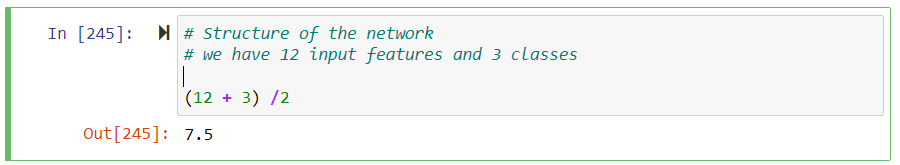
The training sets and testing sets are both encoded to achieve uniform output from the models.



## 6.2 Create a Neural Network

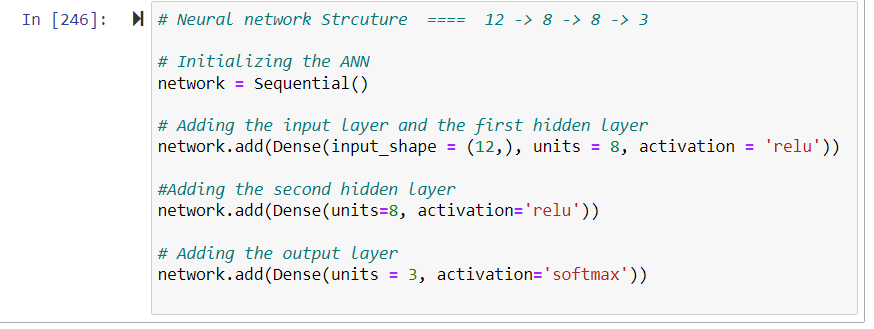
### 6.2.1 Determine the size of the units

The size of the units in the neural network is determined by the sum of the number of input features and the number of classes divided by 2. There are 12 input features and 3 classes of output.



### 6.2.2 Neural Network Build

The neural network structure consists of an input layer, first hidden layer, second hidden layer, and an output layer. The activation parameter used for the neural network in the inner layer is relu and the output layer uses softmax.



## 6.3 Compiling the Neural network

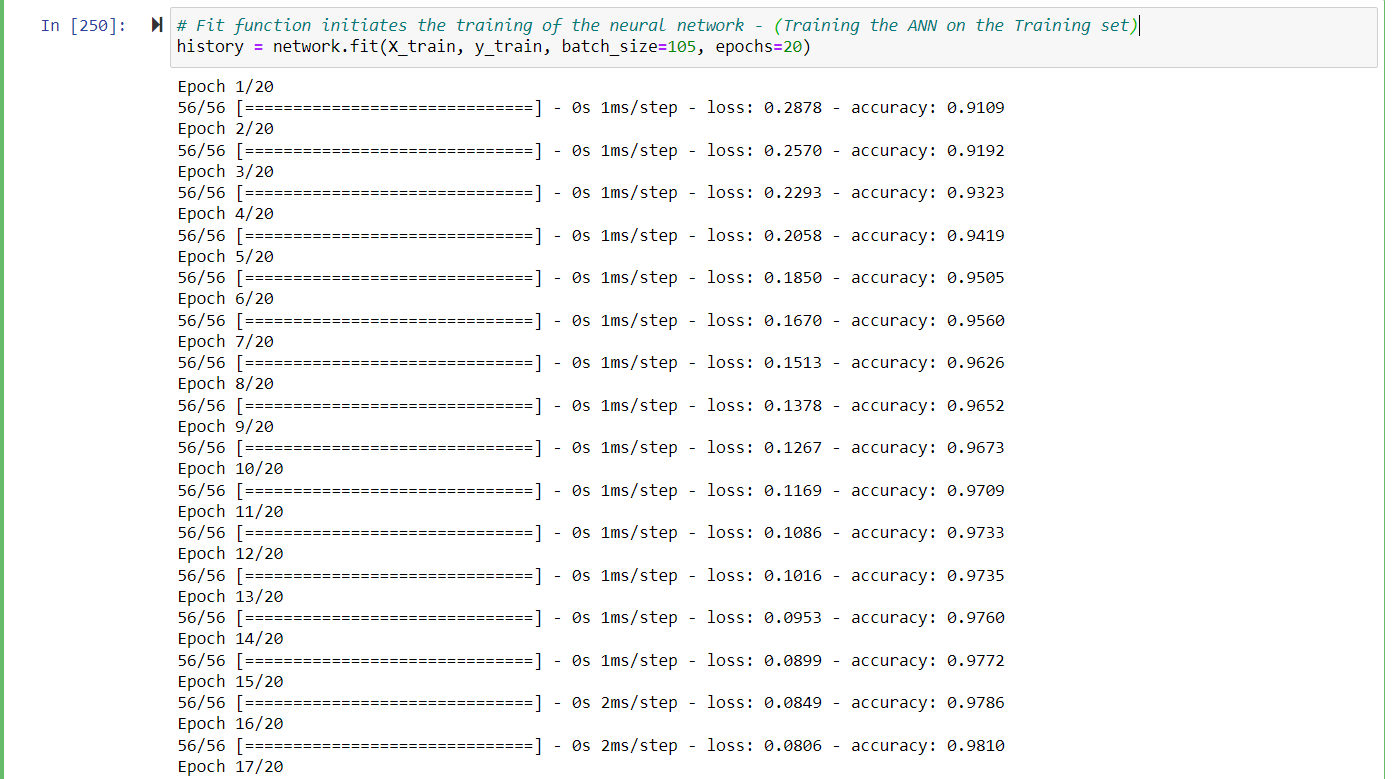
The loss function used is cross-entropy categorical loss and Adam is used as the optimizer and the metric used is accuracy.



The network summary shows that all the parameters imputed into the network were trainable.

6.3.2 Training the dataset

History is used to fit the data into the network for training. Some parameters are important while fitting data into a neural network for training. The batch size is set to 105 which means the number of data that goes for training at a point in time is 105. The epoch is set to 20 which indicates the number of cycles the neural network would perform training on the data set.

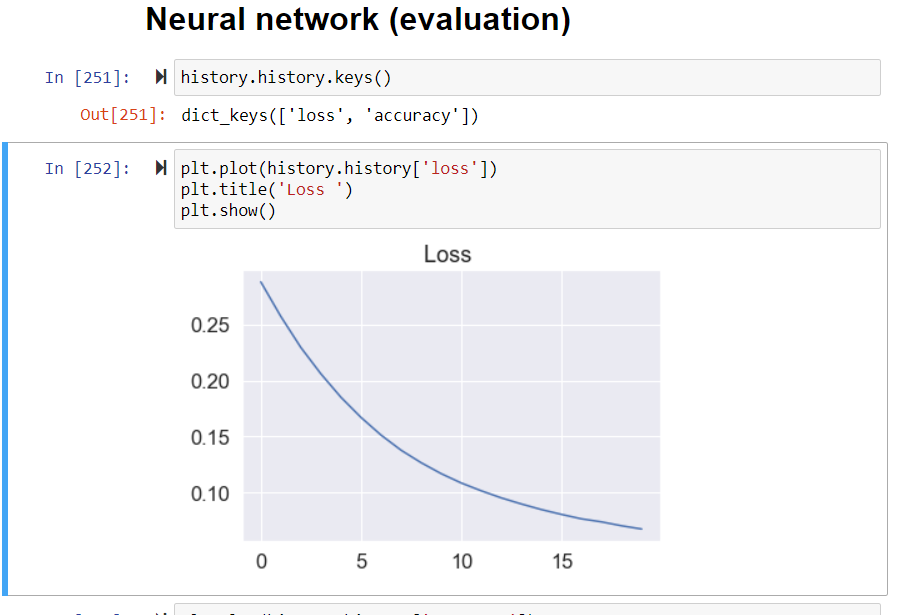


The data was trained for 20 cycles.

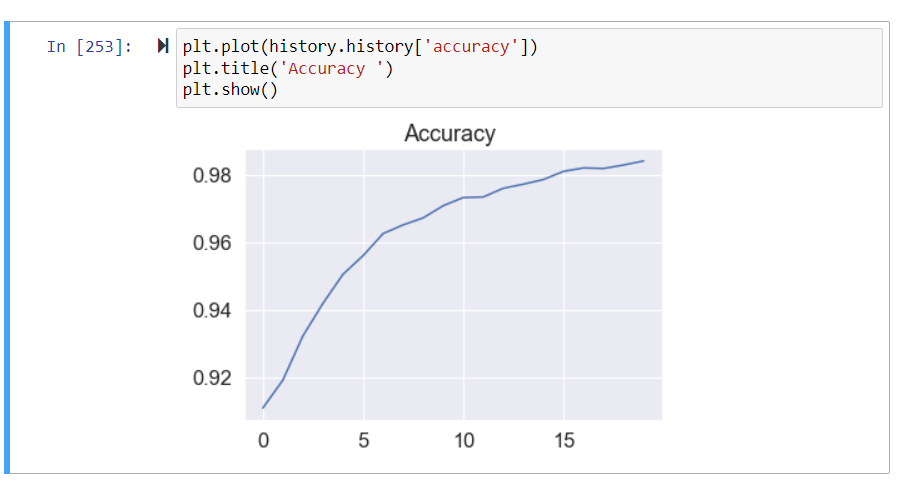
With an average accuracy of 95 percent.

### 6.3.3 Neural network evaluation

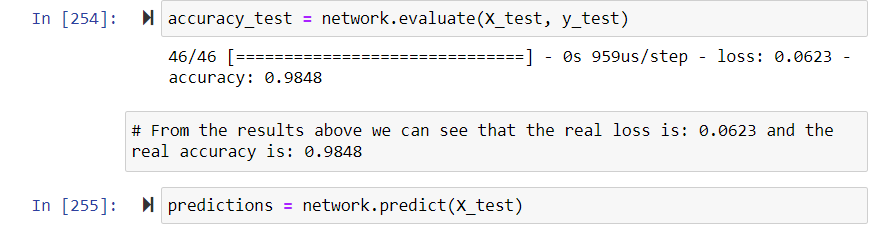
The evaluation is carried out by a plot showing the loss against accuracy. Also, an accuracy test was carried out to validate the outcome of the network.



The loss plot shows the loss in a downward slope meaning the loss dropped after each iteration.



The Accuracy plot shows the accuracy rising after each iteration which shows the neural network is very effective.

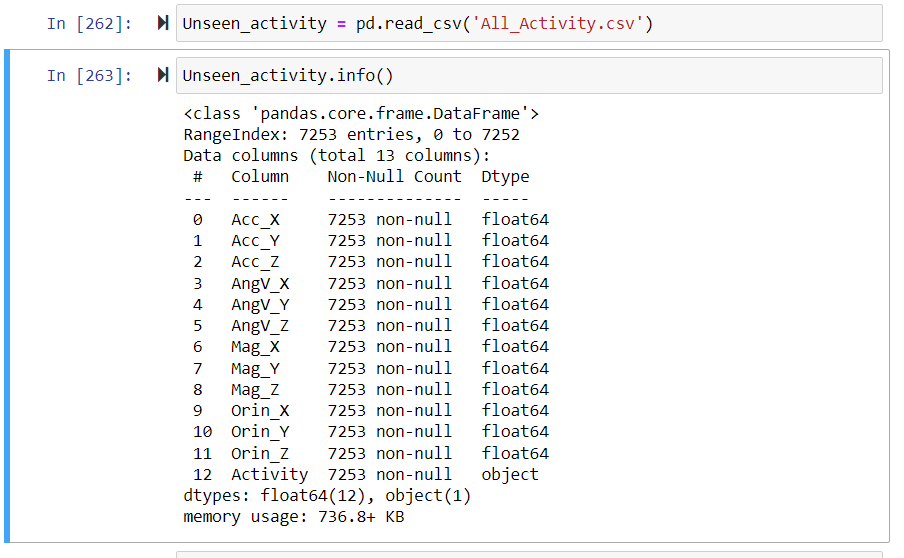


Based on predictions made from the training of the dataset, the accuracy test is carried out.

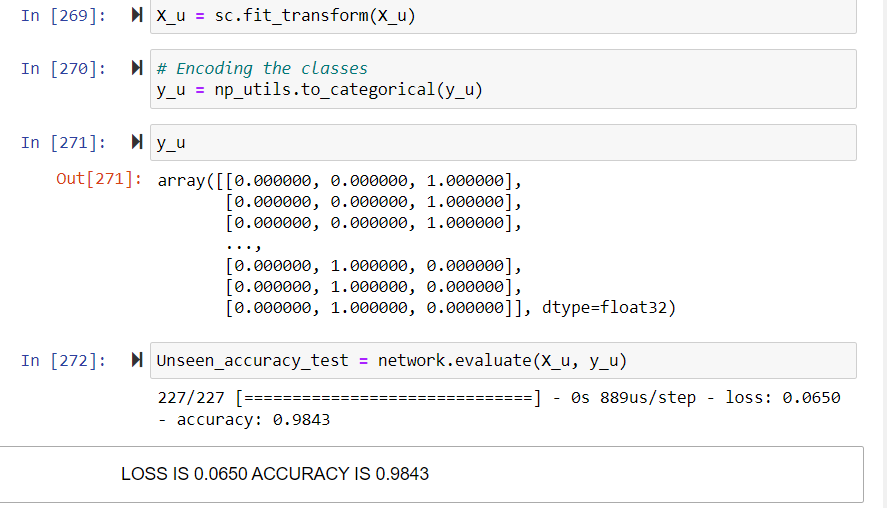
The loss is 0.0623 and the accuracy is 0.9848 which is 98 percent.

## 6.4 Test Model on Unseen dataset

This step involves importing a dataset with the same structure to test the neural network. In this report, I made use of All\_activity.csv which was a version of the first imported CSV for this test.



### 6.4.1 Evaluation of result on Unseen Test value



The accuracy of the neural network on the unseen data is 98 percent and the loss is 0.0650

# 7.0 EXPERIENCE DURING THE REPORT

My experience during the report was a good one. I had a good time from the first step which involved the generation of the datasets from Matlab to the last step which involved the training of the data sets. The implementation of the models was difficult at first, then I had to study and read up solely on the models to understand the parameters and implement them. Overall, it was a great experience working on this report.

LINK TO MY FIVE MINUTES YOUTUBE VIDEO ON YOUTUBE-<https://youtu.be/X8dYpRzG8AU>

# REFERENCES

Brownlee Jason (2019). *How to use Data Scaling Improve Deep Learning Model Stability and Performance*. [Online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/ [Accessed 30 Nov. 2021].

Sanket Doshi (2019). *Various Optimization Algorithms For Training Neural Network*. [Online] Medium. Available at: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6 [Accessed 30 Nov. 2021].

Christophe Pere (2020). *What are Loss Functions? - Towards Data Science*. [Online] Medium. Available at: https://towardsdatascience.com/what-is-loss-function-1e2605aeb904 [Accessed 30 Nov. 2021].

Gupta Dishashree (2020). *Activation Functions | Fundamentals Of Deep Learning*. [Online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/ [Accessed 30 Nov. 2021].

Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. [Online] Medium. Available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [Accessed 30 Nov. 2021].

Ashwin Raj (2020). *Perfect Recipe for Classification Using Logistic Regression*. [Online] Medium. Available at: https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592 [Accessed 30 Nov. 2021].

Koehrsen, W. (2018). *Visualizing Data with Pairs Plots in Python - Towards Data Science*. [Online] Medium. Available at: https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166 [Accessed 30 Nov. 2021].

Prasad Patil (2018). *What is Exploratory Data Analysis? - Towards Data Science*. [Online] Medium. Available at: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 [Accessed 30 Nov. 2021].

‌