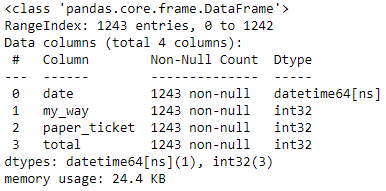
Assessment # 1

# Problem statement

* Predict the daily total patronage figure (number of boardings) on the Light Rail Network in Canberra

Data Preparation Steps

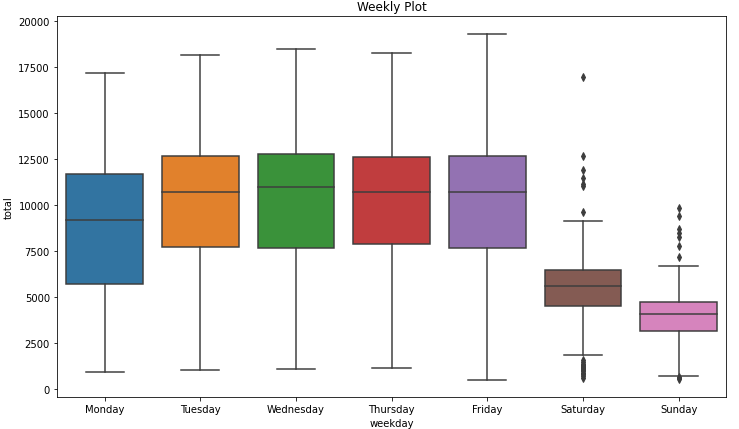
* Reading all the csv into a dataframe.
* Renaming all the column names for readability.
* Removing all special characters, punctuation and spaces from the columns.
* Changing the datatype for the column.



* Sorting the dataframe by date column to convert into Time Series problem.
* Checking the duplicate rows and missing values in the dataframe.

Exploratory Data Analysis

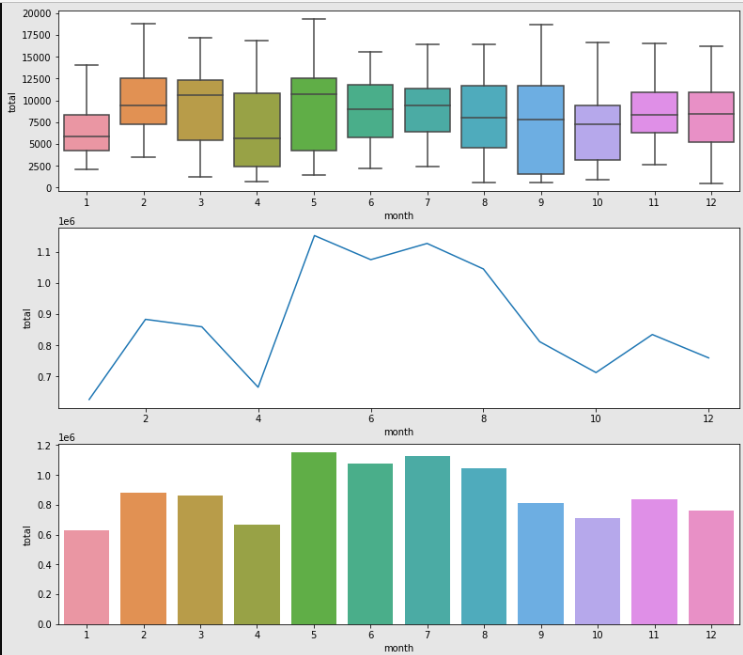
* Checking the total number of patronage in the weekly plot.



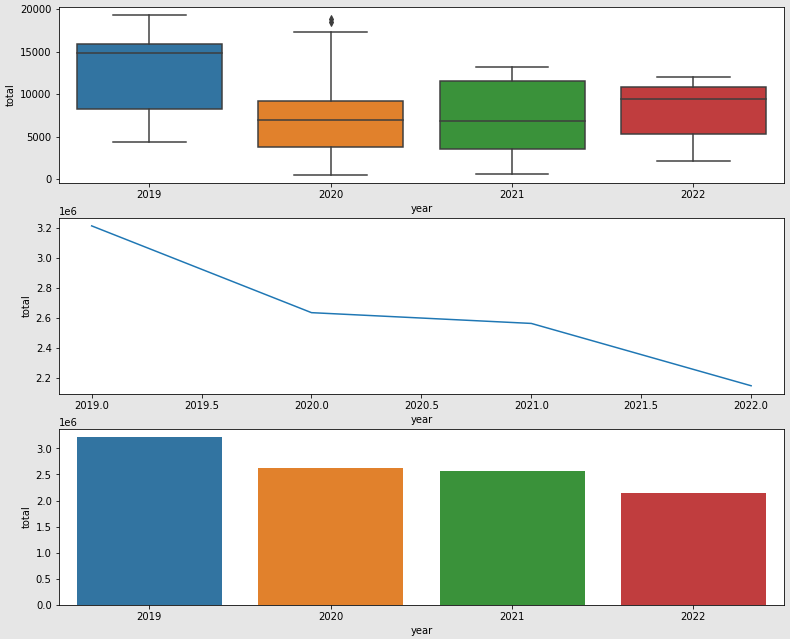
* Checking the distribution for the columns across the dates in the data.



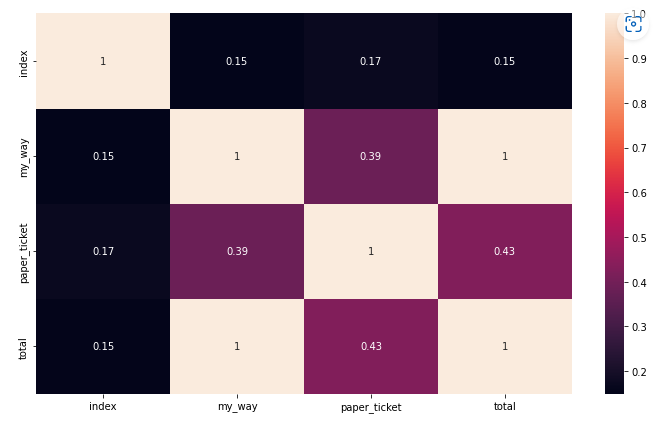
* Checking the total number of patronage in the monthly plot.



* Checking the total number of patronage in the yearly plot.



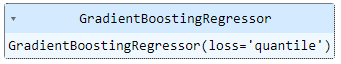
* Checking correlation between the columns in the data.
* The column ‘my\_way’ and ‘total’ are highly positively correlated based on the Heatmap.



Model Building

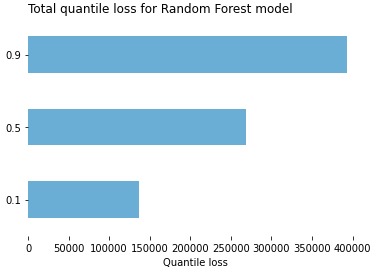
* Considering the 'my\_way','paper\_ticket','total' columns in X and ‘total’ column in Y.
* Splitting the data into train and test data with timedelta function, having last 3 months data in test data remaining data in the train data.
* Building the ML model with –Random Forest, Gradient Boost Regression models.

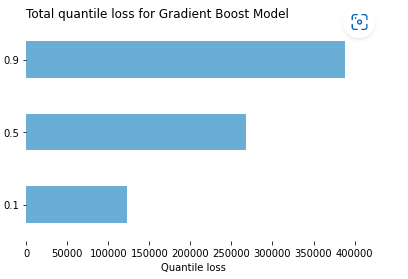




Model Evaluation

* Creating the Quantile Loss for the Random Forest and Gradient Boost Regression models.





Assessment # 2

# Problem statement

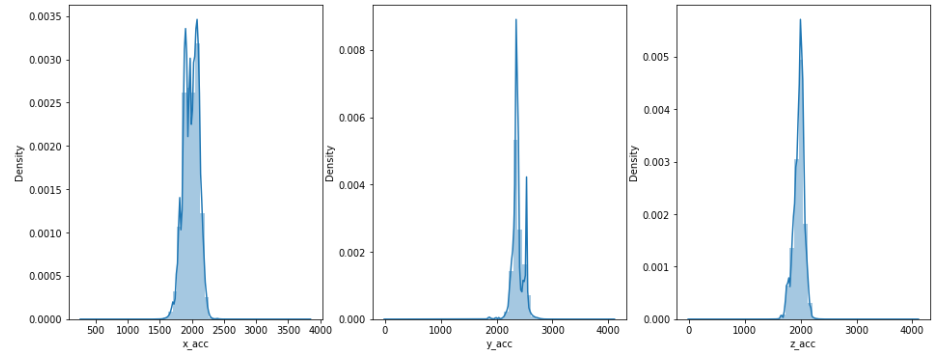
There are 15 participants who wore accelerator meter to collect data on various actions. The accelerator meter data contains x acceleration, y acceleration, z acceleration, and label. Your task is to build a model that can classify an action based on the input accelerator meter data (x acceleration, y acceleration, z acceleration)

Data Preparation Steps

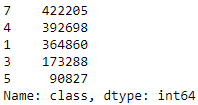
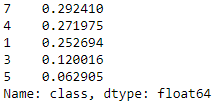
* Reading all the 15 csv files and concatenating them into a single file/dataframe.
* Checking and dropping the duplicate rows.
* Handling the multi-activities classes in the Label column (target), by replacing class 0 and class 2 as class 5, and also replacing class 6 as class 4 respectively.

Exploratory Data Analysis

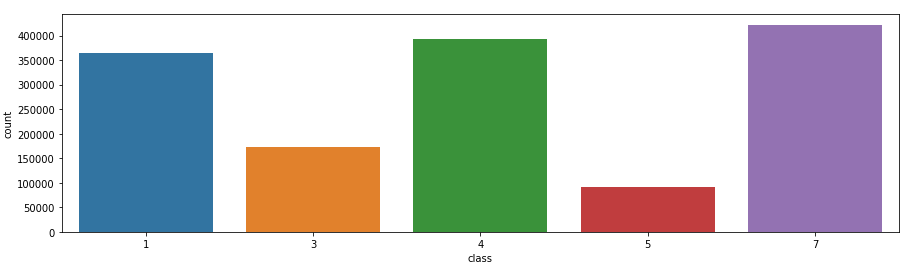
* Checking the data distribution for the continuous columns (x acceleration, y acceleration, and z acceleration).



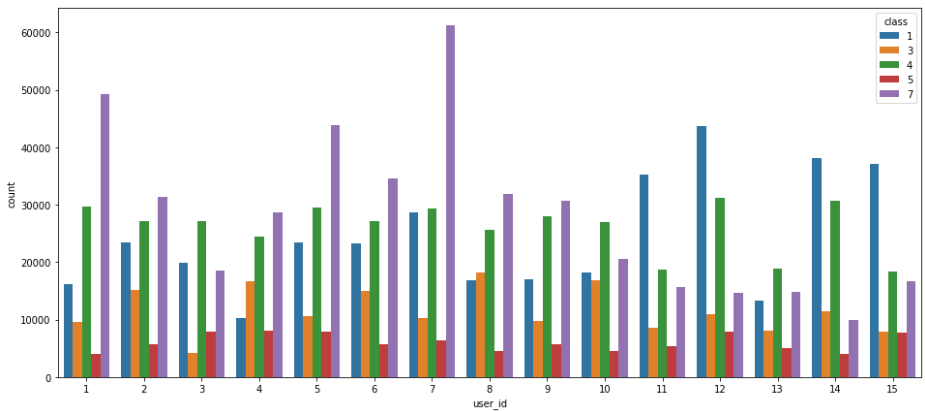
* Most(~50%) of the x\_acceleration lies in between 1700 and 2200
* Most(~50%) of the y\_acceleration lies in between 2150 and 2700
* Most(~50%) of the z\_acceleration lies in between 1800 and 2200
* Class 7 has highest count while class 5 has lowest count.

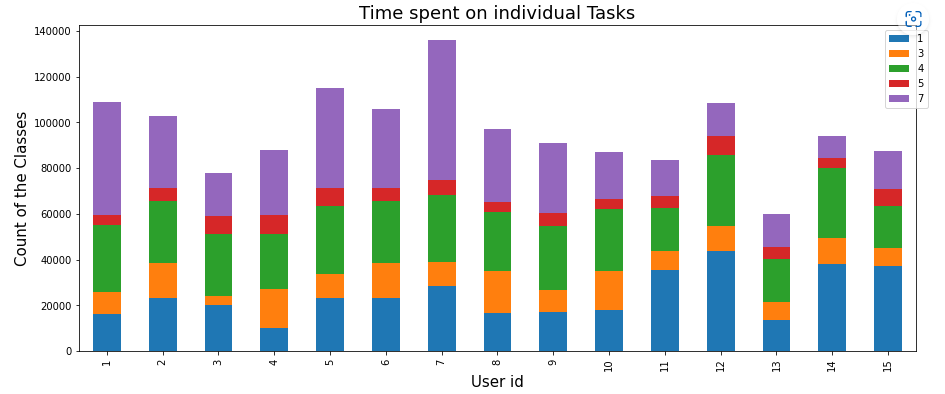
* Below is the count of the class in the target column after updating.



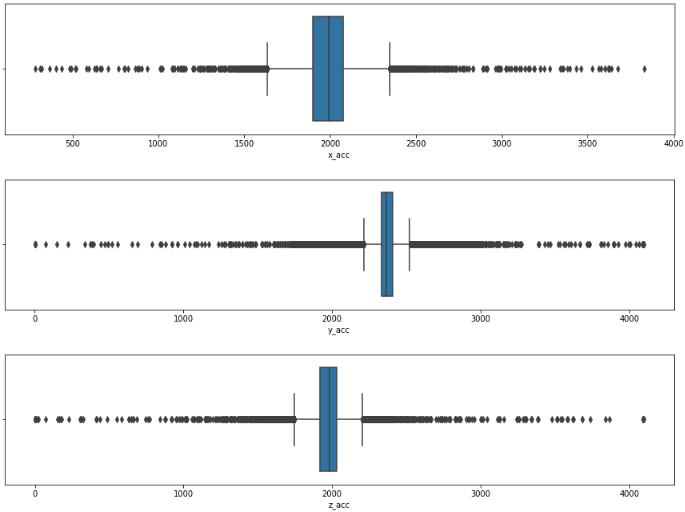
* All the users have some amount of activity in all the classes.



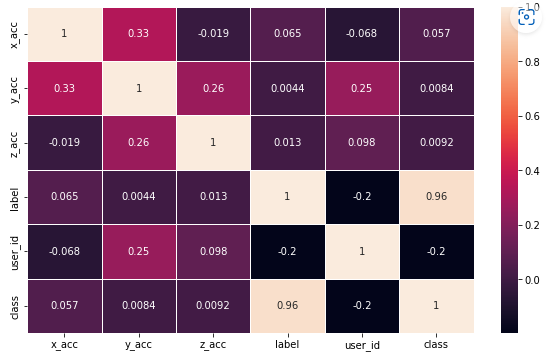
* Below is the stacked plot showing the ‘Time spent on individual Tasks’ for each user across all the classes.



* Checking outliers across the x acceleration, y acceleration, z acceleration columns.

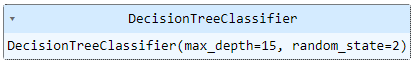


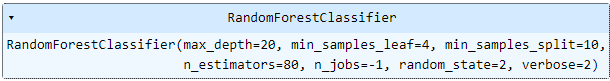
* Checking correlation for all the columns in the data.

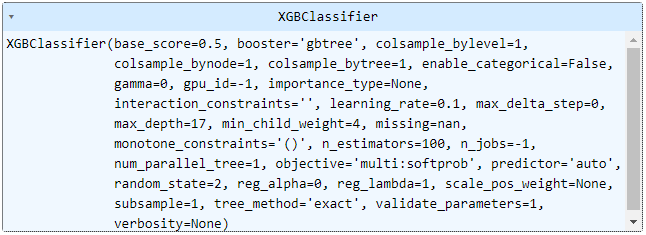


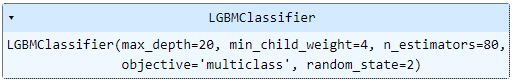
Model Building

* Considering the x acceleration, y acceleration, z acceleration columns in X and class column in Y.
* Splitting the data into train and test data with test\_size as 20%.
* Scaling the data with StandardScaler library.
* Building the ML model with – Decision Tree, Random Forest, XGBoost, LGBoost classifier models.



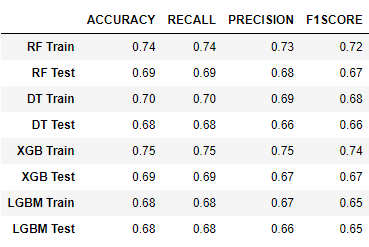






Model Evaluation

* Checking the train and test evaluation results across all the models.



* Since there is some imbalance in the classes, we can use F1 Score as the metric for evaluating the model performances.
* The Random Forest Classifier gives slightly better F1 Score than other models; hence we can select the Random Forest as the best ML model.

Interface using Flask API

* Save the best model in a pickle file to load and predict.
* Creating a POST method for the prediction model to classify which action that a participant has undertaken.
* Creating an .html file for getting the input (accelerator meter data (x acceleration, y acceleration, z acceleration) and shows the predicted class in the output along with its model prediction accuracy.

