**ReadMe Data side:**

**Workflow follows these workbooks in order**

**Points in polygon.R** – This reads in the stops.csv file (too large to be uploaded to github), merges this with the MSOA GeoJSON to perform a points in polygon analysis of each bus sotp, train station and metro/tram/underground station in the UK and outputs the locations csvs. This was also used to get the latitude and longitude of each MSOA centroid to add it to the flows data.

**Travel\_Time\_bus.ipynb (Car and Rail also)** – are scripts used to read in the distance and travel data known as ‘dis1.csv’, ‘dis2.csv’ and ‘disc3.csv’ from the Data folder, which are combined with the flow data from ‘Data/wu03ew\_msoa.csv’ to calculate the average travel time results. The results are then outputted as ‘Data/travel\_time\_bus.csv’, ‘Data/ travel\_time\_car.csv’ and ‘Data/ travel\_time\_rail.csv’.

**Merging and variable exploration.ipynb** – This is the first script that begins to merge the existing datasets. From the Data folder, this reads in ‘wu03ew\_msoa.csv’ which contains all the flow data for each msoa, the ‘Bus\_locations.csv’, ‘Train locations.csv’ and ‘Metro\_locations.csv’, which contain all the information on each number of stops/stations in each MSOA, ‘household cars.csv’ which contains all the cars in each MSOA, and ‘travel\_time\_car.csv’, ‘travel\_time\_bus.csv’ and ‘travel\_time\_rail.csv’ which contains information on all the travel times. Each of these datasets are merged into a single csv and outputted as ‘Data/cleaned\_transport.csv’. The latter parts of this script are used later and so have been changed to remove variables after initial analysis and cleaned transport is updated.

**Exploration.ipynb and Yeo-Johnson Transformation.ipynb** – These scripts take in the ‘cleaned\_transport.csv’, explore the variables and their distribution and then transform and standardise the data. The first performs the log transformation then each of the standardisations, the latter performs the Yeo-Johnson transformation and following standardisations. These result in the csvs within the Data folder of ‘transport\_log\_then\_normal.csv’, ‘transport\_log\_then\_range.csv’, ‘transport\_log\_then\_idr.csv’, ‘yeojohnson\_range\_st.csv’, ‘yeojohsnon\_zscore\_st.csv’ and ‘yeojohnson\_idr\_st.csv’.

**Clustering\_templace.ipynb and other clustering workbooks** – The clustering template explains the clustering methodologies we used, along with how they work and their code, which was subsequently used in the other clustering workbooks. These workbooks perform the clustering algorithms on csvs produced above, and outputs the results in the Data/Clustering\_Results folder, with the clustering method added to the end of the name of each file.

**Evaluating clusters.ipynb and mapping clusters.R** – These workbook reads in the clustering results and produces plots and maps to explore the results. The latter merged all the clustering outputs into a single csv called ‘clusterlabels\_cleaneddata.csv’ which is in the clustering\_results folder. These plots, along with maps produced in RStudio were used to analyse the clustering results to reach the final conclusion of which clustering results performed best.

**Clustering\_Json\_Script.ipynb** – This used the ‘clusterlables\_cleaneddata3.csv’ to create a json to put on the Firebase database.

**Socioeconomic Varibles.ipynb** – This workbook produces the geodemographic variables csv known as ‘Data/Socioeconomic\_data/socioeconomic\_all.csv’ by cleaning and combining all the socioeconomic factors into a single dataframe. These factors are then to be used in the classification workbook.

**Classification workbook.ipynb** – This workbook performs the final classification results and explains the method and process, outputting the RandomForest results as pngs.

**Data for story.ipynb** – uses the existing data from the previous workbooks to produce summary statistics that will be used to inform the story and on the website through grouby methods.

**Flows to sql-Copy1.ipynb** – This workbook is used to put all the resulting information onto sql which is then used through the API (with all personal information removed).