OHEMENG KOFI NTIKORAH

PORTFOLIO

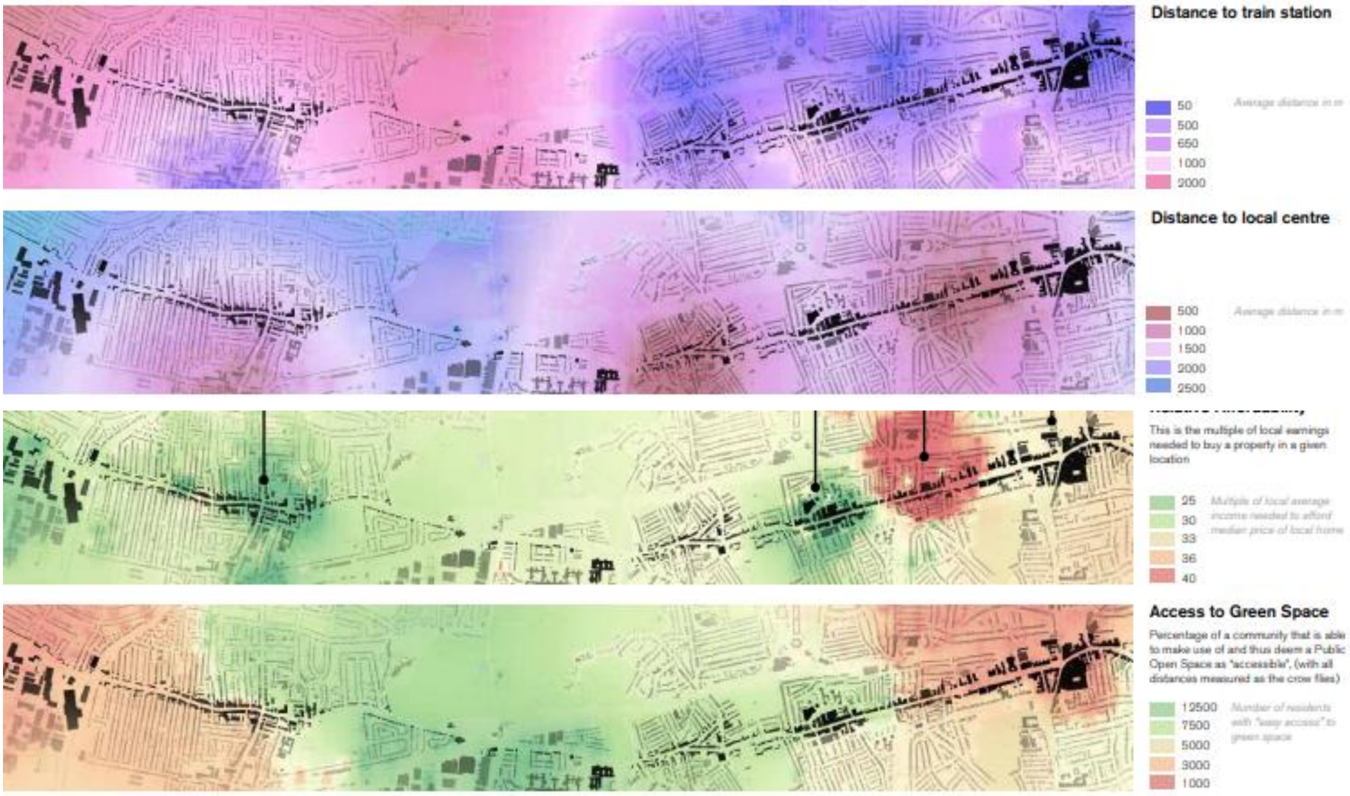
SPATIAL ANALYSIS AND SETTING UP OF A PIPELINE FOR SPATIAL ANALYSIS
GIS CONSULTANT, WALULEL

This project required scripting spatial processes which were used to analyze metrics such as air quality, missing postcodes, background noise and area friendliness for London. At the end of the project, scripts written in R programming languages were stored in the cloud on bit bucket, these scripts with the aid of DRAKE::R (MAKE) could be run in R studio to automate various spatial processes. The figures below show script written to track updates of postcodes in London. Added postcodes are termed as missing postcodes. Five nearest postcodes to original postcodes are determined after which a count of these missing postcodes are calculated for boroughs and sectors in London. This and many other scripts are uploaded to bit bucket with other dependencies to be run in RStudio.

```r
library(stringr)
library(rgdal)
library(dplyr)
library(sf)
library(nngeo)
library(spatstat)
library(tidyr)
library(readxl)
library(tibble)
library("xlsx")
library(GISTools)
library(Lahman)

p= read.csv("trypostcodes.csv")

ukwide= read.csv("nspl.csv")

BoroDiss <- readOGR("dissolved borough.shp")
r <- p[c("X","Y", "field_1_1")]
coordinates(r) <- ~X + Y

rsf <- st_as_sf(r, crs = 3857)
st_crs(rsf) = "+init=epsg:3857"
class(rsf)
st_crs(rsf)

U <- ukwide[c("long","lat", "pcd")]
coordinates(U) <- ~long + lat

usf <- st_as_sf(U)
st_crs(usf) = "+init=epsg:4326"
usftrans <- st_transform(usf, crs = 3857)
class(usftrans)
st_crs(usftrans)
Nearest_5 <- st_nn(missing_r,rsf_as_sp_as_sf, returnDist = TRUE, k=5)
Nearest_5.df <- as.data.frame(t(do.call(rbind.data.frame,Nearest_5)))%>%
  select(dist.6,dist.7,dist.8,dist.9,dist.10)

final3 <- st_join(missing_r,rsf,st_nn, k=5)
final4 = final3%>%
  group_by(field_1_1.x)%>%
  mutate(Nearest_5 = seq_along(field_1_1.y))%>%
  spread(key = Nearest_5, value = field_1_1.y)


joined_spread = cbind(final4, Nearest_5.df)



Borough <- readOGR("Undissolved Borough.shp")


Borough.tr <- spTransform(Borough,  CRS("+init=epsg:3857") )

Borough <- spTransform(Borough, CRSobj = CRS(proj4string(clipselect_as_sp)))
poly.counts(clipselect_as_sp, Borough) -> All_UK_Count_Postcodes_per_Borough_202
setNames(All_UK_Count_Postcodes_per_Borough_2020, Borough.tr@data$NAME_1)
View(Borough.tr@data)

head(Borough.tr)
dfbor<- as.data.frame(All_UK_Count_Postcodes_per_Borough_2020)
dfborpoints <- cbind(dfbor, Borough.tr)



Sector <- readOGR("Sector.shp")
```

HEATMAP COMPOSITION FOR RADIAL ROUTES BY TRANSFORMING THOROUGHFARES INTO BOULEVARDS
JAS BHALLA ARCHITECTS WITH WALULEL
GIS CONSULTANT, WALULEL

The William Sutton Prize for Place making and Affordable Housing Designs is a competition held to harness potential urban insights innovations. In 2019 Jas Bhalla Architects with Walulel won the prize for affordable housing designs. Walulel provided geospatial intelligence by composing heat maps for two areas in London, St. Andrews Park, Uxbridge Road, Hillingdon and 505-509 Uxbridge Road, Hillingdon. The primary assessment criteria was to identify routes which were suitable for densification. Using GIS, a workable solution was presented by presenting analytical mapping which could be used to increase the speed of site selection. Heat maps generated incorporated social conditions, connectivity and environmental impacts. Some datasets used were, relative affordability, green space accessibility, air quality, income levels, distance to train stations.



Safegraph is a data provider with specialty in providing points of interest data (POI), foot traffic data and building footprints. I was contracted to evaluate places data for areas in US, Canada and Great Britain by assuming the persona of the director of data acquisitions for UBER In the end I presented an evaluation report. The report gave context to visitor locations to be used for advertisement. It also provided meaning to navigation purposes in terms of pick up and drop off locations. Recommendations were made for robust data provisions along with insights which could be generated from data. The reasons to opt for Safegraph over other data providers were also highlighted in the report. Geoprocessing algorithms such as proximity/buffer, counting algorithm, OD destination matrix, and line to path in QGIS were used. R studio was also used to validate distance matrix output from QGIS.

Testing POI

This was done to serve as a base for the POIs to be analysed/evaluated by gridding Safegraph polygon to estimate POI square footage of the building. This was achieved by creating 1m grids on the coverage of these POIs(Vancouver, Illinois and Greater Manchester) to cover the extent of POIs.

Method

- Converting latitude and longitude to projected coordinates(EPSG: 3857) to make it planar
- Using a grid algorithm, create a 1m by 1m grids and display them as polygons. This creates a network of rectangles on the POI area (coordinates displayed as left, top, bottom, right)

Result

All POIs within a particular grid box were captured. This coordinate is compared to the coordinates from google map.

Validation

This shows refined granularity of Safegraph's data. All POIs fall at the centre of polygons and within specific grids. This validates the square footage of buildings and when compared with google map, polygons were just as close.

USE CASES

***Use Case 1:** Contextualizing the types of places people visit for advertising purposes.*

To clearly understand this, it is important to know how places data are clustered.

Method

Buffer (500m) : A 500m buffer was defined around POIs(Vancouver, Manchester, Chicago points). This segregated the POIs, giving clear understanding of POIs which are many in the buffer. The greater the number of POIs within the buffer the greater the likelihood to attract customers to these POIs. This enables customer attribution which is helpful for future customer behavior predictions.

GEOSPATIAL DATA, MOBILITY DATA SOURCING EXPERT

The ever-pressing need for mobility data in the geospatial sector is notable. As part of harnessing the true potential mobility data brings, I was contracted to source mobility data from providers. In sourcing data activities included signing onto geospatial platforms, making enquiries on websites of providers, posting data requests onto platforms, scheduling meetings with data providers to discuss data needs and uses cases. At the end, providers who met data needs were compiled on a Google sheet based on a number of criteria. A document was provided to that effect. Spatial analysis was used to test data samples from providers. Movinpandas, a python library was used to visualize mobility data. The second part of this project required sourcing geospatial data from providers such as Open Street Map, data. World to mention. Other geospatial platforms were studied and reviewed. A Google document and a Google sheet was created at the end of reviews. Data sourced was validated using geospatial processes in QGIS, R and python. The end goal is to build a geospatial platform to provide location intelligence.

Grou	https://www.ep	Census LEHD 201	Assessing and
Grou	https://www.ep	Travel time API, H	Scenario plann
Grou	https://www.ep	National Transit D	Evaluating and
Grou	https://www.ep	2019 Census Tige	Scenario plann
	Link 1	**LInk 2**	**Source**
build	https://www.eia.gov/consumption/		Energy Inform
grou	https://hudgis-hu	https://hudgis-hu	Office of Polic
	https://www.mrl	https://data.cityc	National Land

IDEAL RECOMMENDATION Provision of datasets on foot traffic, unique identifiers, geographic coordinates granularity of pois with api requests.This study was targeted to assessing data providers with lowest pricing packages but some requirements still needed to be set to sieve the few from the many. Based on these requirements, four data providers were considered. These data providers are **GRAVY ANALYTICS, PREDICIO, INTUIZI, KUWALA and CARTO.**

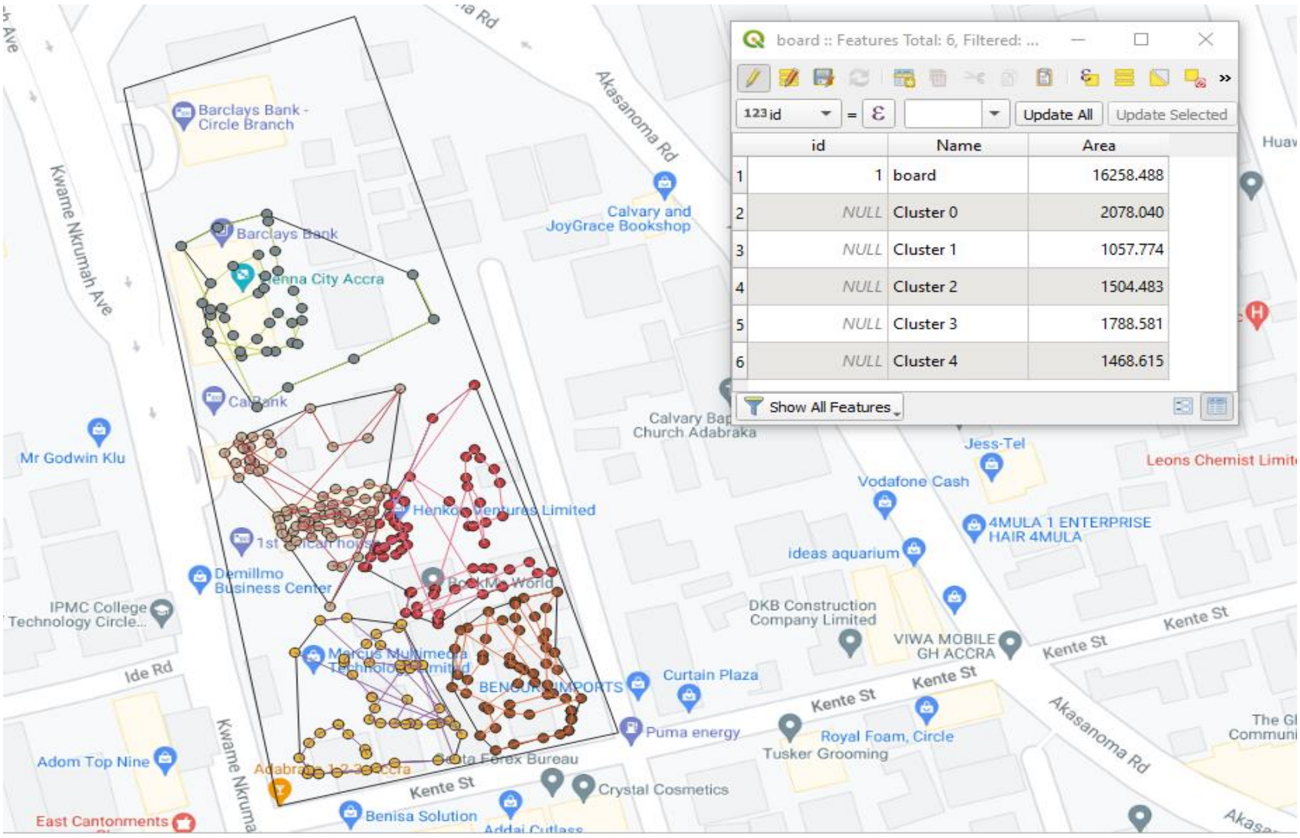
Other recommendations

Pricing. On the geospatial market, the average amount quoted by providers is $5000/month. These providers fall below this price but offer data for foot traffic attribution. **GRAVY ANALYTICS AND PREDICIO** satisfy this requirement. Based on foot traffic use case, data can be purchased at $2000 from these two providers. **INTUIZI** offers a $3000 package which is negotiable.

Package Offerings. Though this project seeks to use live stream data, in a case where data is needed to be purchased one-time, these providers offer moderate payment plans. Kuwala offers this one-off package at 9700 euros. Intuizi provides a 3 month basic subscription at

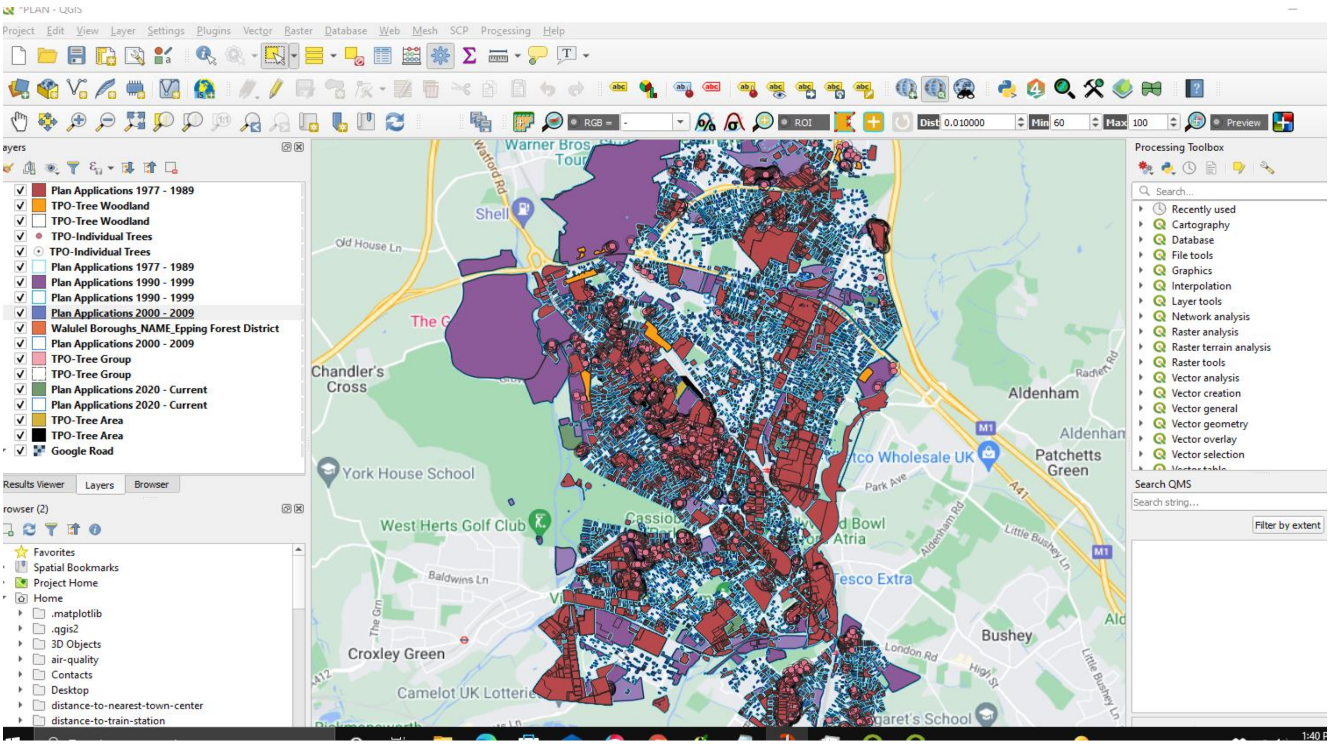
SPACE USE RATIO FOR GEOLOCATION

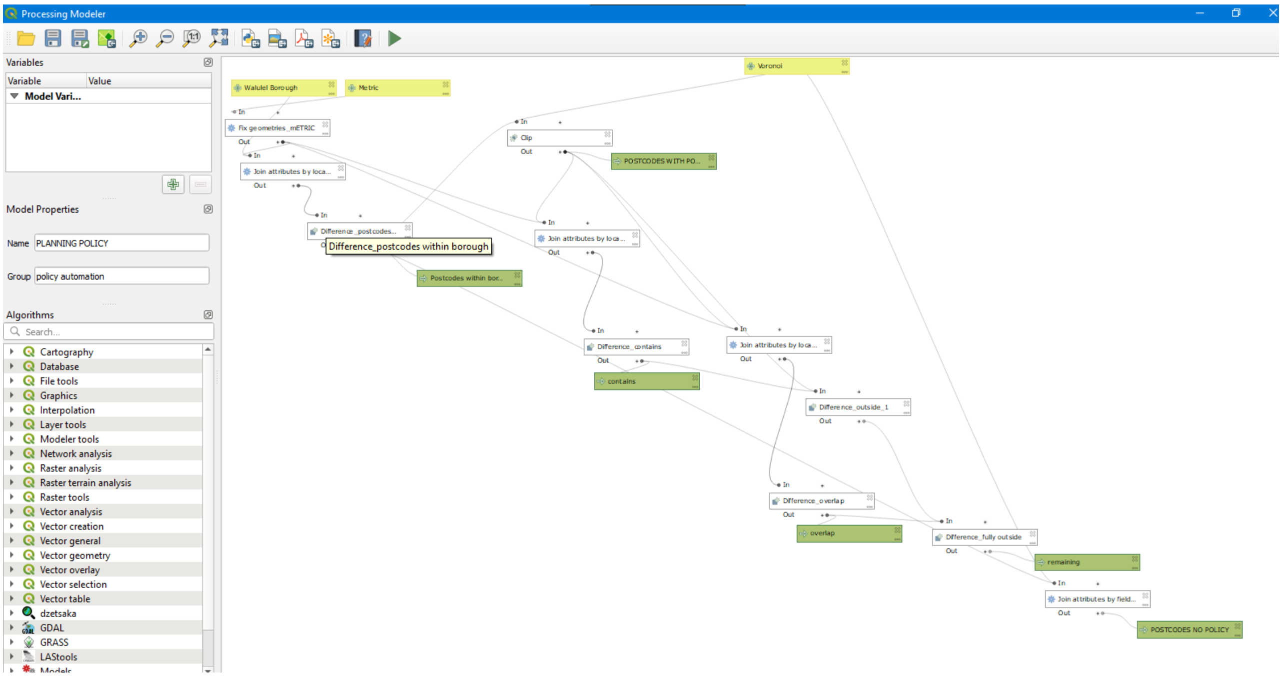
The outcome of this project was to implement a geolocation framework which can help to track users on WaInsight platform for advertisement. A space use ratio was developed in QGIS to set the grounds on feature development.



SCRAPING DATA FOR POLICY PLANNING- model, python script, r geojson scrape

To provide insight on policies, layers were retrieved and scraped using data from wms, wfs, wcs, rest api connections to be analysed. On this project, policy layers via these connections were added to QGIS environment and analysed to generate postcodes within these policies.





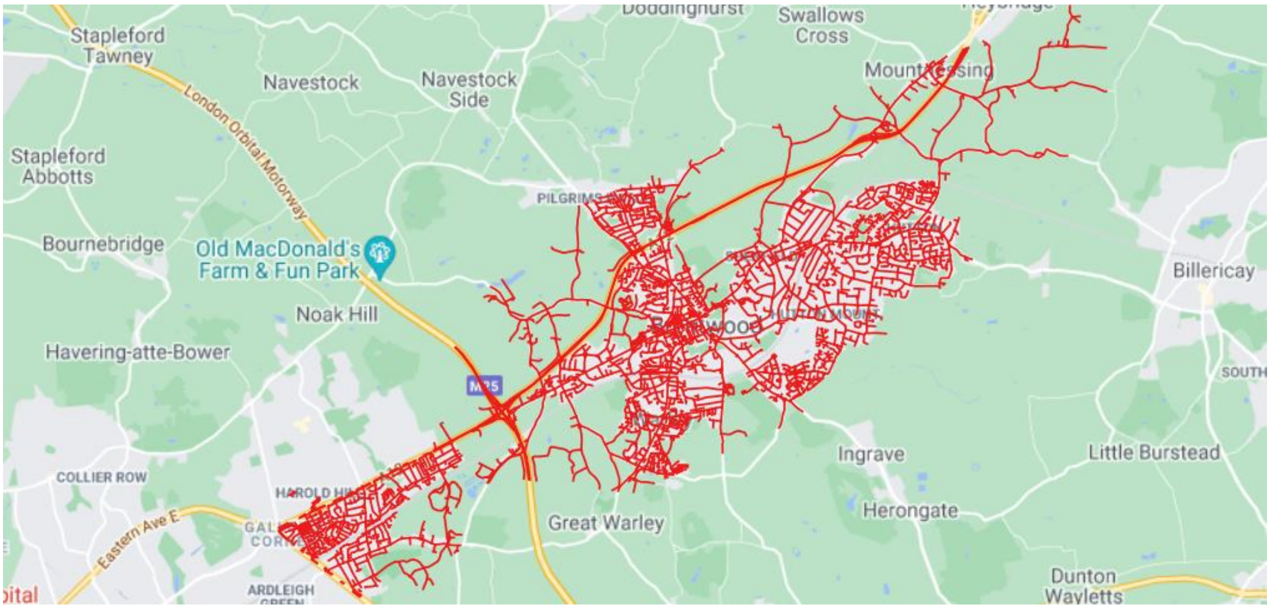
```r
library(sf)
library(geojsonsf)
##install.packages('Rcpp')
library(Rcpp)
library(sp)
library(rgdal)
library(dplyr)
url <- "https://www[redacted]uk/site/custom_scripts/repo
sf <- geojson_sf(url)
##rename sf to geo
geo <- sf
##find class  and vector types of geo and vector types
class(geo)
unique(geo$DESIGNATION)
unique(types)
##extract polygons, lines and points from geo
geo2 <- st_collection_extract(geo, "POLYGON")
geo3 <- st_collection_extract(geo, "LINE")
geo4 <- st_collection_extract(geo, "POINT")
##export extracted polygons. lines and points to shapefile
st_write(geo2, "geo2.shp")
st_write(geo3, "geo3.shp")
```

```python
            'YFIELD': 'Y',
            'ZFIELD': None,
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['CreatePointsLayerFromTable'] = processing.run('qgis:createpointslayerfromtable', alg_params, context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(9)
        if feedback.isCanceled():
            return {}

        # Delete duplicates by attribute
        alg_params = {
            'FIELDS': 'field_1_1',
            'INPUT': outputs['CreatePointsLayerFromTable']['OUTPUT'],
            'OUTPUT': parameters['Polygpoly']
        }
        outputs['DeleteDuplicatesByAttribute'] = processing.run('native:removeduplicatesbyattribute', alg_params, context=context, feedback=feedback, is_child_algorithm=True)
        results['Polygpoly'] = outputs['DeleteDuplicatesByAttribute']['OUTPUT']
        return results

    def name(self):
        return 'policy'

    def displayName(self):
        return 'policy'

    def group(self):
        return 'modelplanpol'

    def groupId(self):
        return 'modelplanpol'

    def createInstance(self):
        return Policy()
```

```python
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingParameterVectorLayer
from qgis.core import QgsProcessingParameterFeatureSink
from qgis.core import QgsCoordinateReferenceSystem
import processing


class Policy(QgsProcessingAlgorithm):

    def initAlgorithm(self, config=None):
        self.addParameter(QgsProcessingParameterVectorLayer('boroughpostcodes', 'Policypolygon', types=[QgsProcessing.TypeVectorPolygon], defaultValue=None))
        self.addParameter(QgsProcessingParameterVectorLayer('policy', 'postcode_voronoi', types=[QgsProcessing.TypeVectorAnyGeometry], defaultValue=None))
        self.addParameter(QgsProcessingParameterFeatureSink('Polygpoly', 'polygpoly', type=QgsProcessing.TypeVectorAnyGeometry, createByDefault=True, defaultValue=None))

    def processAlgorithm(self, parameters, context, model_feedback):
        # Use a multi-step feedback, so that individual child algorithm progress reports are adjusted for the
        # overall progress through the model
        feedback = QgsProcessingMultiStepFeedback(10, model_feedback)
        results = {}
        outputs = {}

        # Fix geometries_policypoly
        alg_params = {
            'INPUT': parameters['boroughpostcodes'],
            'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT
        }
        outputs['FixGeometries_policypoly'] = processing.run('native:fixgeometries', alg_params, context=context, feedback=feedback, is_child_algorithm=True)

        feedback.setCurrentStep(1)
        if feedback.isCanceled():
            return {}
```

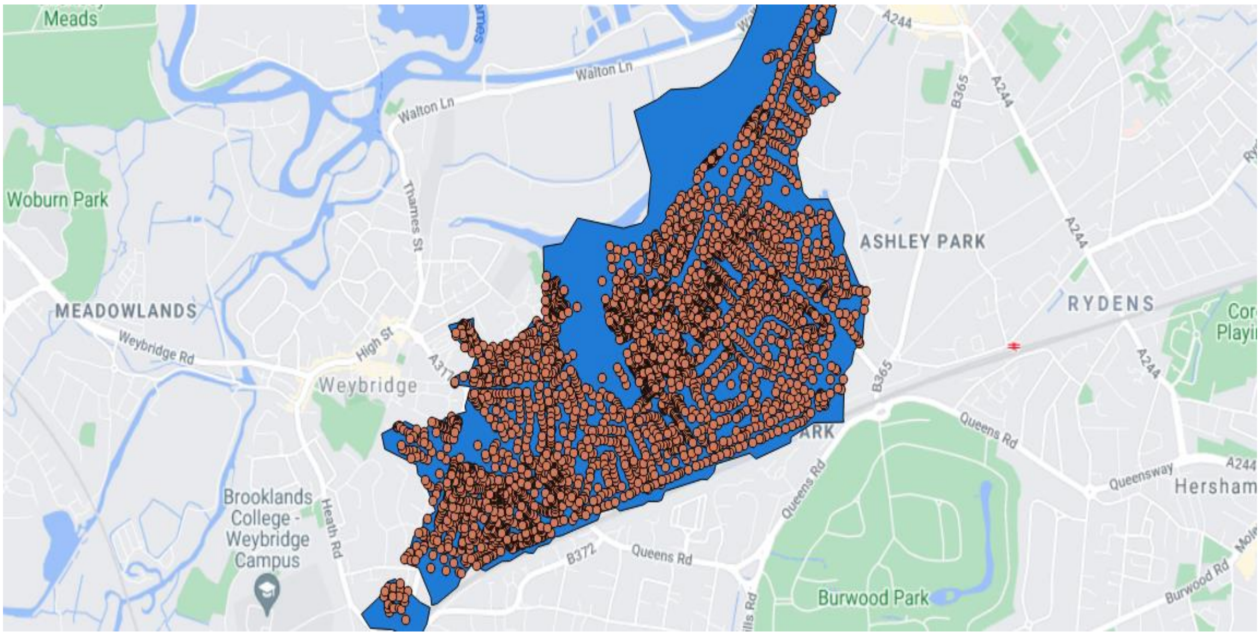
ROAD CLASSIFICATION

On this project, a database of classified roads was built. This informs users of possible alternative routes to use.



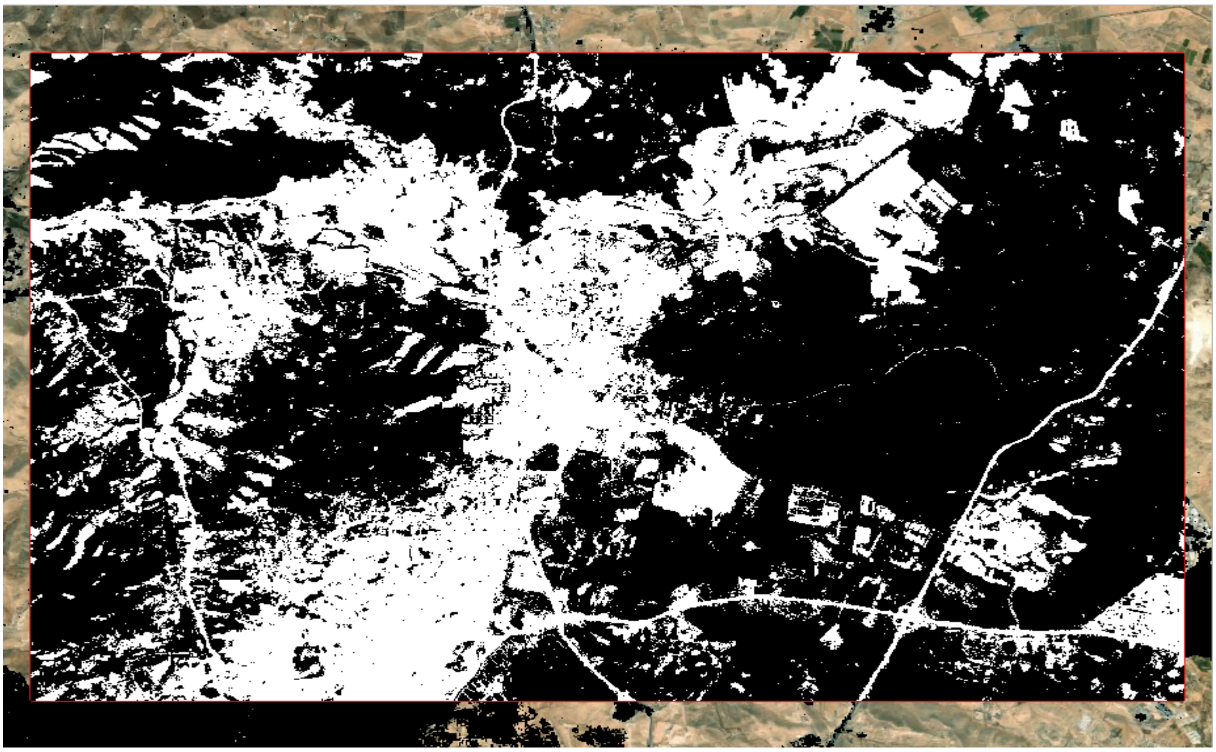
BUILDING CLASSIFICATION

Buildings were classified by style, use and architecture using QGIS and google maps.



MACHINE LEARNING WITH QGIS

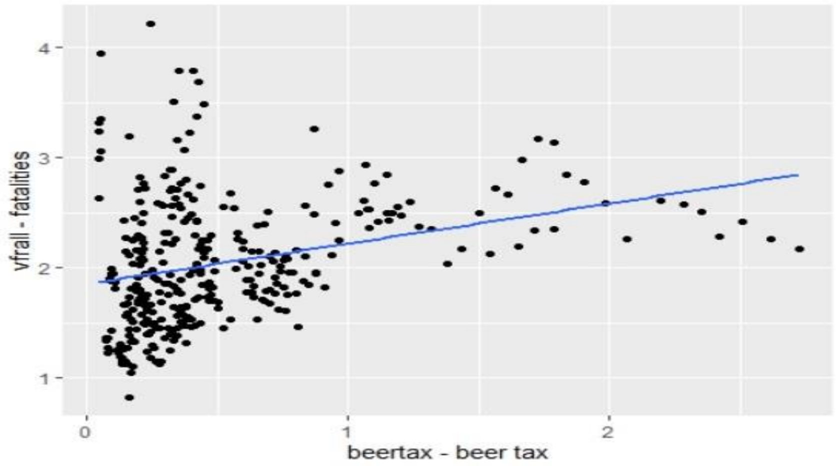
Dzetsaka, a machine learning algorithm was used to classify built areas and no-built areas.



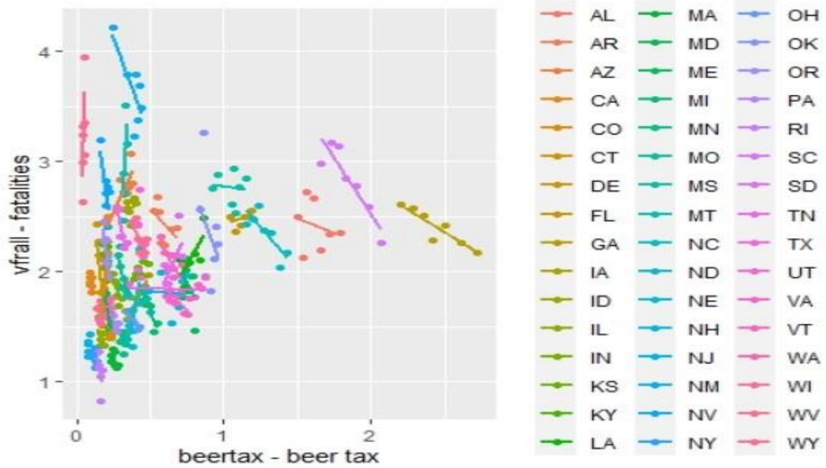
REGRESSION AND ECONOMETRICS

Regression models were built and graphs were interpreted.

```
##i
ggplot(fatality,aes(x =  beertax, y = vfrall)) +
  geom_point() +
  geom_smooth(method="lm",se=FALSE) +
  labs(x = 'beertax - beer tax', y = 'vfrall - fatalities')

## `geom_smooth()` using formula 'y ~ x'
```




```
##ii
ggplot(fatality,aes(x =  beertax, y = vfrall,colour = state)) +
  geom_point() +
  geom_smooth(method="lm",se=FALSE) +
  labs(x = 'beertax - beer tax', y = 'vfrall - fatalities')

## `geom_smooth()` using formula 'y ~ x'
```




```
##2b
model2b = lm_robust(vfrall ~ beertax, data = fatality, fixed_effects = state)
tidy(model2b)
```

DATA VISUALISATION OF PROJECT AREAS

Cartographer, Freelance

Composed a map of a forest reserve and its enclaves, Achimota Forest, Ghana. Map was generated in QGIS by analyzing area of study and applying appropriate geo-algorithms. The map seeks to define the surroundings of the rich Achimota Forest Reserve and its environs.