Configuration Manual

MSc Research Project
Data Analytics

Mark Meyler
Student ID: x21171254

School of Computing
National College of Ireland

Supervisor: Mohammed Hasanuzzaman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mark Meyler
Student ID:	x21171254
Programme:	MSc in Data Analytics **Year:** 2025
Module:	Research Project
Supervisor:	Mohammed Hasanuzzaman
Submission Due Date:	11/12/2025
Project Title:	Identifying Theft Hotspots: A Machine Learning Approach to Aggregated Data Analysis

Word Count: 3597 **Page Count:** 12

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Mark Meyler

Date: 10/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Mark Meyler
Student ID: x21171254

1 Introduction

This configuration manual details the process to replicate the results presented in the research paper for 'Identifying Theft Hotspots: A Machine Learning Approach to Aggregated Data Analysis'. The method is set up to use a Geographically Weighted LightGBM (GW-LGBM) to disaggregate crime data at a coarse aggregation level to a finer resolution.

Note the terms administrative boundary level and aggregation level are used interchangeably throughout.

2 System Configuration

2.1 Hardware Specifications

The process was performed on a device with the following specifications:

Table 1: Hardware Specifications

Component	Specification
Processor	Intel(R) Core(TM) Ultra 7 165H (1.40 GHz)
RAM	32 GB
Storage	954 GB
Operating System	Windows 11 Home, 64-bit

2.2 Software Requirements

The full methodology was developed and performed in VSCode using Python 3.12.7. The below Python packages were used at different stages of the process. Packages noted as 'Standard' are part of the standard Python library and do not need to be installed.

Table 2: Python Packages

Package	Version
pandas	2.2.2
numpy	1.26.4
geopandas	1.1.1
zipfile	Standard
shapely	2.1.1
tqdm	4.66.5
matplotlib	3.9.2

mapclassify	2.10.0
warnings	Standard
contextily	1.6.2
pysal	25.7
esda	2.8.0
seaborn	0.13.2
splot	1.1.7
statsmodels	0.14.2
scikit-learn	1.5.1
scipy	1.13.1
random	Standard
shap	0.47.2
lightgbm	4.6.0

3 Datasets

3.1 Crime Data

The crime data used in the analysis has been sourced from the Police UK open database[1]. This database allows for a custom download of street-level crime information broken down by police force. Data can be extract over a rolling three year period. For the purpose of this project, the selected field are as per Fig 1. The output after pressing ‘Generate File’ comes in the form of a zip file containing an individual folder for each month, with each month possessing a CSV file of the data. The code described later in this manual is designed to input this zip folder in the format it has been downloaded.

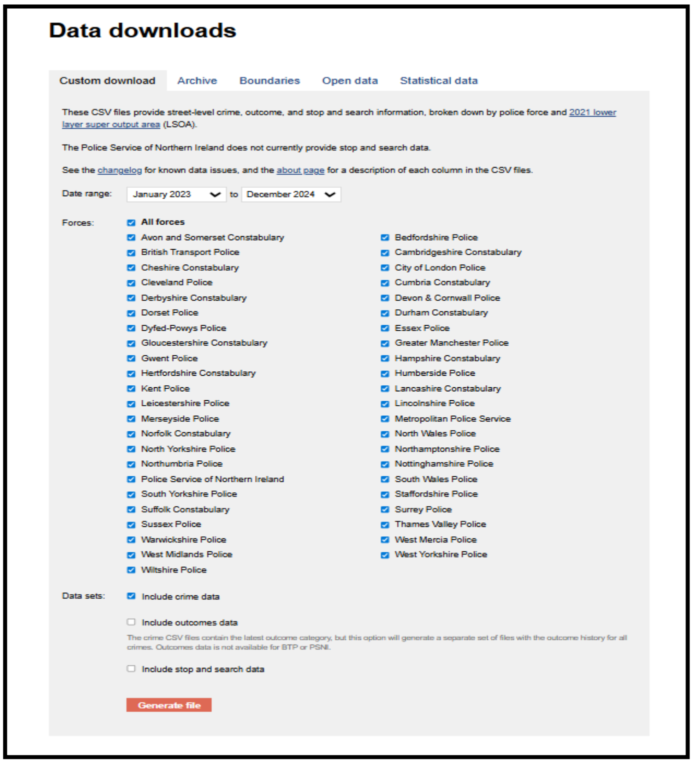

Fig 1. Selections used in Police UK Open Database to generate Crime Data

[1] Available at: https://data.police.uk/data/

NOTE: As part of the submission, the cleaning of this data into a burglary dataset has already been performed and provided in a CSV file. This is because the raw crime data is over 1.6GB when zipped and is too large to submit. As a result, the code has been adapted to just read in this CSV file, but to replicate the process undertaken in the project the if data has been downloaded as per Fig 1, relevant code just needs to be uncommented.

3.2 Boundary Data

The boundary data refers to the administrative area boundaries in the UK. As part of this project, the structure has been set up to use five different sets of boundary data:

1. **Regions[2]:** Highest tier of division in England with 9 regions. Used to define the study area of the analysis
2. **Local Authority Districts[3]:** 331 districts in England and Wales, 309 districts in England alone
3. **MSOA[4]:** Comprised of 2,000 to 6,000 households with a population between 5,000 and 15,000
4. **LSOA[5]:** Comprised of 400 to 1,200 households with a population between 1,000 and 3,000
5. **OA[6]:** Lowest geographical area, comprised of 40 to 250 households with a population between 100 and 625.

The data is downloaded for each of these boundary levels as a Shapefile, which comes within a zip file. For simplification, the names of these files have been amended to 'REGION_Boundaries', 'LAD_Boundaries', 'MSOA_Boundaries', 'LSOA_Boundaries' and 'OA_Boundaries', with the code set up to use them in this format.

NOTE: Due to size restrictions of the submission on Moodle, data has only been provided for Regions 'East of England', 'London', 'South East', 'South West' for the MSOA and LSOA areas. This prevents the replication of experiments conducted that utilise all of England.

3.3 Census Data

Census data has been extracted acquired from Office for National Statistics (ONS)[7]. Each variable in Table 3 needs to be extracted in CSV format for each administrative boundary level in the analysis, unless otherwise stated. These CSVs are provided in separate folders within the 'RAW_CSVS' folder labelled with the aggregation level, i.e. 'MSOA'. CSVs need to be named and stored in the folders as per Table 3 in order for the 'Data Cleaning' processes to work as expected.

The majority of the data used came from readily available datasets that can be found by searching for the 'Dataset ID' shown in Table 3. Other data was sourced as follows

[2] https://geoportal.statistics.gov.uk/datasets/9ab00f1e709b4423a9f70c748ee5951e_0/
[3] https://geoportal.statistics.gov.uk/datasets/52b4dd427ca54be8a2f7fa1abeb264d6_0/
[4] https://geoportal.statistics.gov.uk/datasets/3b3b1cb15a744855881b1489741b861b_0/
[5] https://geoportal.statistics.gov.uk/datasets/2bbaef5230694f3abae4f9145a3a9800_0/
[6] https://geoportal.statistics.gov.uk/datasets/f23c50d2bffa4a8d9693bcf4ed72bdaf_0/
[7] https://www.ons.gov.uk/

- The 'Age' variable was custom created[8] from ONS using the variable and categories shown in Table 3. Categories refers to the available groupings within the variable.

- Workplace population density was extracted from a different ONS source[9]. The Dataset ID produces a zip file which contains extracts for each administrative boundary area. The relevant file for the analysis needs to be extracted and saved with the respective 'CSV Name' shown in Table 3.

- Education establishments[10], public transport access nodes[11] and parks and gardens[12] were extracted from Gov.UK, and are aggregated to the required administrative boundaries within the code. These are kept in a separate folder to all other data, as they are re-used for all administrative boundary working datasets.

NOTE: Again due to size restrictions only census data in respect of MSOA and LSOA boundaries has been provided.

Table 3. The source of each of the variables extracted from census data, and the folder and CSV names required to utilise the code

Folder	Dataset ID	Variable	Categories	CSV Name
RAW_CSVs	N/A	Age	6	age_grouped_6
RAW_CSVs	TS004	Country of Birth	12	area_of_birth
RAW_CSVs	TS062	NS-SEC	10	class_type
RAW_CSVs	TS011	Deprivation Dimensions	6	deprivation_dimensions
RAW_CSVs	TS067	Education	8	education
RAW_CSVs	TS037	General Health	6	general_health
RAW_CSVs	TS044	Accommodation Type	8	household_classification
RAW_CSVs	TS054	Household Tenure	9	household_tenure
RAW_CSVs	TS063	Occupation	10	occupation_type
RAW_CSVs	TS006	Population Density	N/A	population_density
RAW_CSVs	TS008	Sex	2	sex
RAW_CSVs	TS061	Travel to Work Method	12	transport_method
RAW_CSVs	WP002	Workplace Population	N/A	workplace_population_density
OTHER	N/A	Educational Establishments	N/A	educational-establishment
OTHER	N/A	Transport Access Nodes	N/A	transport-access-node
OTHER	N/A	Parks & Gardens	N/A	park-and-garden

3.4 Ordnance Survey Data

The below data was collected from the Ordnance Survey Open Data hub. All files were extracted as GeoPackage files and renamed as noted below for simplification.

- Built Up Areas[13] - File renamed to 'built_up_areas'

[8] https://www.ons.gov.uk/filters/3bc85bd1-d028-4e7d-8811-afb9c4f6518c/dimensions
[9] https://www.nomisweb.co.uk/sources/census_2021_wp
[10] https://www.planning.data.gov.uk/dataset/educational-establishment
[11] https://www.planning.data.gov.uk/dataset/transport-access-node
[12] https://www.planning.data.gov.uk/dataset/park-and-garden
[13] https://osdatahub.os.uk/data/downloads/open/BuiltUpAreas

- Greenspace Areas[14] - File renamed to 'greenspace_areas'
- Rivers[15] - File renamed to 'rivers'
- Roads[16]- File renamed to 'roads'

NOTE: The Roads dataset has not been provided in the submission as its compressed size is nearly 1GB. The code that reads on the Road data has been commented out, if the data has been downloaded by the user themselves this just needs to be uncommented.

3.5 Weather Data

Monthly temperature[17] and precipitation[18] were extracted from the Met Office Climate Data Portal in Shapefile formats. These are downloaded within a zip file which have been renamed to 'Monthly_Precipitation_Observations' and 'Monthly_Precipitation_Observations' respectively.

4 Process and Code Description

4.1 Methodology

The methodology can be run through the Jupyter Notebook 'FINAL_METHODOLGY.ipynb'. The only user input that is required is in the cell as per Fig 2. Following entering these inputs, the 'Restart Kernel' and 'Run All' buttons in Jupyter Notebook should be pressed to ensure accurate running of the whole method. This will automatically perform the Data Cleaning, Data Exploration and Model Development steps. The required inputs are:

'coarse_aggregation_level' – This is the coarse administrative boundary level to be used to fit the models. The code has been developed to allow for any of the boundaries noted above to be entered but only 'MSOA' has been tested, and due to submission size restrictions this is the only data that has been provided.

'finer_aggregation_level' – This is the administrative boundary level that the analysis is looking to disaggregate the coarse_aggregation_level to. The code has been developed to allow for any of the boundaries noted above to be entered but only 'LSOA' has been tested, and due to submission size restrictions this is the only data that has been provided.

'crime_type' – The type of theft crime to be analysed. This must be entered as a list. The code has been developed to allow for any combination of 'Bicycle theft', 'Burglary', 'Other theft', 'Robbery', 'Shoplifting', 'Theft from the person' and 'Vehicle crime' to be analysed, but only 'Burglary' was tested as part of this project.

'region' – The region of England to be analysed. The code has been developed to allow for any combination of 'North East', 'North West', 'Yorkshire and The Humber', 'East Midlands',

[14] https://osdatahub.os.uk/data/downloads/open/OpenGreenspace
[15] https://osdatahub.os.uk/data/downloads/open/OpenRivers
[16] https://osdatahub.os.uk/data/downloads/open/OpenRoads
[17] https://climatedataportal.metoffice.gov.uk/datasets/a58146dc208a4563887afdfc55e52b82_0
[18] https://climatedataportal.metoffice.gov.uk/datasets/d817269301044beeba61b707cc50d059_0

INPUTS

```python
# Specify the coarser aggregation level that the model will be built at
coarse_aggregation_level = 'MSOA'

# Specify the refined aggregation level that crime data will be predicted at
finer_aggregation_level = 'LSOA'

# Specify the Crime Type to be investigated
# Available crime types are: ['Bicycle theft', 'Burglary', 'Other theft', 'Robbery', 'Shoplifting', 'Theft from the person', 'Vehicle crime']
# NOTE: Only 'Burglary' has been tried and tested
crime_type = ['Burglary']

# Specify the region of England where crime estimations are to be refined
# Available regions are: ['North East', 'North West', 'Yorkshire and The Humber', 'East Midlands', 'West Midlands', 'East of England', 'London', 'South East', 'South West']
# NOTE: Only 3 scenarios have been tested: All regions, London only and ['East of England', 'London', 'South East', 'South West']
region = ['North East', 'North West', 'Yorkshire and The Humber', 'East Midlands', 'West Midlands', 'East of England', 'London', 'South East', 'South West']
```

✓ 0.0s Python

Fig 2. Jupyter Notebook cell with required user inputs to run the full methodology

'West Midlands', 'East of England', 'London', 'South East' and 'South West' to be analysed but only the following tests have been conducted:

- 'London'
- 'East of England', 'London', 'South East', 'South West'
- All Regions

NOTE: As discussed in Section 3, data has not been provided to run All Regions due to size restrictions in the submission.

4.2 Data Cleaning

The functionality behind the data cleaning has been developed within the 'DATA_CLEANING_FUNCTIONS.py' script. The functions from this script need to be imported into the 'FINAL_METHODOLGY.ipynb'. All functions developed in this script include descriptions of their purpose, the parameters required, the required format of those parameters and any references used for the development of the function. This code assumes that all data is stored in the manner in which it has been provided in the submission. The functions within this script are:

'boundary_data_cleaning' - This function is designed to produce datasets of UK administrative area boundaries to the same desired regions.

'census_data_cleaning' - This purpose of this function is to convert each of the individual census datasets into an single dataset for the selected aggregation level.

'crime_data_cleaning' - The purpose of this is to combine the crime data extracts from the Data.Police.UK website, which are available per Police Force per month, into a workable dataset based on the requested crime type. The function is also designed to check for and remove duplicate records. The location point and occurrence year of each incident is also derived.

'crime_data_aggregation' - The purpose of this function is to aggregate the incident level crime data produce from the 'crime_data_cleaning' to the requested UK administrative boundary level.

```
CREATING CRIME DATASET: Reading Incident Level CSVs: 100%|██████████| 1056/1056 [02:04<00:00,  8.49it/s]
MSOA BOUNDARIES DATA CLEANED
MSOA CENSUS DATA CLEANED
MSOA CRIME DATA CLEANED
MSOA WEATHER DATA CLEANED
MSOA OS DATA CLEANED
MSOA CLEANING COMPLETED
LSOA BOUNDARIES DATA CLEANED
LSOA CENSUS DATA CLEANED
LSOA CRIME DATA CLEANED
LSOA WEATHER DATA CLEANED
LSOA OS DATA CLEANED
LSOA CLEANING COMPLETED
```

Fig 3. Jupyter Notebook cell showing completion of each cleaning step for each dataset

'weather_data_cleaning' - This function is designed to aggregate weather data to the requested administrative level. The function operates by reading in datasets that include monthly observations over the period from 1991 to 2020 and calculates an annual average per area.

'os_data_cleaning' - This function is designed to aggregate features available from the Ordnance Survey Data Hub to the requested administrative boundary level. The features available to be aggregated using this function are Built Up Areas, Greenspace Areas, Rivers and Roads. Built Up Areas and Greenspace Areas are converted to the proportion of the overall area, e.g. 50% of administrative area is defined as a built up area. Rivers and Roads are converted to a density, e.g. 0.3 km per km2 are covered by road.

'create_working_datasets' - The purpose of this function is to create the training and test datasets that will be used to fit and test the model respectively, by taking the outputs from the other cleaning functions for the respective aggregation levels.

Depending on the regions selected in the analysis, the data cleaning process can take a considerable amount of time. The detail shown in Fig 3 is printed in the output to inform the user of what stage the cleaning process is at.

These data cleaning functions include the steps for features engineering to create the dataset that consists of the independent and dependent variables in Table 4. Note that there are other features within the working dataset such as 'geometry', 'CENTROID' and 'COUNT' which are used at other stages of the model, and are removed from the data fitting the model at the scaling stage.

Table 4. Full list and description of independent and dependent variables

Variable	Variable Type	Description
age_6_groups_1	Independent	Aged 15 years and under
age_6_groups_2	Independent	Aged 16 to 24 years
age_6_groups_3	Independent	Aged 25 to 34 years
age_6_groups_4	Independent	Aged 35 to 49 years
age_6_groups_5	Independent	Aged 50 to 64 years
age_6_groups_6	Independent	Aged 65 years and over
population	Independent	Population
job_class_type_1	Independent	L1, L2 and L3: Higher managerial, administrative and professional occupations
job_class_type_2	Independent	L4, L5 and L6: Lower managerial, administrative and professional occupations
job_class_type_3	Independent	L7: Intermediate occupations
job_class_type_4	Independent	L8 and L9: Small employers and own account workers

job_class_type_5	Independent	L10 and L11: Lower supervisory and technical occupations
job_class_type_6	Independent	L12: Semi-routine occupations
job_class_type_7	Independent	L13: Routine occupations
job_class_type_8	Independent	L14.1 and L14.2: Never worked and long-term unemployed
job_class_type_9	Independent	L15: Full-time students
area_uk	Independent	Country of birth is UK
area_africa	Independent	Country of birth is in Africa
area_asia	Independent	Country of birth is in Middle East or Asia
area_america	Independent	Country of birth is in The Americas and the Caribbean
area_other	Independent	Country of birth is in Antarctica and Oceania (including Australasia) and Other
area_british_overseas	Independent	Country of birth is British Overseas
area_europe	Independent	Country of birth is in Europe other than UK
deprivation_0_dim	Independent	Household is deprived in one dimension
deprivation_1_dim	Independent	Household is deprived in two dimensions
deprivation_2_dim	Independent	Household is deprived in three dimensions
deprivation_3_dim	Independent	Household is deprived in four dimensions
education_level_0	Independent	No qualifications
education_level_1	Independent	Level 1 and entry level qualifications
education_level_2	Independent	Level 2 qualifications
education_level_3	Independent	Apprenticeship
education_level_4	Independent	Level 3 qualifications
education_level_5	Independent	Level 4 qualifications
education_level_6	Independent	Other, e.g. vocational or work-related qualifications
general_health_1	Independent	Very good health
general_health_2	Independent	Good health
general_health_3	Independent	Fair health
general_health_4	Independent	Bad health
general_health_5	Independent	Very bad health
house_class_1	Independent	Detached
house_class_2	Independent	Semi-detached
house_class_3	Independent	Terraced
house_class_4	Independent	In a purpose-built block of flats or tenement
house_class_5	Independent	Part of a converted or shared house, including bedsits
house_class_6	Independent	Part of another converted building, for example, former school, church or warehouse
house_class_7	Independent	In a commercial building, for example, in an office building, hotel or over a shop
house_class_8	Independent	A caravan or other mobile or temporary structure
house_tenure_1	Independent	Owned: Owns outright
house_tenure_2	Independent	Owned: Owns with a mortgage or loan
house_tenure_3	Independent	Shared ownership: Shared ownership
house_tenure_4	Independent	Social rented: Rents from council or Local Authority
house_tenure_5	Independent	Social rented: Other social rented
house_tenure_6	Independent	Private rented: Private landlord or letting agency
house_tenure_7	Independent	Private rented: Other private rented
occupation_1	Independent	Managers, directors and senior officials
occupation_2	Independent	Professional occupations
occupation_3	Independent	Associate professional and technical occupations
occupation_4	Independent	Administrative and secretarial occupations
occupation_5	Independent	Skilled trades occupations
occupation_6	Independent	Caring, leisure and other service occupations
occupation_7	Independent	Sales and customer service occupations
occupation_8	Independent	Process, plant and machine operatives
occupation_9	Independent	Elementary occupations
sex_Female	Independent	Females
sex_Male	Independent	Males
population_density	Independent	Population Density
workplace_population_density	Independent	Workplace Population Density
transport_method_1	Independent	Work mainly at or from home
transport_method_2	Independent	Travel to work by underground, metro, light rail, tram
transport_method_3	Independent	Travel to work by train
transport_method_4	Independent	Travel to work by bus, minibus or coach

transport_method_5	Independent	Travel to work by taxi
transport_method_6	Independent	Travel to work by motorcycle, scooter or moped
transport_method_7	Independent	Travel to work by driving a car or van
transport_method_8	Independent	Travel to work by passenger in a car or van
transport_method_9	Independent	Travel to work by bicycle
transport_method_10	Independent	Travel to work by on foot
transport_method_11	Independent	Other method of travel to work
unemployed	Independent	Unemployed
educational_establishment_count_per_km2	Independent	Educational establishments per km^2
park_and_garden_count_per_km2	Independent	Parks and gardens per km^2
transport_access_node_count_per_km2	Independent	Transport access nodes per km^2
average_temperature	Independent	Average annual temperature
average_precipitation	Independent	Average annual precipitation
built_up_areas_proportion	Independent	Proportion of area defined as built up
greenspace_areas_proportion	Independent	Proportion of area defined as greenspace
road_density	Independent	Road density
river_density	Independent	River density
LOG_CRIME	Dependent	Log of number of crimes per 1,000 people. Used for regression evaluation
RISK_GROUP	Dependent	Group from 1, lowest risk, to 5, highest risk, based on LOG_CRIME. Used for classification evaluation

4.3 Data Pre-Processing

The functionality behind the data pre-processing has been developed within the 'DATA_PRE_PROCESSING_FUNCTIONS.py' script. The functions from this script need to be imported into the 'FINAL_METHODOLGY.ipynb'. All functions developed in this script include descriptions of their purpose, the parameters required, the required format of those parameters and any references used for the development of the function. The functions within this script are:

'spatial_data_split' - This function is designed to split the dataset into a training and validation dataset based on the spatial autocorrelation cluster, i.e. to keep the same proportion of spatial autocorrelation clusters in each dataset. Developed using logic from Rey, Arribas-Bel and Wolf (2023).

'custom_scale_data' - This is a custom function designed to scale the independent features using RobustScaler from sklearn, and to produce an 'xy' array consisting of the centroids of the areas.

'mi_vif_feature_selection' - This function is designed to perform feature selection based on the mutual information between a feature and the dependent variable. The multicollinearity in the final feature subset is addressed by iteratively adding features based on their mutual information, and discarding if any of the features in the resulting subset breach the variance inflation factor threshold. Logic based on research by Cheng *et al.* (2022).

'hotspot_identification' - The purpose of this function is to clssify areas as hotspots or coldspots. This is done using the 'StdMean' logic behind a choropleth chart using 'mapclassify'.

- There are 5 classes produced:
  1) Coldspot - More than 2 standard deviations below the mean
  2) Low-Risk Area - Between 1 and 2 standard deviations below the mean
  3) Average - Within 1 standard deviation of the mean
  4) High-Risk Area - Between 1 and 2 standard deviations above the mean

5) Hotspot - More than 2 standard deviations above the mean

4.4 Data Exploration

The functionality behind the data exploration has been developed within the 'DATA_EXPLORATION_FUNCTIONS.py' script. The functions from this script need to be imported into the 'FINAL_METHODOLGY.ipynb'. All functions developed in this script include descriptions of their purpose, the parameters required, the required format of those parameters and any references used for the development of the function. The functions within this script are:

'custom_choropleth_chart' – Custom choropleth chart function based on logic from Rey, Arribas-Bel and Wolf (2023).

'custom_shap_choropleth_chart' - Custom choropleth chart function specifically designed for plotting the calculated SHAP values. It centres the colour scheme around 0 to keep all SHAP illustrations consistent, by using normalisation. Again uses logic from Rey, Arribas-Bel and Wolf (2023) as well as official Mathplotlib documentation for normalisation process.

'spatial_autocorrelation' - This function is designed to calculate and display the spatial autocorrelation distribution of a selected feature.

'custom_correlation_coefficient_matrix' - This function is created to split the features into those that have a strong correlation, a moderate correlation and a weak correlation with the dependent variable before creating a correlation coefficient matrix for each group. The purpose is due to the high number of independent variables making it difficult to view a correlation matrix with all variables.

'one_way_glm_feature_eval' - This function is designed to estimate the importance of each independent variable by fitting a GLM that only includes the variable in question. Only used for exploration purposes. General logic referenced from Massaron (2016).

'custom_crime_line_chart' - Simple custom function that uses the standard Mathplotlib line chart to plot the crime counts per month.

4.5 Model Development

The functionality behind the development and evaluation of the models has been developed within the 'MODEL_FUNCTIONS.py' script. The functions from this script need to be imported into the 'FINAL_METHODOLGY.ipynb'. All functions developed in this script include descriptions of their purpose, the parameters required, the required format of those parameters and any references used for the development of the function. The general structure of the code was developed using some framework from research by Sun *et al.* (2024). The functions within this script are:

'nn_gaussian_distance' - The purpose of the function is to identify a specified number closest points from a particular point, and to spatially weight the distances using a Gaussian kernel. An adaptive bandwidth is used, determined by the number of closest points rather than a fixed distance. This is because the areas in the research are administrative boundaries of different size and shape.

'fit_local_model' - Function designed to fit a local model for the selected training point. It includes the Gaussian kernel spatial weighting of the selected nearest point to be included in the fitting of the model, and the feature selection process for the local model.

'weighted_shap_values' - The purpose of this function is to calculate the local SHAP values for each test point for each feature to determine the feature importance behind the prediction. The SHAP values are weighted based on the relative importance of each model in the prediction, based on Gaussian kernel function.

'classification_evaluation' - The purpose of this function is to evaluate the classification of the RISK GROUPs via a Confusion Matrix.

'model_predictions' - The purpose of this function is to predict the dependent variable for each of the data points in test dataset using each of the input models (global and local). A final prediction for each test point is generated by weighting the predictions from the specified nearest models using Gaussian kernel. Utilises the 'weighted_shap_values' function to calculate SHAP values if requested.

'prediction_reproportion' - The purpose of this function is to re-proportion the predictions at the refined resolution so that the overall count matches the actual counts at the aggregated resolution. The function includes the ability to perform partial pooling, which adjusts the predictions to account for the variance in the aggregated counts prior to re-proportioning.

'geographically_weighted_model' - The purpose of this function to fit, predict and evaluate geographical data. It is designed to fit a global model using all data points and local models for each data point using a specified nearest number of training points, weighted using Gaussian kernel distances. It is specifically designed and tested to fit models based on crime data at an MSOA aggregation level, and predict crime at a LSOA aggregation level. The functionality includes the ability to produce weighted SHAP values for explainability. Any tree-based model can be input into the function.

The Model Development section in the 'FINAL_METHODOLGY.ipynb' notebook has been set up with the correct inputs to achieve the results for each of the methods described in the research paper (including the Benchmark method). Depending on the selections made the training and evaluation time can vary significantly. Fig 4 shows a snapshot of the descriptions that are printed to advise the user what stage of the process it is currently at. These inputs can be amended to achieve desired variations of the framework. Refer to 'geographically_weighted_model' description for details on which parameters can be adjusted.

4.6 Outputs

The code within the 'FINAL_METHODOLGY.ipynb' notebook will automatically generate all of the outputs within the cell outputs. All charts are generated through the custom functions. The workings behind the predictions as described in the report are output in separate daataframes

The final outputs from the methodology will produce a choropleth chart showing the actual risk groups at the coarse aggregation level (MSOA) and the finer resolution aggregation level (LSOA) as well as the final predicted risk groups for the finer resolution aggregation level.

```
Fitting Local Models: 100%|██████████| 1002/1002 [09:41<00:00,  1.72it/s]
Predicting Training Data: 100%|██████████| 1003/1003 [00:09<00:00, 104.40it/s]
Weighting Training Predictions: 100%|██████████| 1002/1002 [00:03<00:00, 269.52it/s]
TRAINING EVALUATION METRICS - R2: 92.2, MSE: 0.02, MAE: 0.1
              precision    recall  f1-score   support

           1       0.90      0.82      0.86        11
           2       0.82      0.79      0.80       131
           3       0.93      0.95      0.94       717
           4       0.83      0.71      0.77       118
           5       0.71      0.88      0.79        25

    accuracy                           0.90      1002
   macro avg       0.84      0.83      0.83      1002
weighted avg       0.90      0.90      0.90      1002

Predicting Test Data: 100%|██████████| 1003/1003 [00:23<00:00, 42.09it/s]
Weighting Test Predictions: 100%|██████████| 4994/4994 [00:17<00:00, 291.20it/s]
Creating SHAP Explainers: 100%|██████████| 1003/1003 [01:20<00:00, 12.44it/s]
Calculating SHAP Values: 100%|██████████| 1003/1003 [03:28<00:00,  4.80it/s]
Weighting Test SHAP Values: 100%|██████████| 4994/4994 [54:59<00:00,  1.51it/s]
```

Fig 4. Jupyter Notebook output cell showing completion of each step in the model

SHAP charts will also be produced for the features with the 9 highest average absolute SHAP values. These charts are set up to be produced from Method 5. If the 'shap_values' parameter for this model has been set to 'False' no charts will be output.

References

Cheng, J., Sun, J., Yao, K., Xu, M. and Cao, Y. (2022), 'A variable selection method based on mutual information and variance inflation factor', *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 268, p.120652. https://doi.org/10.1016/j.saa.2021.120652.

Massaron, L. (2016). *Regression Analysis with Python: Learn the Art of Regression Analysis with Python*. Birmingham; Bombay: Packt Publishing. C.

Matplotlib.org. (2025). *matplotlib.colors.TwoSlopeNorm — Matplotlib 3.10.7 documentation*. [online] Available at: https://matplotlib.org/stable/api/_as_gen/matplotlib.colors.TwoSlopeNorm.html#matplotlib.colors.TwoSlopeNorm [Accessed 30 Nov. 2025]

Rey, S.J., Arribas-Bel, D. and Wolf, L.J. (2023). *Geographic Data Science with Python*. CRC Press.

Sun, K., Zhou, R., Kim, J. and Hu, Y. (2024) 'PyGRF: An Improved Python Geographical Random Forest Model and Case Studies in Public Health and Natural Disasters', *Transactions in GIS*, 28(7), pp.2476–2491. https://doi.org/10.1111/tgis.13248