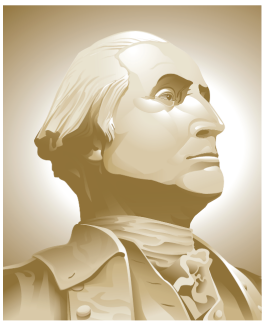


THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Data Science Program

Capstone Report - Spring 2022

Machine Learning Analysis on Contextual and Covariate Features in Satellite Image of Lagos, Nigeria

Jake Lieberfarb
Bradley Reardon

supervised by
Amir Jafari

Abstract

According to IDEAMAPS [1], at current, there is no systematic and scalable approach for mapping 'Deprived' areas throughout cities. With a map outlining which areas of a city are deprived, governments can more efficiently work towards building up the deprived areas without spending countless hours manually searching cities by foot. The approach this project took toward solving this issue is implementing deep learning, classical machine learning techniques, and statistical analysis on open-source data and free low resolution satellite imagery to classify and map areas as deprived or not. This approach proved to be an efficient and low-cost method for solving the issue at hand.

Table of Contents

1 Introduction

According to IDEAMAPS [1], at current, there is no systematic and scalable approach for mapping ‘Deprived’ areas throughout cities. Doing so is increasingly more difficult in low- and middle-income countries that do not have the funds, manpower, and infrastructure for efficient mapping.

A recent methodology for mapping the ‘Deprived’ areas involved satellite imagery and the use of machine learning algorithms to detect and classify areas as ‘Deprived’ or ‘Built-up’. Because high quality satellite imagery is costly and difficult to access, the use of deep learning neural networks (particularly convolutional neural networks) often struggles with detecting deprived areas using low-resolution imagery. One potential solution to this problem while still implementing the use of machine learning algorithms is to apply traditional machine learning classification techniques on contextual and covariate feature data.

This project aimed to apply various deep learning and traditional machine learning classification techniques on low resolution and free satellite imagery as well as on calculated distance contextual and covariate features to detect deprived areas on a 10m^2 level.
The two core streams of the projects to detect deprived areas were as follows:

1 - Processing and usage of raw satellite images
2 - Utilizing covariate and contextual data

This report focused on the second stream to use labeled contextual and covariate features to train traditional classification techniques and classify deprived areas. The contextual and covariate feature data was mainly provided by open-source sources (OpenStreetMap and government data portals) and were computed using GIS software while the labeled satellite imagery was mainly provided by Ideamapsnetwork as ground observation work where deprived areas were mapped manually. In future efforts and cities, this will be conducted by teams local to said cities. The contextual features proved to be inefficient in providing enough information to accurately detect and classify deprived areas. However, the covariate features proved to be very useful in detecting which areas were considered ‘Deprived’.

2 Data Source

The data that was used to train and test the traditional machine learning models. There were two distinct datasets: contextual features and covariate features. Each consisting of labeled data comprised of three classes:

- Built-up (class 0)
- Deprived (class 1)
- Non-built-up (class 2)

2.1 Contextual Features

Contextual features were defined as the statistical quantification of edge patterns, pixel groups, gaps, textures, and the raw spectral signatures calculated over groups of pixels or

neighborhoods. [2]. These features can identify patterns and homogeneity in spatial configurations that go beyond spectral patterns or color intensities [2]. The contextual features dataset contains 144 distinct features of feature types: Fourier, Gabor, HOG, lacunarity, LBPM, LSR, mean, , normalized difference vegetation index (NDVI), ORB, PanTex, and SFS. The features were computed on a moving window size of 30-meters, 50-meters, and 70-meters which allows for capturing patterns at different scales, and zonal statistics – mean, sum, and standard deviation – were calculated on each contextual feature output [2].

The following quote from *Evaluating the Ability to Use Contextual Features Derived from Multi-Scale Satellite Imagery to Map Spatial Patterns of Urban Attributes and Population Distributions* [2] describes the contextual feature types in detail:

" Fourier Transform. Fourier transform captures the frequency of patterns across an image. Any signal can be represented as a series of sinusoidal signals ; thus, an image can be decomposed into sine and cosine waves with various amplitudes and frequencies . The Fourier transform consists of magnitude and phase parts, with the former usually displayed as the output image (power spectrum). In these magnitude outputs, low-frequency features, such as water, are located closer towards the origin (center), with increasing frequency farther from the origin . A radial profile can be derived from a power spectrum, within which pixel frequencies can be summarized. Fourier produces two outputs: mean and variance.

Gabor. Gabor is a linear filter used for edge detection. Multiple filters consisting of strips are created by a sinusoidally modulated Gaussian function, forming the filter bank. The size, shape, and orientation of the filters can be set, and the various orientations enable extraction of features with those associated orientations. A Gabor wavelet transformation is outputted. There are 16 Gabor outputs: mean, variance, and 14 individual filters that examine different angles. Histogram of Oriented Gradients. HOG identifies the orientation and magnitude of shades, distinguishing settlement and non-settlement classes. Gradient magnitudes in both the x and y directions are calculated for each pixel and combined to obtain the magnitude and direction of the gradient. The image is divided into subregions (cells), and within each, the gradient direction bins the pixels by angles (1◦–180◦). The magnitude of each pixel is distributed to its associated bin, with the magnitude value split among two bins if the gradient direction falls between two. The aggregated magnitudes in each bin form a histogram (vector) for the cell.

Next, four cells (and their four histograms) are concatenated into a block and normalized. All block vectors are combined to form the final HOG vector, and statistics can be extracted. The five statistical outputs are the maximum, mean, variance, skew, and kurtosis.
Lacunarity. Lacunarity measures the homogeneity of the landscape via the spatial distribution of gap sizes. For heterogeneous images, all gap sizes are not the same; thus, the image is not translationally invariant, and lacunarity is high. For instance, in urban areas, there are gaps between buildings; in high density areas, there tend to be less gaps. Variation in gap sizes is scale dependent.

One way to calculate lacunarity involves a moving window in which the number of holes is calculated. First, an intensity surface, where the plane is the image and the z-axis (height) is the intensity (value) of the pixels, is created. A moving window of a set size is centered over one pixel, with a smaller gliding box placed in the upper left corner. If necessary, multiple boxes are stacked so all the pixel intensities fall within. The relative height is calculated using the minimum and maximum pixel values (or the boxes in which they fall) within the column. As the gliding box moves across the image window, all the relative heights are summed, and a formula

is used to calculate lacunarity for that center pixel. The window repeats the process across the image. Only one lacunarity value is calculated.

Line Support Regions. LSR extracts straight lines from imagery, which can determine the area and spatial configuration of settled areas. Gradient orientations on an image are first calculated and used to group pixels into LSRs with similar gradient orientations. The groups that do not have enough support (pixels appropriated to a region, as described in Burns et al.) are removed. A plane fit to the pixel intensities in each line support region using a least squares fit and a horizontal plane of average pixel intensities, both weighted by local gradient magnitude, are created. A line is extracted where the two planes intersect. The line's length, width, contrast (intensity change over Remote Sens. 2021, 13, 3962 10 of 27 the line), steepness (slope of intensity change), and straightness can subsequently be obtained. LSR produces three outputs: line length, line mean, and line contrast.

Local Binary Pattern. LBPM assesses the homogeneity of an image, detecting bright and dark spots, flat areas, and edges. After the radius and number of neighbors are specified, the value of a center pixel is compared with those of its surrounding neighbors. If the center pixel value is smaller or equal, the neighbor is given a value of 1; otherwise, the value is 0. The values around the center pixel are taken sequentially (forming a binary string) and inputted into an equation to obtain the LBPM code for the center pixel. Patterns with more than two 0-1 or 1-0 switches are not uniform, with two or less considered uniform. A histogram is built with separate bins for each uniform pattern and one bin for all non-uniform patterns; this is based on Ojala et al.'s observation that certain uniform patterns appear more frequently in textures. Five statistical outputs of LBPM are produced: maximum, mean, variance, skew, and kurtosis.

Mean. The mean of the image is calculated using inverse distance weighting (IDW). IDW is an interpolation method where the influence of a point on an unknown point is inversely related with distance and dependent on the specified power setting, which controls the rate at which the influence of points decreases with increasing distance. For SpFeas, pixels near the center of a frame are given higher weights. In addition to mean, the variance of the pixels within the scale used is also calculated.

Normalized Difference Vegetation Index. NDVI assesses vegetation by incorporating a pixel's value in the NIR and red regions. High values (towards 1) reflect a higher density of green vegetation, and low values (towards -1) reflect a lower density. NDVI values are generally lower in and negatively correlated with built-up areas due to sparser vegetation. Both the mean and variance of NDVI are calculated for each scale.

ORB. A feature-based matching method introduced by Rublee et al., ORB combines the Features from Accelerated Segment Test (FAST)—a feature detector—and Binary Robust Independent Elementary Features (BRIEF)—a feature descriptor—approaches.

The FAST algorithm is used to identify key points at each level in a scale pyramid of the image, and the Harris corner measure orders the key points and rejects edges picked up by FAS. Intensity centroid is used to assign an orientation to the corner. BRIEF selects a random pair of pixels around a key point, compares their intensity values, and assigns them binary values. The orientation from the intensity centroid is used to steer BRIEF towards this orientation, as BRIEF is not invariant to rotation. A greedy algorithm takes all the pairs and creates a subset (usually 256) of uncorrelated pairs, forming a 256-bit feature descriptor output (rotated BRIEF or rBRIEF). Five statistical outputs from ORB are produced: maximum, mean, variance, skew, and kurtosis.

PanTex. PanTex extracts built-up areas from panchromatic imagery using the GLCM approach. The textural contrast is calculated in all directions within a window around a pixel. The minimum value is taken, and the output with all the minimum values is the PanTex index. For urban areas, this minimum value would be consistently high. Pesaresi et al., used minimum values over average values, reasoning those averages produce an edge effect that could overestimate built-up areas. PanTex produces one output, which is the minimum contrast.

Structural Feature Sets. SFS extracts information on direction-lines. Lines from the center pixel are created in all directions. For a direction-line, a pixel is compared with the center pixel to determine whether it is considered homogenous. If it is, it is added to the direction line; the line keeps extending until a pixel is not considered homogenous based on set threshold levels or until the line reaches a set maximum length. This is repeated for all line directions. A histogram is built from the lines, and statistics can be extracted. SFS produces six outputs: maximum line length, minimum line length, mean, w-mean (weighted mean), standard deviation, and maximum ratio of orthogonal angles." [2]

2.2 Covariate Features

The covariate features dataset consists of 61 distinct features from the following domains: Contamination, Facilities & services, Housing (HH) Housing (HO), Infrastructure, Physical hazards & assets, Population counts, SFS (HH), Social hazards & assets, Unplanned urbanization. These feature values were sourced from various open-source and government sources such as humdata.org, worldpop.org, geopode.org, spatialdatya.dhsprogram.com, and more. Please see the appendix for a table that lists each covariate feature, a short description of the feature, the data type, the domain the feature belongs to, and the link of the original data.

Covariate Feature Types	Count
Facilities & Services	8
Housing (HO)	1
Infrastructure	5
Physical Hazards & Assets	23
Population Counts	2
SES (HH)	14
Social Hazards & Assets	5
Unplanned Urbanization	3

(Figure 1: Covariate Feature Type Counts)

3 Problem Statement

There were two questions the researchers hoped to address in this study:

1. Can feature importance methods derive any contextual or covariate features that are useful in identifying 'Deprive' and 'Built-up' areas?

2. Through the use of classical machine learning models and statistical analysis, are contextual and/or covariate features useful in identifying ‘Deprived’ and ‘Built-up’ areas.

4 Related Work

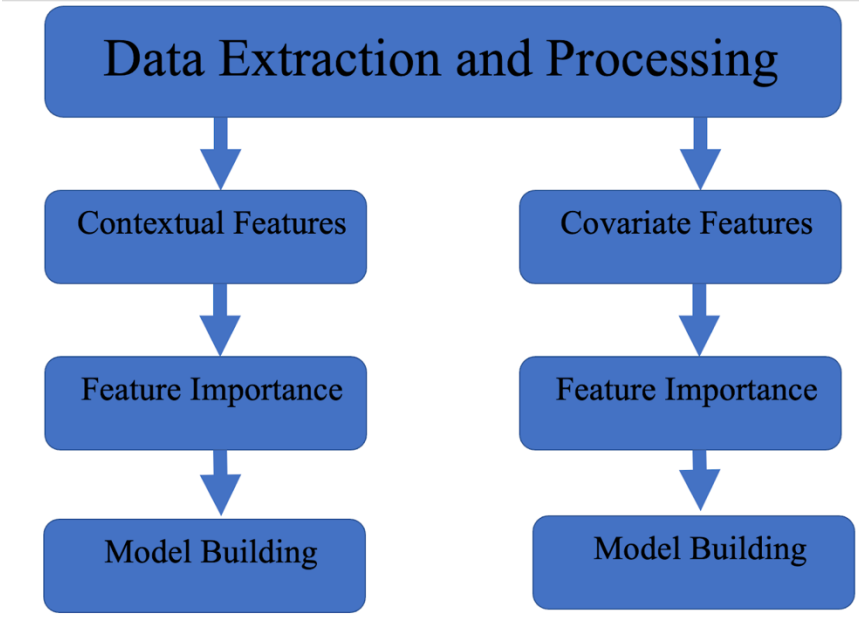
Chao et al [2] found that analyzing contextual feature data from Sentinel-2 (10 m pixels) images in Accra, Belize, Ghana and Sri Lanka was useful in modeling population density and human modified landscape. The model these researchers developed yielded a coefficient of determination of 85% at very high spatial resolutions (<2m). With low resolution imaging (10 m) the model had an R^2 of 84%.

Saarela and Jauhiainen [3] utilized logistic regression with L1 penalization and non-linear (random forest) on the breast cancer data from UCI Archives and a running injury dataset. The goal of this research project was to see if combining feature importance techniques could provide more reliable results. The UCI breast cancer dataset contained 31 feature readings for breast masses identified (*smoothness, radius, symmetry*, …) as either malignant or benign. The different feature importance methods did have some overlap of features. The nine most important features for both methods had three overlapping values. The running injury dataset had a total of 85 features (*run level, left hip abductor, knee flexion peak of both legs*…) that described if someone was either injured or not. Random forest detected 22 important feature and logistic detected 61 important features. There was an overlap of only 13 features. This study demonstrated that introducing different feature importance methods can add to the robustness of the analysis. However, there may be limited overlap between different methods.

5 Solution and Methodology

5.1 Overview of Section

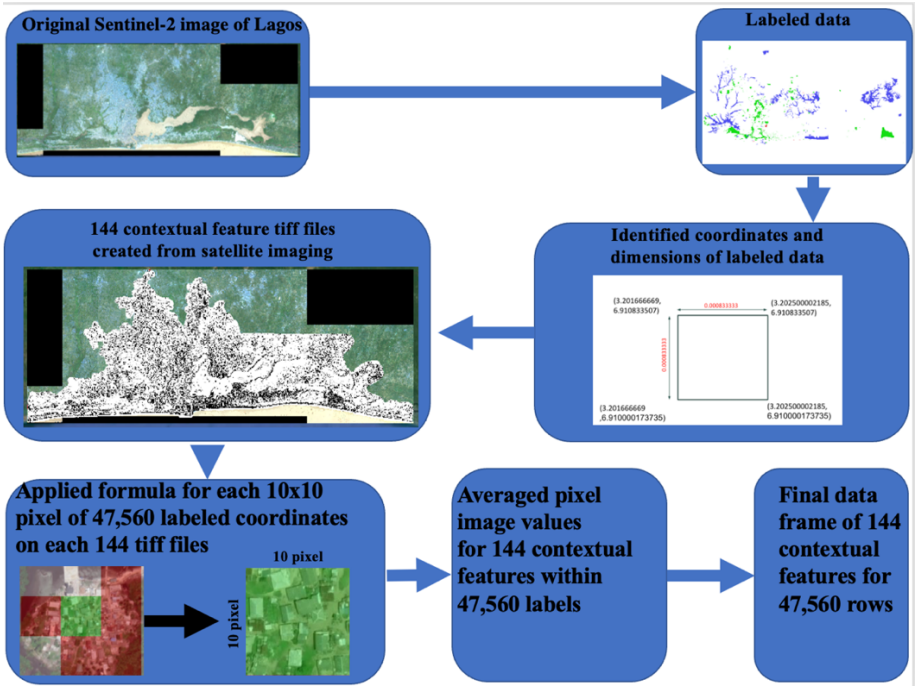
The solution and methodology sections were split up in two phases: contextual features and covariate features. The two sections underwent the same pipeline after the contextual and covariate features were processed.



(Figure 2: Overview of Modeling)

The data was processing steps for the contextual features and covariate had a subtle difference in the steps to form their respective final data frame.

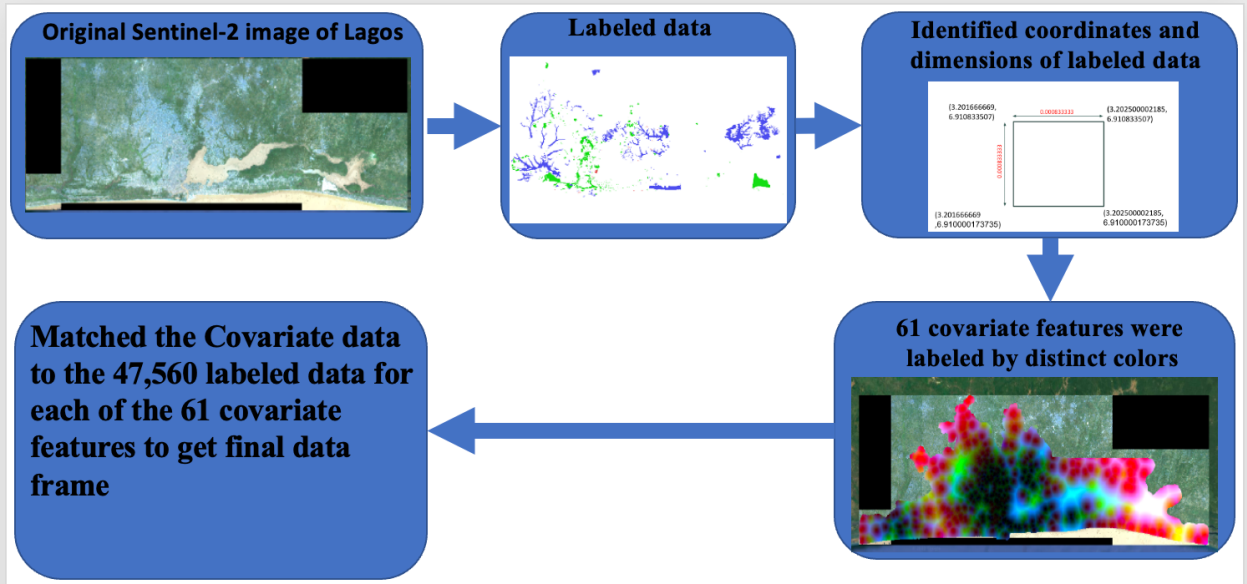
5.2.1 Data Extraction and Processing on Contextual Features



(Figure 3: Contextual Feature extraction methodology)

Through satellite imaging analysis, 144 statistical measurements for each pixel of Lagos, Nigeria were calculated. The original labeled data was stored in *lag_training_2021.tiff*. The labels of 'Not-Built-Up', 'Built-Up', and 'Deprived' were assigned to 10x10 coordinate grid sections (comprising 100 individual pixels). These measurements were originally attached to the latitude and longitude of the upper left pixel of each coordinate grid section. Each of these labels were 10x10 pixels with a length of 0.00008333. The researchers calculated the center pixel values for each pixel within each label section and matched it to the corresponding features from each contextual feature tiff file. Next, the contextual feature coordinates for each pixel were merged to the expanded coordinates of the larger label coordinate section (10x10 grid) and averaged to have a final data frame of 144 features for each 47,560 labels. The researchers then investigated the contextual features to assess if they were useful in identifying the three classes of 'Not-Built-Up', 'Built-Up', and 'Deprived'. The analysis was initially conducted on all three classes, and then *Not-Built-Up* was removed and the same analysis was conducted on just the *Built-Up,* and *Deprived* classes.

5.2.2 Data Extraction and Processing on Covariate Features



(Figure 4: Methodology for processing Covariate Features)

The methodology for processing the covariate features was done by matching the labeled coordinates with the 61 covariate features found within one file, *lag_covariates_compilation.tiff*. The 61 distinct colors or 'bands' found within the tiff file corresponded to the different covariate features that required mapping. Once the covariate values were correctly mapped, the final data frame had 61 columns and 47,560 rows.

After the contextual and covariate features were correctly matched to the labeled data, five different feature importance methodologies were introduced to rank the features based on their ability to identify the labeled features.

5.3 *Feature Standardization*

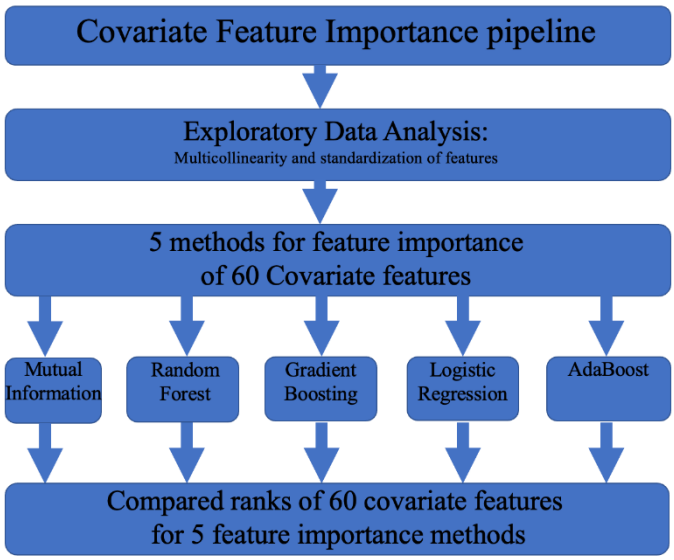
As part of preparing the contextual and covariate data for analysis. The features were standardized using *StandardScaler[4]*.

$$z = \frac{x - u}{s}$$

(Figure 5: Standardization Formula Used on Contextual and Covariate Features)

5.4 Feature Importance Methods

Five feature importance methods were implemented on the contextual and covariate datasets: mutual information extraction, random forest feature importance, logistic regression coefficient values, gradient boosting feature importance, and adaboost feature importance.



(Figure 6: Pipeline for Feature Importance on Contextual and Covariate Datasets)

5.4.1 Mutual Information Extraction

To implement mutual information feature importance on the contextual and covariate features, the tools of *SelectKBest[5]* and *mutual_info_classif* [6]. The SelectKBest method ranks the total number of features 'k' inputted into the function. For contextual and covariate features, k was 144 and 60, respectively. The method *mutual_info_classif* was introduced for feature importance as it measures the dependency of each individual feature with the target value. The function depends on nonparametric methods related to entropy estimation from k-nearest neighbors distances[7]. Mutual information is closely related to entropy and provides results from the range of zero to 1[8].

$$H(X) = -\sum_{x_i \in X} P(X = x_i) * \log\left(P(X = x_i)\right)$$

(Figure 7: Entropy Calculation)

5.4.2 Random Forest Feature Importance

For Random Forest feature importance. The function *RandomForestClassifier*[9] was introduced to rank importance to each contextual and covariate feature through the method '*feature_importances_'*. Feature importance was calculated as the mean and standard deviation of accumulation of the impurity decrease within each decision tree of the random forest[10].

5.4.3 Logistic Regression Feature Importance

To identify feature importance in logistic regression, full models were constructed for the contextual and covariate features. The function *LogisticRegression* [11] was utilized for model building. Next, the coefficient values (*B*) for the 144 contextual features and 60 covariate features were extracted and ordered in descending order.

$$P(Y = 1) = \frac{1}{1 + e^{-(B_0 + B_1 + \cdots + B_{144})}}$$

$$P(Y = 1) = \frac{1}{1 + e^{-(B_0 + B_1 + \cdots + B_{61})}}$$

(Figure 8: Logistic Formula for Contextual and Covariate Models)

5.4.4 Gradient Boosting Feature Importance

The model *GradientBoostingClassifier* was introduced to extract feature importance on the contextual and covariate dataset. The attribute *feature_importances_* was utilized to rank the features based on the total reduction of the criterion within each feature (Gini importance) [12].

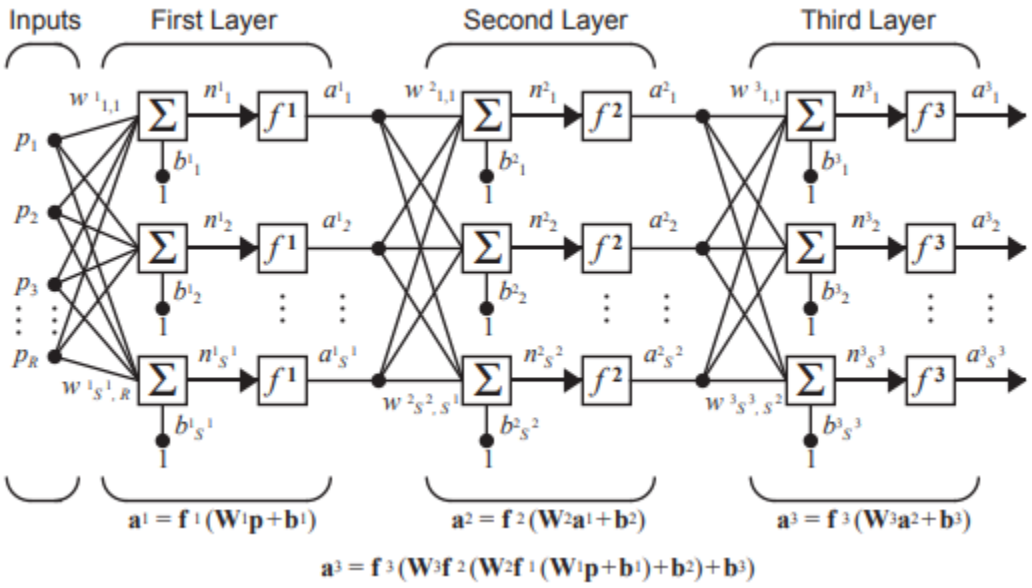
5.4.5 Adaptive Boost (AdaBoost) Feature Importance

The fifth feature importance method utilized in this analysis was through *AdaBoostClassifier*. This attribute *feature_importances_* was implemented to identify the features that were doing the best at predicting the target variable. This implementation was done through calculating the Gini importance of each feature [13].

5.5 Models

5.5.1 Multi-Layer Perceptron

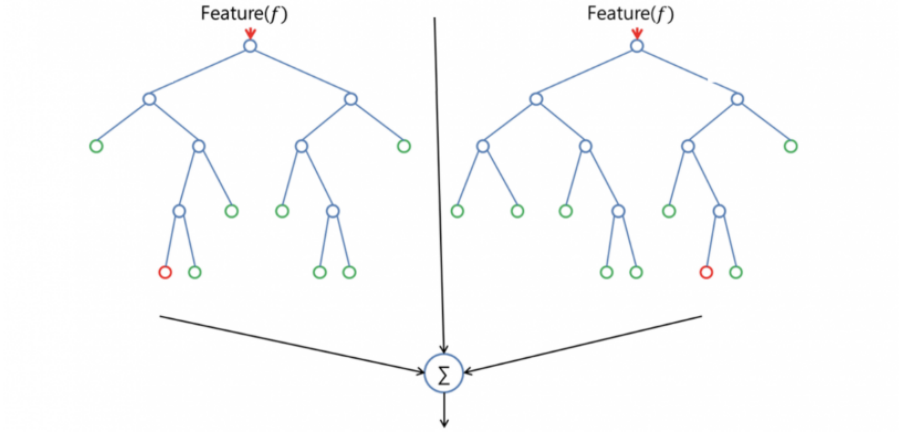
"A multilayer perceptron is a neural network connecting multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. Each node, apart from the input nodes, has a nonlinear activation function. An MLP uses backpropagation as a supervised learning technique. Since there are multiple layers of neurons, MLP is a deep learning technique." [15]



(Figure 9: Multilayer Perceptron Architecture[16])

5.5.2 Random Forest

"Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction." [17]



(Figure 10: Random Forest Containing Two Decision Trees [20])

A decision is made at each feature node per tree using calculated values (entropy, information gain) to determine which leaf (class) the feature node best represents.

5.5.3 Gradient Boosting

"Machine learning boosting is a method for creating an ensemble. It starts by fitting an initial model (e.g. a tree or linear regression) to the data. Then a second model is built that focuses on accurately predicting the cases where the first model performs poorly. The combination of these two models is expected to be better than either model alone. Then you repeat this process of boosting many times. Each successive model attempts to correct for the shortcomings of the combined boosted ensemble of all previous models.

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error:

- If a small change in the prediction for a case causes a large drop in error, then next target outcome of the case is a high value. Predictions from the new model that are close to its targets will reduce the error.

- If a small change in the prediction for a case causes no change in error, then next target outcome of the case is zero. Changing this prediction does not decrease the error.

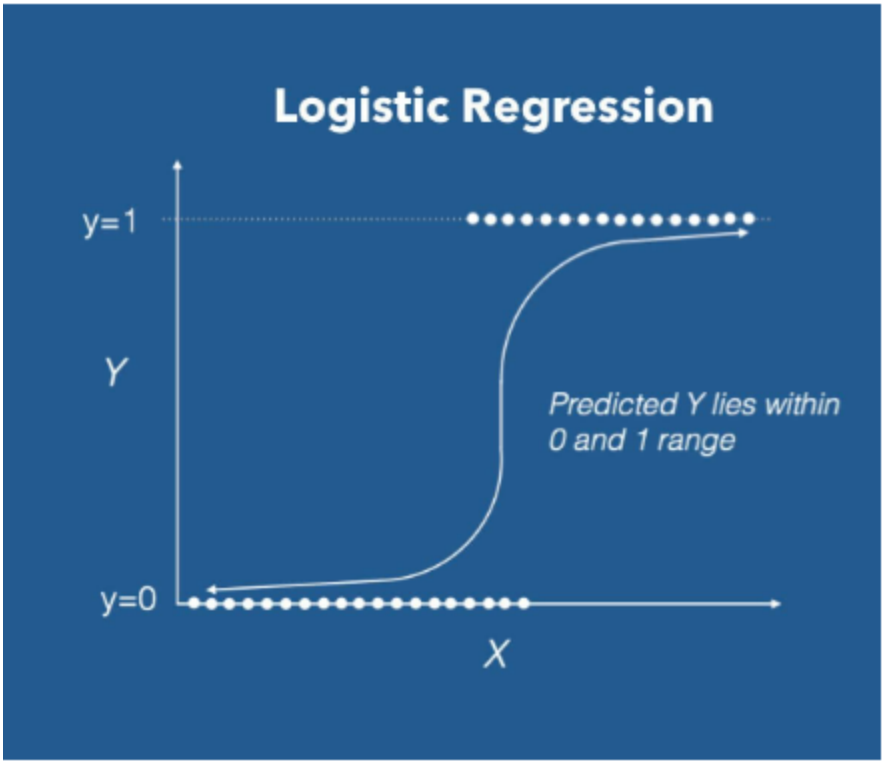
The name *gradient boosting* arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training case." [18]

5.5.4 Logistic Regression

"Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives.

A logistic regression model can take into consideration multiple input criteria. In the case of college acceptance, the logistic function could consider factors such as the student's grade point average, SAT score and number of extracurricular activities. Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into one of two outcome categories." [19]



(Figure 11: Logistic Regression Activation Function [20])

5.6 Statistical Assessments

5.6.1 ANOVA Test Assumptions

The researchers wanted to implement a statistical test to confirm if the top ranked features were statistically significant in predicted the area descriptions of 'Deprived' and 'Built-up'. To compare the categorical dependent variable to the independent quantitative data, an analysis of variance (ANOVA) test was constructed. There were three main assumptions that needed to be test before ANOVA was to be run [21]:

1. $\boldsymbol{Normality} - \textit{indepdendent variables follow a normal distribution}$
2. $\boldsymbol{Variance\ equality} - \textit{the variance of the different groups should be the same}$
3. $\boldsymbol{Independent\ \ Observations} - \textit{dependent variables were independently selected}$

(Figure 12: ANOVA Assumptions)

For the third assumptions, the area descriptions were assumed to be independently selected. Furthermore, outliers were not taken out of this assessment as there was massive class imbalance between the 'Deprived' and 'Built-up' areas. The researchers did not want to lose any potentially significant information. All statistical testing was conducted with an alpha of 0.05.

5.6.2 Kolmogorov – Smirnov Test for Normality

The Kolmogorov – Smirnov (K-S) test was introduced to address the first assumption of the ANOVA test. It was used to statistically prove if the distributions of the features for 'Deprived' and 'Built-up' followed a normal distribution. This test gave the researchers a

quantifiable metric on the distribution of the classes within the features. If the data was found to not follow a normal distribution, nonparametric procedures would be introduced to assess If there was a statistical difference between convariate features on 'Deprived' and 'Built-up' areas. [22]

$$H_o: The\ data\ follows\ a\ normal\ distribution$$
$$H_a: The\ data\ does\ not\ follow\ a\ normal\ distribution$$
$$K - S_{Test\ Statistics} = max_{1 \le i \le N}(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i)$$

(Figure 13: Kolmogorov-Smirnov Test Hypothesis test and Test Statistic Calculation)

Within the K-S test statistic calculation, F refers to the distribution being tested.

5.6.3 Levene test for equality of variance

The Levene test was introduced to test the second assumption of the ANOVA test. This assessment identified if the continuous features had equal variance between 'Deprived' and 'Built-up' classes. This test is essential to perform before any parametric testing as its results would indicate which statistical assessment to use.[23]

$$H_o: \sigma_1^2 = \sigma_2^2 \ldots = \sigma_k^2$$
$$H_a: \sigma_i^2 \neq \sigma_j^2 \text{ for at least one pair (i,j)}$$

(Figure 14: Levene Hypothesis test)

5.6.4 Kruskal- Wallis H-test

The Kruskal -Wallis test is a non-parametric test utilized when the data is not normally distributed. It assigns ranks to the values for testing rather than actual data points This test determines if the mean ranks of two or more groups are different by calculating the H- test statistic [24]. Mean rank refers to the average of ranks for observations within each. Furthermore, while previous research has recommended different non-parametric testing procedures [25], the researchers decided to go with Kruskal Wallis H-test on the data for its ability to work with data of unequal variance.[26]

$$H_o: The\ mean\ ranks\ for\ the\ samples\ are\ equal$$
$$H_a: The\ mean\ ranks\ for\ the\ sampels\ are\ not\ equal$$
$$H_{test\ statistic} = [\frac{12}{n(n+1)} \sum_{j=1}^{c} \frac{T_j^2}{n_j}] - 3(n+1)$$

(Figure 15: Kruskal Wallis Hypothesis Test and Statistic)

In the test statistic calculation, *T* refers to the sum of ranks in the j^{th} sample. N is the sum of samples sizes for all samples.

5.6.5 Chi Square test of independence

There were some categorical features found within the data as well. To compare these features to the area description, a chi square test of independence was introduced[27]:

$$H_o: The\ categorical\ features\ are\ independent$$

$$H_a: The\ categorical\ features\ are\ not\ independent$$

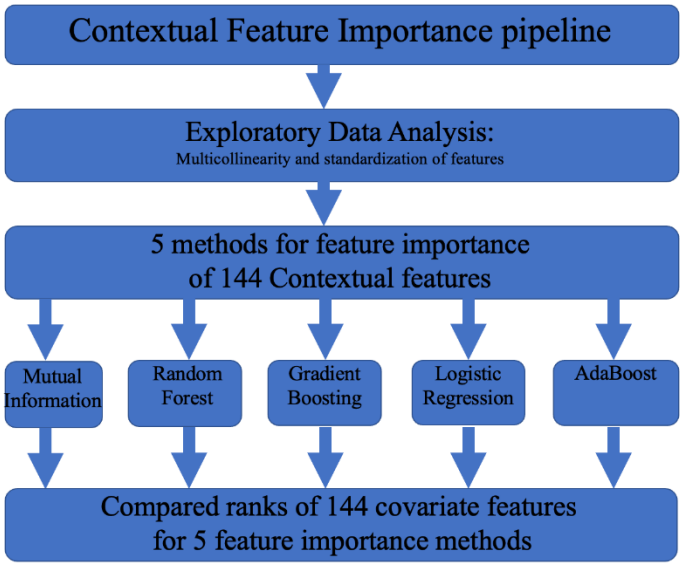
$$X^{2*} = \sum_{i=1} \frac{(O_i - E_i)^2}{E_i}$$

(Figure 16: Chi-Square Independence Hypothesis Test and Statistic)

for the Chi-Square test of independence, O_i refered to the original observation and E_i referred to the expected observation.

5.7 Contextual Feature Analysis Overview

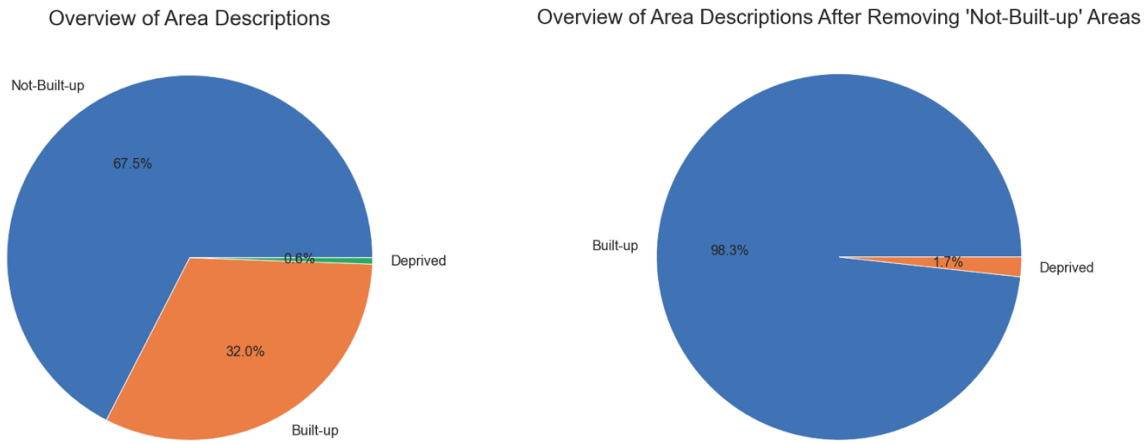
The contextual features were analyzed through exploratory data analysis and then feature importance. Once the contextual features were processed, exploratory data analysis was conducted on the entire data frame to identify any instances of multicollinearity. Also, the data was split on train/validation/test or 60/20/20.



(Figure 17: pipeline for Contextual Features)

5.7.1 Exploratory Data Analysis: Overview of Contextual Feature Target Variables

The first exploratory data analysis step taken for contextual features was to visualize the distribution of the target variables.

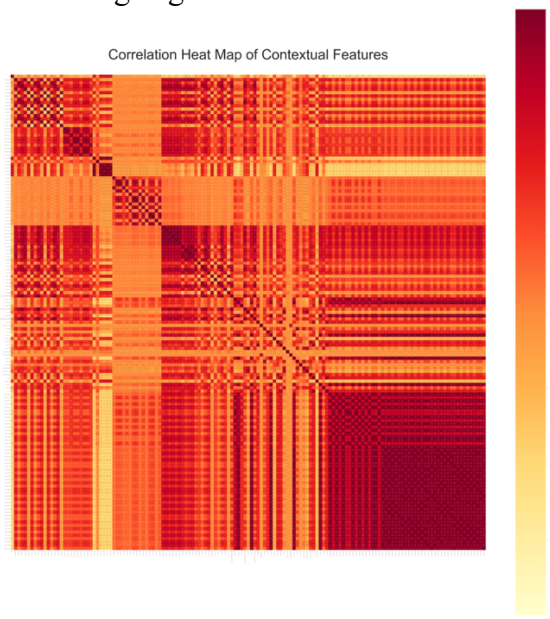


(Figure 18: Overview of Contextual Features)

After removing the 'Not-Built-up' class, the data frame had 15,471 samples of which 15202 (98.26%) were 'Built-up' areas and 269 (1.74%) were 'Deprived' areas.

5.7.2 Exploratory Data Analysis: Contextual Features Heat Map

The second data exploration technique the researchers utilized was to construct a heat map of the 144 contextual features. Dark red colors indicated a strong positive correlation and light-yellow colors indicated a strong negative correlation.



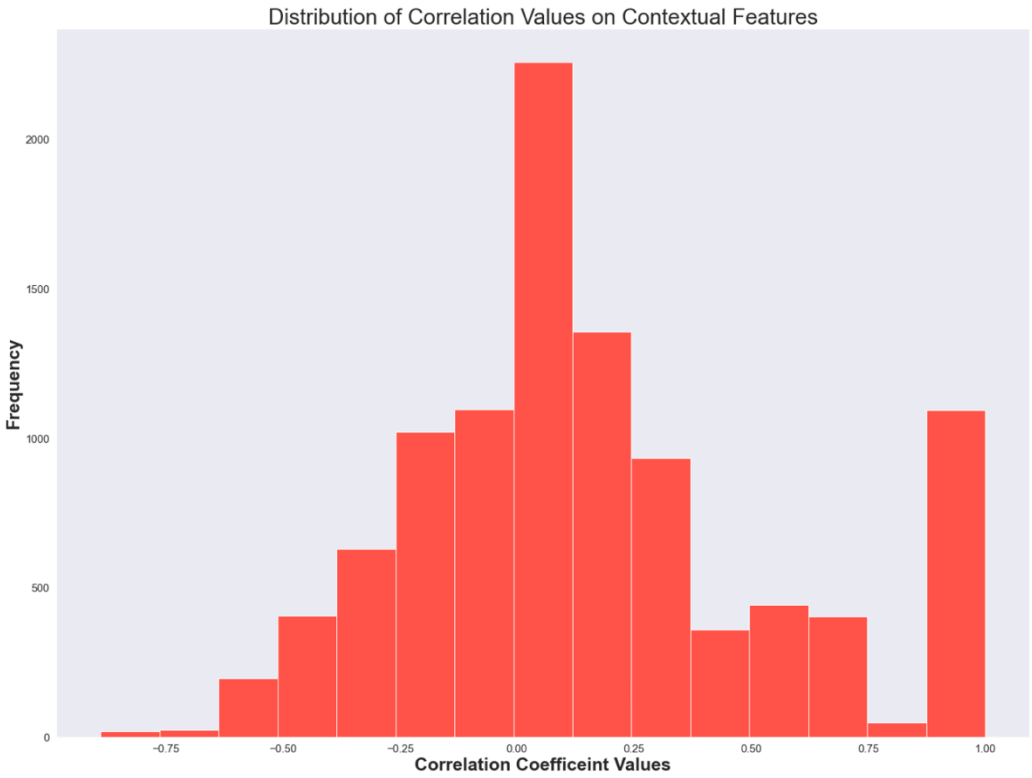
(Figure 19: Correlation Heat Map of Contextual Features)

From the heat map did indicate that there were instances of highly correlated values. Most notably, in the bottom right quadrant, many of the values had almost a correlation coefficient of 1. Upon inspection of these values, it was found that these were all from the

'gabor' contextual feature files. Further analysis needed to be conducted the distribution of correlated values.

5.7.3 Exploratory Data Analysis: Contextual Features Histogram Distribution

A histogram of the correlation features was constructed to visualize the range of correlations that the contextual features had with each other.

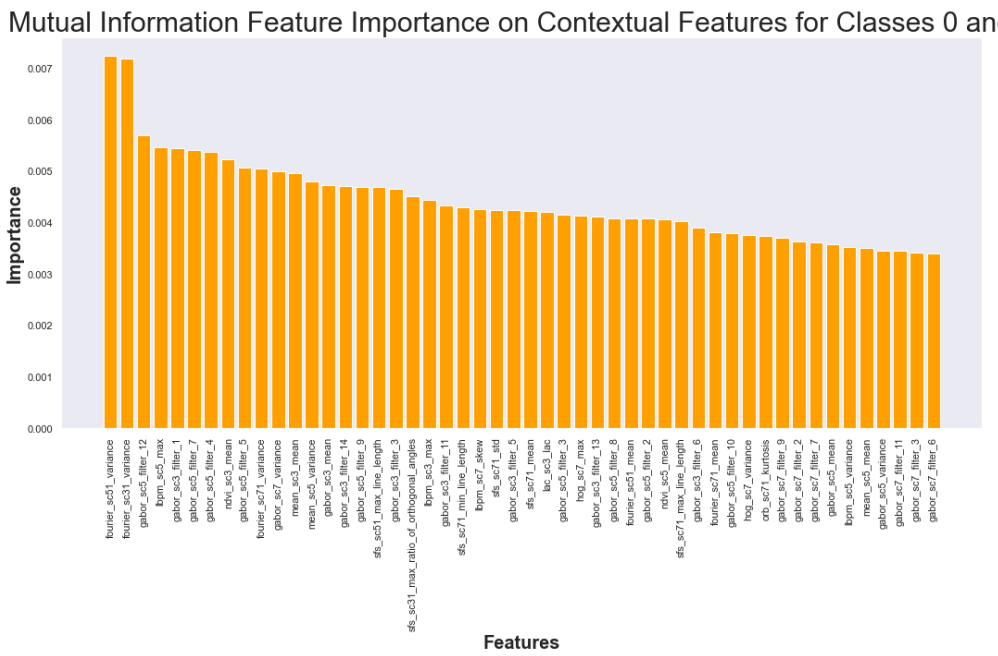


(Figure 20: Histogram of Correlation Coefficient of Contextual Features)

Upon inspection of the distribution of Correlated features for the Contextual features, many of the correlated values were close to 0.00. Furthermore, a few values were within the range of multicollinearity which the researchers defined as -0.75 to -1 and 0.75 to 1. Once exploratory analysis was completed, the contextual features were standardized for feature importance ranking in the subsequent section.

5.7.4 Contextual Feature Importance: Mutual Information

The mutual information score for each of the contextual features was calculated and ordered from most useful to least.



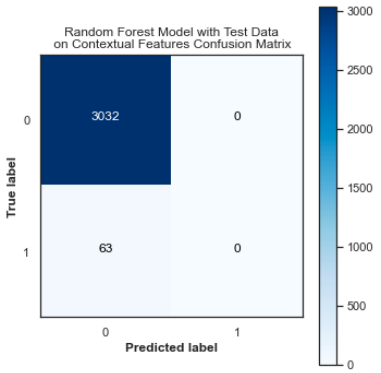
(Figure 21: Mutual Information Feature Importance)

Mutual information feature importance ranges from 0 to 1. By looking at the y-axis, all of the contextual features had low importance scores. Noticeably, the contextual feature that had the highest value was 'fourier_sc51_variance'.

5.7.5 Contextual Feature Importance: Random Forest Test Results

Grid search method was applied to the Random Forest model with the specific criteria of:

min_samples_split_grids = [2,10, 20, 50, 100]
min_samples_leaf_grids = [1,10, 20, 50, 100]
Cross validation = Stratifiedkfold()[28]



Random Forest Model with Test Data
on Contextual Features Classification Report

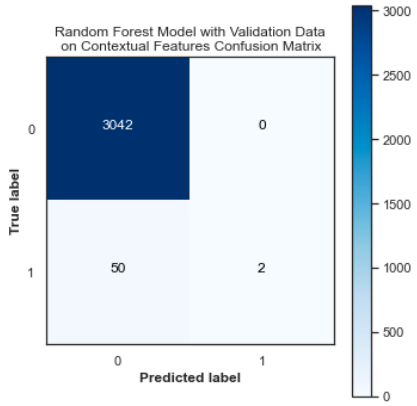
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 22: Random Forest Test Results Contextual Features)

The random forest model did not perform well on the testing data for the contextual features. From grid search, the best features for model_min_sample_leaf = 1 and model_min_sample_split = 2. The was unable to identify any of the deprived area.

5.7.6 Contextual Feature Importance: Random Forest Validation Results

Validation test set was conducted on the data on the random forest model.



Random Forest Model with Validation Data on Contextual Features Classification Report

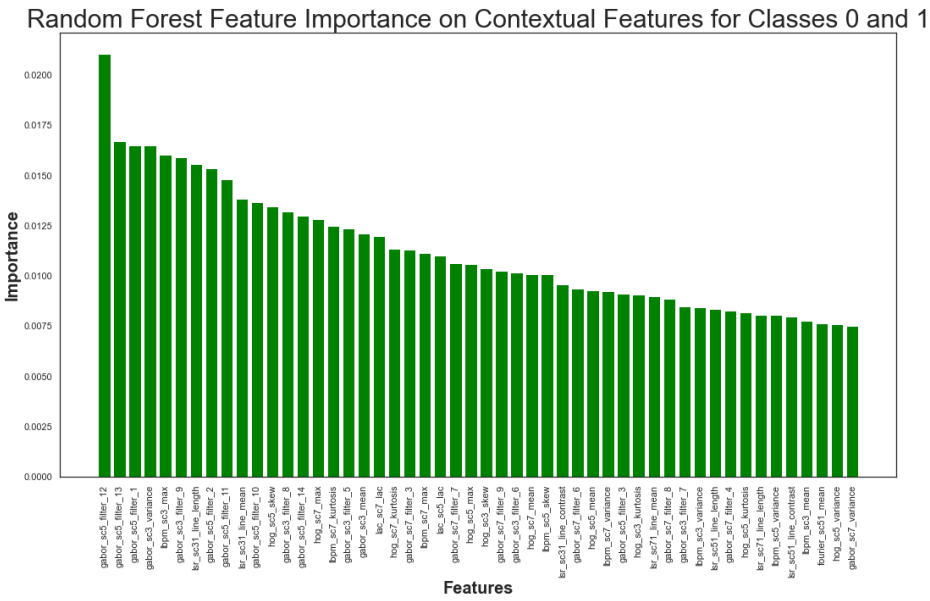
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	1.00	0.04	0.07	52
accuracy			0.98	3094
macro avg	0.99	0.52	0.53	3094
weighted avg	0.98	0.98	0.98	3094

(Figure 23: Random Forest Validation Results Contextual Features)

The validation data did slightly better than the testing data with f1-score on the 'deprived' area being 0.07. However, these results were still not significant for the model to be considered effective.

5.7.7 Contextual Feature Importance: Random Forest Feature Importance

Once testing and validation was complete. Feature Importance was conducted on the random first model.



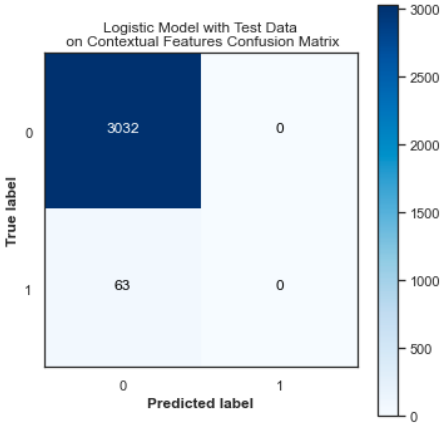
(Figure 24: Random Forest Feature Importance)

Many of the contextual feature values had low importance values as indicated by looking at the y-axis of the graph. The feature that was most significant was the 'gabor_sc5_filter_12'.

5.7.8 Contextual Feature Importance: Logistic Regression Test Results

The subsequent model test was the logistic regression model. Grid search was applied with the following parameters:

tol_grid = [10 ** -5, 10 ** -4, 10 ** -3, 10 ** -2, 10 ** -1]
C_grid = [0.001, 0.0001, 0.1, 1, 10]
Cross validation = StratifiedKfold



Logistic Model with Test Data
on Contextual Features Classification Report

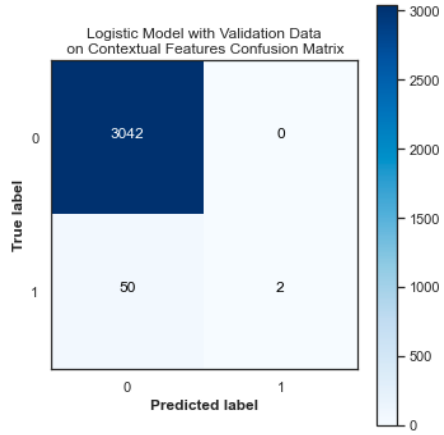
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 25: Logistic Regression Test Results Contextual Features)

The logistic model did not perform well on predicting the 'deprived' area. It was unable to correctly predict any. From grid search, it was found that the optimal parameters were model_C = 1 and model_tol = 1e-05.

5.7.9 Contextual Feature Importance: Logistic Regression Validation Results

Validation testing was conducting on the logistic model as well.



Logistic Model with Test Data
on Contextual Features Classification Report

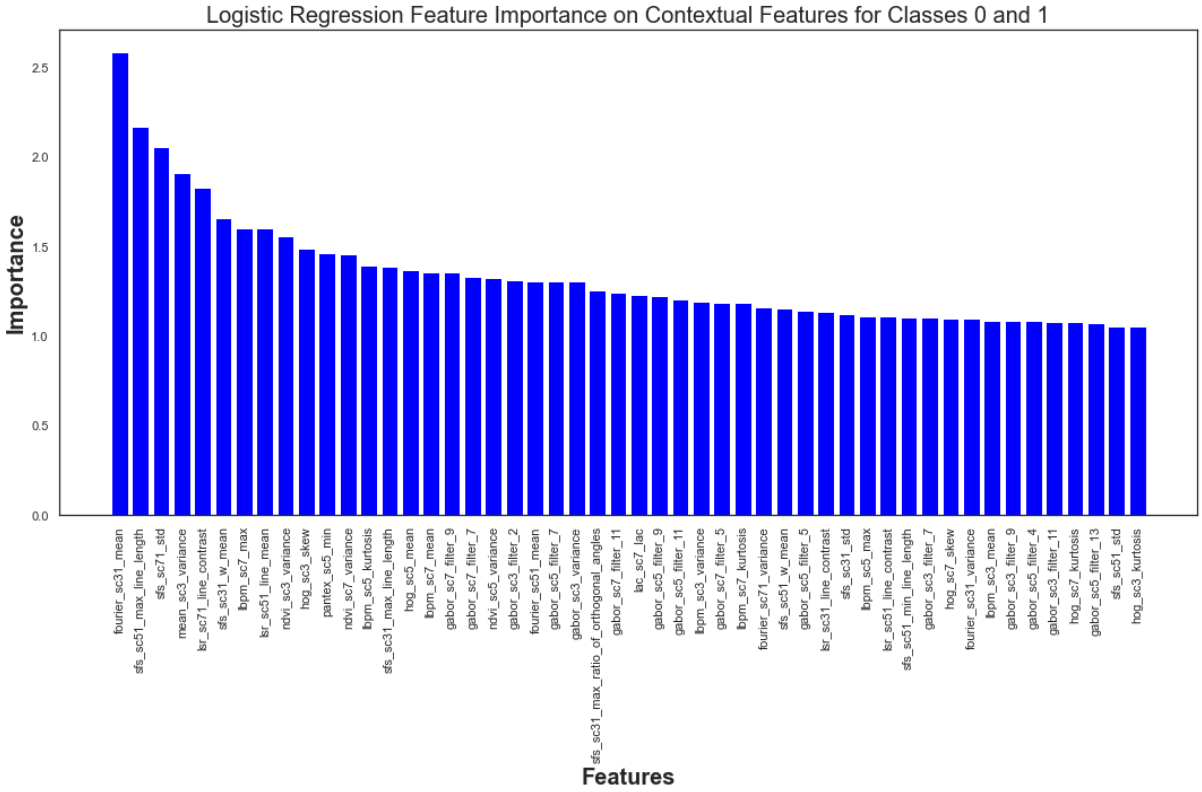
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	1.00	0.04	0.07	52
accuracy			0.98	3094
macro avg	0.99	0.52	0.53	3094
weighted avg	0.98	0.98	0.98	3094

(Figure 26: Logistic Regression Validation Results Contextual Features)

The validation data did slightly better than the testing data with f1-score on the 'Deprived' area being 0.07. However, these results were still not significant for the model to be considered effective.

5.7.10 Contextual Feature Importance: Logistic Regression Feature Importance

Feature importance for the logistic regression model was constructed by ordering the coefficients of the 144 contextual features.



(Figure 27: Logistic Regression Feature Importance on Contextual Features)

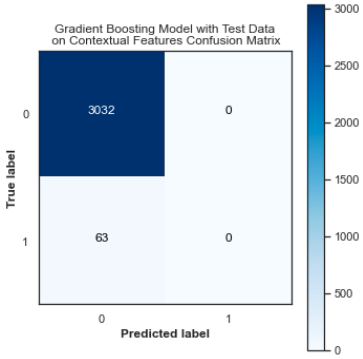
The contextual feature with the largest coefficient value was found to be 'fourier_sc31_mean'.

5.7.11 Contextual Feature Importance: Gradient Boosting Test Results

Testing was conducted on a gradient boosting model. Grid search was performed with the following parameters:

'loss': ['deviance'],
'criterion': ['friedman_mse', 'mse'],
'n_estimators': [100],
'subsample': [1.0, 0.6],
"learning_rate": [0.01, 0.05],
"min_samples_split": np.linspace(0.1, 0.5, 3),

"min_samples_leaf": np.linspace(0.1, 0.5, 3),
"max_depth": [3, 8],
"max_features": ["log2", "sqrt"],
Cross validation: StratifiedKfold



Gradient Boosting Model with Testing Data
on Contextual Features Classification Report

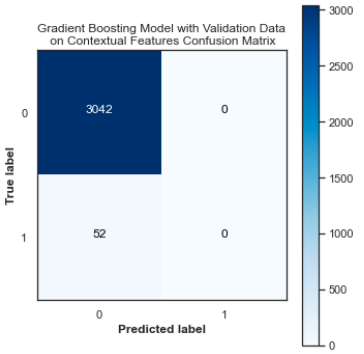
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 28: Gradient Boosting Test Results Contextual Features)

The gradient boosting model was unable to identify any of the 'deprived' areas. From grid search, the best parameters were {'criterion': 'friedman_mse', 'learning_rate': 0.01, 'loss': 'deviance', 'max_depth': 3, 'max_features': 'log2', 'min_samples_leaf': 0.1, 'min_samples_split': 0.1, 'n_estimators': 100, 'subsample': 1.0.

5.7.12 Contextual Feature Importance: Gradient Boosting Validation Results

Validation set was assessed on the gradient boosting model as well.



Gradient Boosting Model with Validation Data
on Contextual Features Classification Report

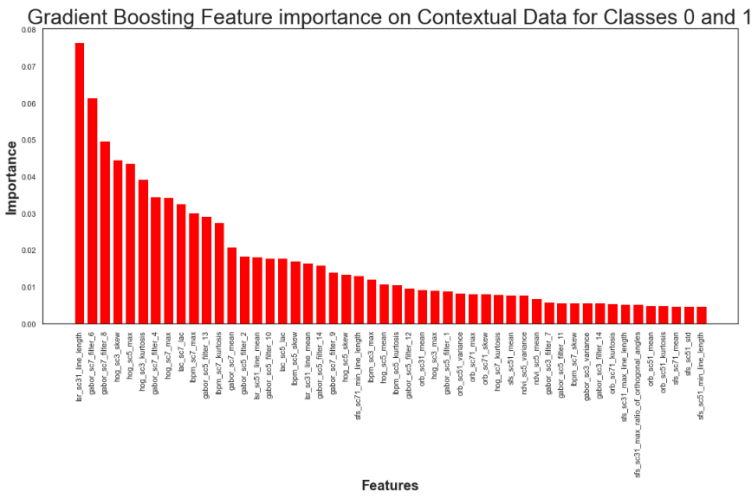
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	0.00	0.00	0.00	52
accuracy			0.98	3094
macro avg	0.49	0.50	0.50	3094
weighted avg	0.97	0.98	0.97	3094

(Figure 29: Gradient Boosting Validation Results Contextual Features)

The model did not perform well and it was unable to capture any of the 'deprived' areas.

5.7.13 Contextual Feature Importance: Gradient Boosting Feature Importance

Once test and validation were concluded for gradient boosting, feature importance was implemented on the model



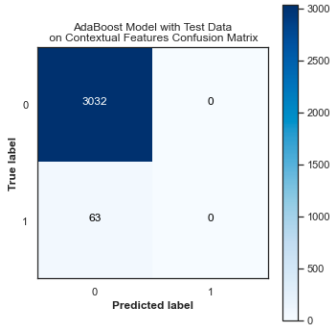
(Figure 30: Gradient Boosting Feature Importance on Contextual Features)

The contextual feature that performed the best was 'ibpm_sc7_kutosis'. Noticeably, by looking at the y-axis, many of these features were found close to zero which would imply that they were not significant in predicting 'deprived' or 'built-up' areas.

5.7.14 Contextual Feature Importance: AdaBoost Test Results

The final feature importance model that was constructed was the adaboost model. Grid search was applied to this model with the following parameters:

'n_estimators' = [50, 100, 150, 200],
"learning_rate" = [0.01, 0.05, 0.025]
Cross validation: StratifiedKfolds



AdaBoost Model with Testing Data
on Contextual Features Classification Report

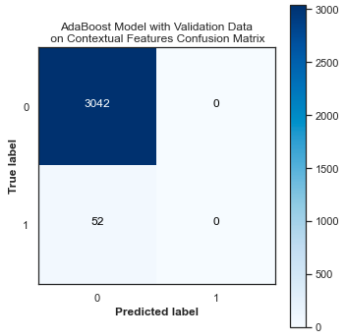
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 31: AdaBoost Test Results Contextual Features)

Similar to the other models in the contextual features, adaboost did not perform well with the contextual features. From grid search the optimal parameters were found to be a learning rate = 0.01 with n_estimators = 50.

5.7.15 Contextual Feature Importance: AdaBoost Validation Results

A validation set was introduced to the adaboost model.



AdaBoost Model with Validation Data
on Contextual Features Classification Report

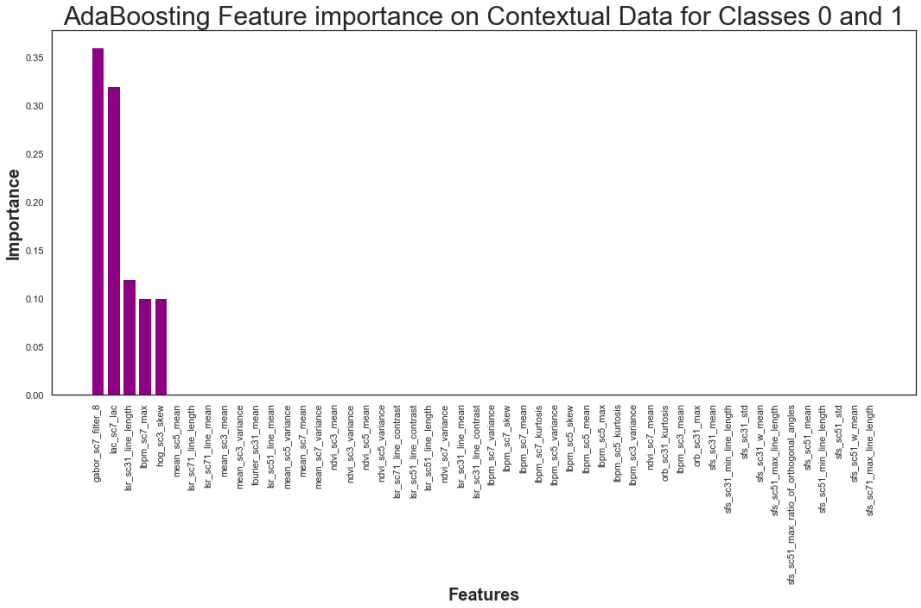
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	0.00	0.00	0.00	52
accuracy			0.98	3094
macro avg	0.49	0.50	0.50	3094
weighted avg	0.97	0.98	0.97	3094

(Figure 32: AdaBoost Validation Results Contextual Features)

The validation data for adaboost did not perform well on the contextual features. It was unable to classify any of the 'deprived' areas.

5.7.16 Contextual Feature Importance: AdaBoost Feature Importance

After testing and validating was concluded, feature importance of the contextual feature for adaboost was explored.



(Figure 33: AdaBoost Feature Importance for Contextual Features)

It was found that only five contextual features were found to be of importance. They were, in descending order, gabor_sc7_filter_8, lac_sc7_lac, lac_sc31_line_length, ibpm_sc7_max, and hog_sc3_skew.

5.7.17 *Contextual Feature Importance: Comparing Models*

Model	Validation F1 for 'Deprived'
Random Forest	0.07
Logistic Regression	0.07
Gradient Boosting	0.00
Adaboost	0.00

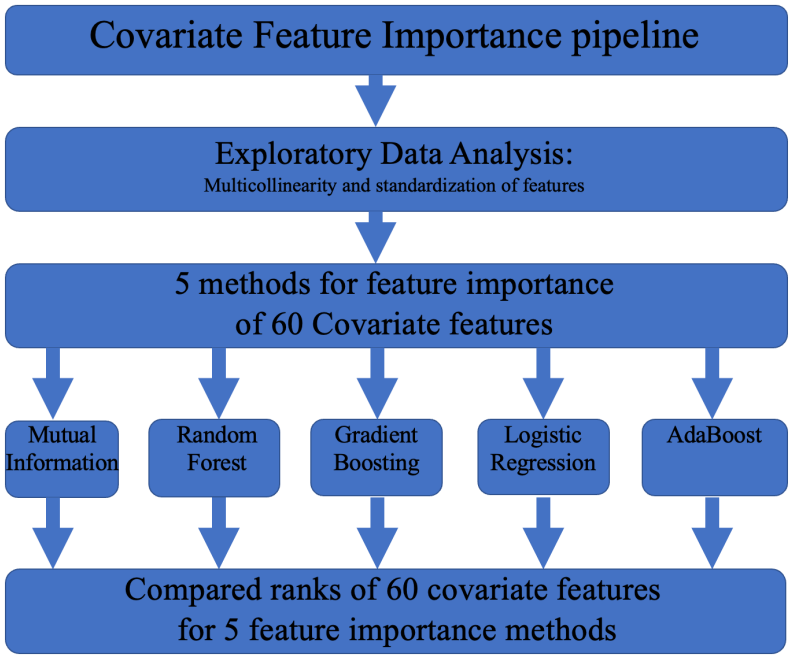
(Figure 34: Overview of Validation Results for Minority Class for Contextual Features)

Overall, the contextual features were not useful in identifying 'Deprived' and 'Built-up' areas. All the models with either the testing or validation data had significantly or zero f1 scores on the deprived category. When comparing five of the feature importance models, the variable that was most significant was 'lbpm_sc7_max'. Since the contextual features were not found to be useful in predicting the 'Deprived' and 'Built-up' areas, no further analysis was conducted on the data.

5.8 Covariate Feature

5.8.1 Covariate Feature Analysis Overview

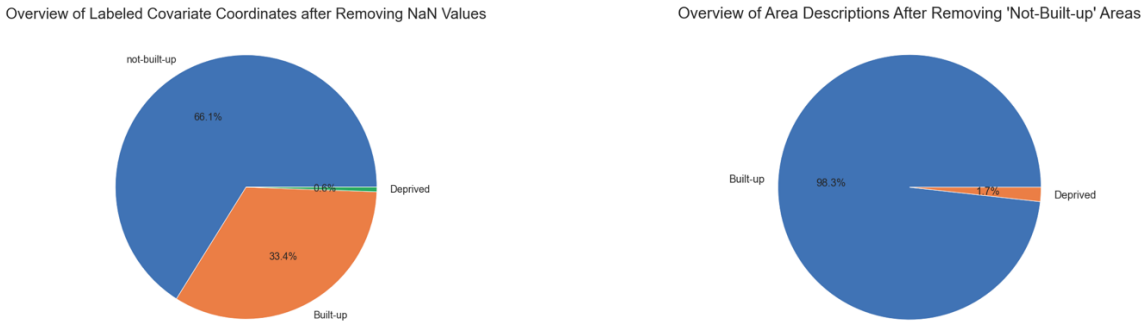
The covariate features were analyzed through exploratory data analysis and then feature importance. Once the covariate features were processed, exploratory data analysis was conducted on the entire data frame to identify any instances of multicollinearity. Also, the data was split on train/validation/test or 60/20/20.



(Figure 35: pipeline for Covariate Features)

5.8.2 Exploratory Data Analysis: Overview of Covariate Feature Target Variables

Prior to any formal analysis of the covariate features, the data was check for any ‘nan’ values. There were 1987 nan values found within the dataset. They were 1986 values for the ‘not-Built-up’ class and one for the ‘Built-up’ class. Furthermore, it was found that one covariate feature ‘ph_gdmhz_2005’ contained ‘nan’ values so it was removed.

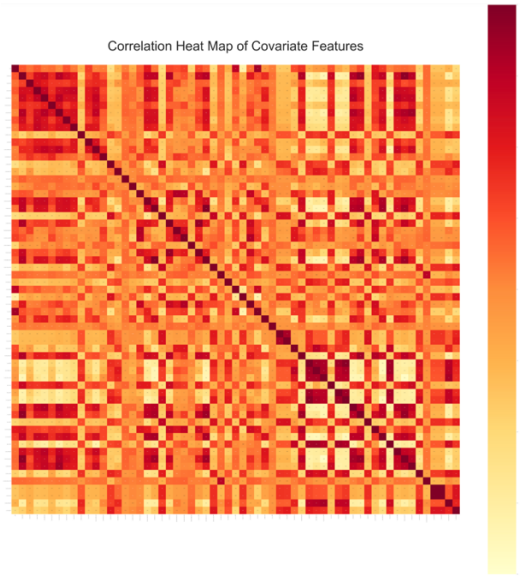


(Figure 36: Overview of Target Variable for Covariate Features)

After removing the ‘not-Built-up’ class, the final data frame for analysis consisted of 15,470 samples and 60 covariate features. There were 15201 samples for the ‘Built-up’ class, and 269 samples for the ‘Deprived’ class.

5.8.3 Exploratory Data Analysis: Covariate Features Heat Map

The second data exploration technique the researchers utilized was to construct a heat map of the 60 covariate features. Dark red colors indicated a strong positive correlation and light-yellow colors indicated a strong negative correlation.

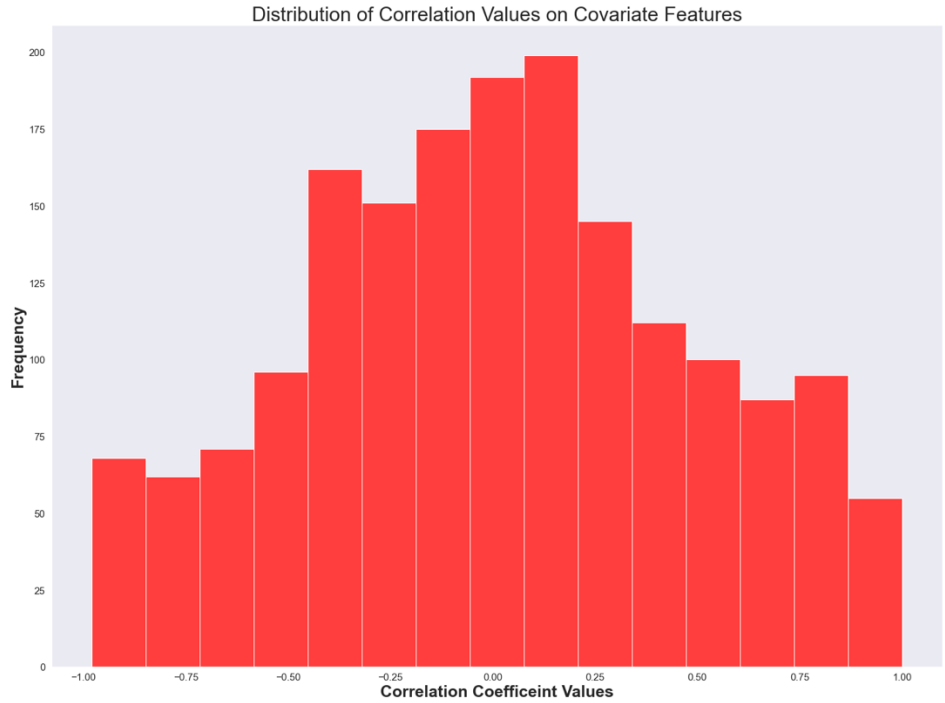


(Figure 37: Correlation Heat Map for Covariate Features)

From the heat map did indicate that there were instances of highly correlated values. Most notably, in the bottom right quadrant, many of the values had almost a correlation coefficient of 1. This variable was removed from analysis. There did appear to be some checkered pattern within the heat map. This was most prominent in the bottom right corner.

5.8.4 Exploratory Data Analysis: Covariate Features Histogram Distribution

A histogram of the correlation features was constructed to visualize the range of correlations that the covariate features had with each other.

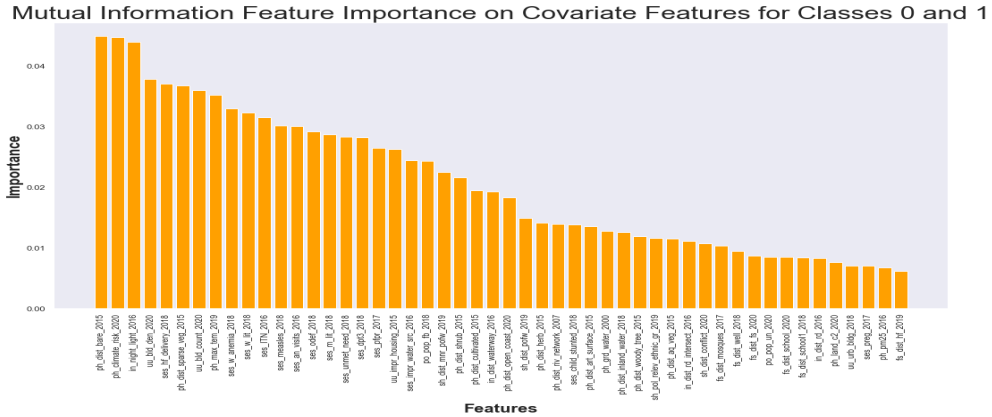


(Figure 38: Histogram of Correlation Coefficient of Covariate Features)

From observing the distribution of correlation values, it appeared that there was a significant number of features that appeared in the range between -0.8 to -0.5 and for 0.5 to 0.8. Noticeably, there was many covariate features that had zero correlation with each other.

5.8.5 Covariate Feature Importance: Mutual Information

The mutual information score for each of the contextual features was calculated and ordered from most useful to least.



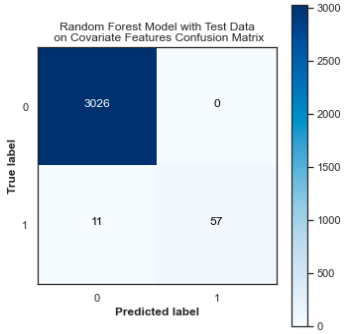
(Figure 39: Mutual Importance Feature Selection on Covariate Features)

Mutual information feature importance ranges from 0 to 1. The covariate feature found to be the most significant was ‘ph_dist_bare_2015’. The remaining covariate features decreased gradually in their significance.

5.8.6 Covariate Feature Importance: Randoom Forest Test Results

Grid search method was applied to the Random Forest model with the specific criteria of:

min_samples_split_grids = [2,10, 20, 50, 100]
min_samples_leaf_grids = [1,10, 20, 50, 100]
Cross validation = Stratifiedkfold



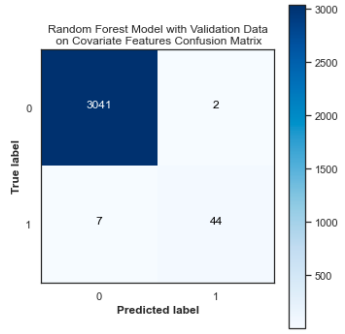
Random Forest Model with Test Data on Covariate Features Classification Report	precision	recall	f1-score	support
0	1.00	1.00	1.00	3026
1	1.00	0.84	0.91	68
accuracy			1.00	3094
macro avg	1.00	0.92	0.96	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 40: Random Forest Test Results on Covariate Features)

The model performed well on the covariate data. It had an F1 score of 0.91 for the 'Deprived' areas. From grid search, it was found that the best parameters were model_min_smaples_leaf = 1 and model_min_samples_split = 2.

5.8.7 Covariate Feature Importance: Random Forest Validation Results

A validation set was constructed for the random forest model as well.



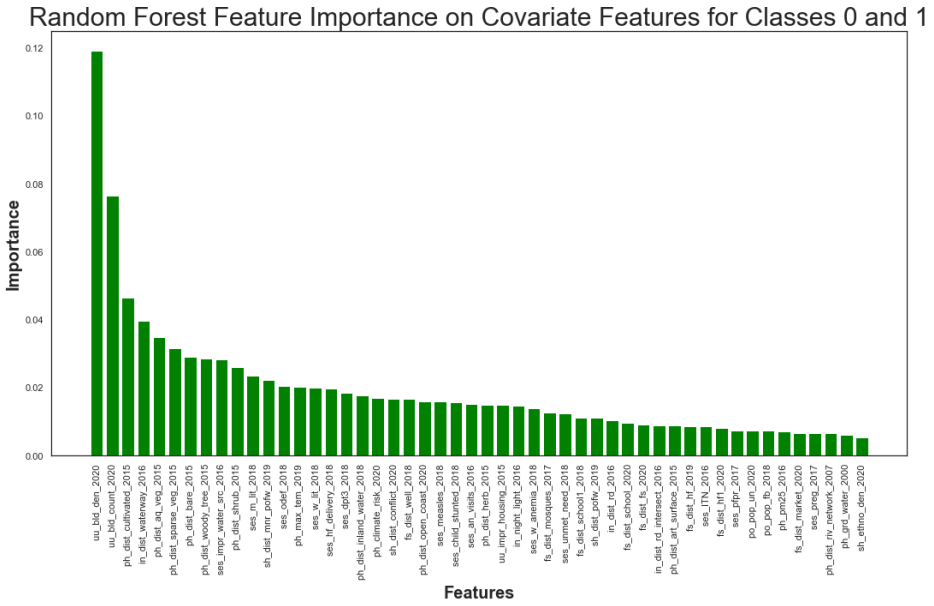
Random Forest Model with Validation Data on Covariate Features Classification Report	precision	recall	f1-score	support
0	1.00	1.00	1.00	3043
1	0.96	0.86	0.91	51
accuracy			1.00	3094
macro avg	0.98	0.93	0.95	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 41: Random Forest Validation Results on Covariate Features)

The validation results were superb. The F1 score for the 'Deprived' class was found to be 0.91.

5.8.8 Covariate Feature Importance: Random Forest Feature Importance

Once testing and validation was concluded for the random forest model, feature importance was constructed for the covariate features.



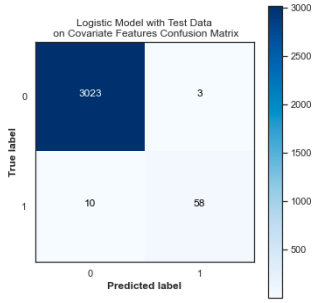
(Figure 42: Random Forest Feature Importance on Covariate Features)

The covariate feature that was found to be the most significant, according to random forest feature selection, was 'uu_bld_den_2020'.

5.8.9 Covariate Feature Importance: Logistic Regression Test Results

Logistic regression feature importance was constructed for the covariate features. Grid search was applied with the following parameters:

tol_grid = [10 ** -5, 10 ** -4, 10 ** -3, 10 ** -2, 10 ** -1]
C_grid = [0.001, 0.0001, 0.1, 1, 10]
Cross validation = StratifiedKfold



Logistic Model with Test Data
on Covariate Features Classification Report

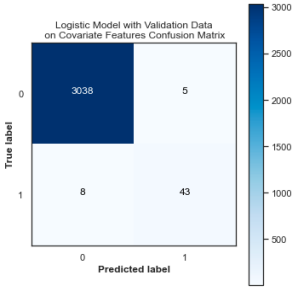
	precision	recall	f1-score	support
0	1.00	1.00	1.00	3026
1	0.95	0.85	0.90	68
accuracy			1.00	3094
macro avg	0.97	0.93	0.95	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 43: Logistic Regression Test Results on Covariate Features)

The test results from the logistic model were very good. The F1 score for the 'Deprived' class was 0.90. From grid search, the optimal parameters were found to be model_C = 10 and model_to = 1e-05.

5.8.10 Covariate Feature Importance: Logistic Regression Validation Results

A validation set was tested on the logistic model



Logistic Model with Test Data
on Covariate Features Classification Report

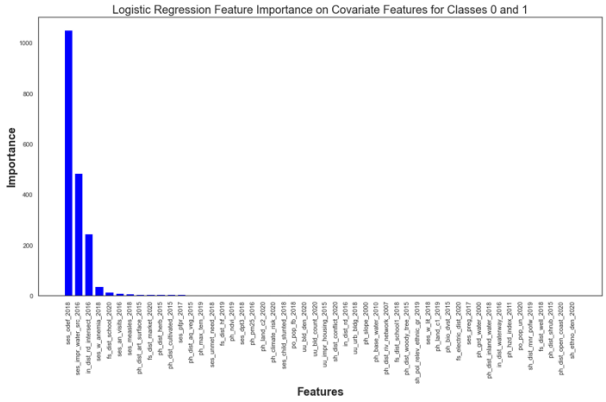
	precision	recall	f1-score	support
0	1.00	1.00	1.00	3043
1	0.90	0.84	0.87	51
accuracy			1.00	3094
macro avg	0.95	0.92	0.93	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 44: Logistic Regression Validation Results on Covariate Features)

The validation model did superb. It had an F1 score of 0.87 for the 'Deprived' area. This model was able to identify 43 out of 51 'Deprived' areas.

5.8.11 Covariate Feature Importance: Logistic Regression Feature Importance

With testing and validation completed for the logistic model, feature importance for the covariate features was constructed



(Figure 45: Logistic Regression Feature Importance on Covariate Features)

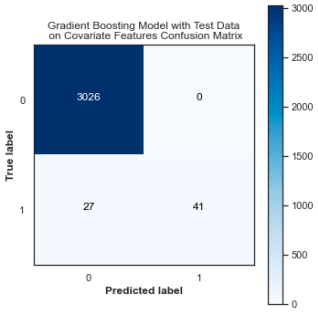
The value that was found to be the most important according to feature importance through the logistic regression method was 'ses_odef_2018'.

5.8.12 Covariate Feature Importance: Gradient Boosting Test Results

Gradient boosting model was constructed for the test data. Grid search was performed with the following parameters:

- 'loss': ['deviance'],
- 'criterion': ['friedman_mse', 'mse'],
- 'n_estimators': [100],
- 'subsample': [1.0, 0.6],
- "learning_rate": [0.01, 0.05],
- "min_samples_split": np.linspace(0.1, 0.5, 3),
- "min_samples_leaf": np.linspace(0.1, 0.5, 3),
- "max_depth": [3, 8],
- "max_features": ["log2", "sqrt"],

- Cross validation: StratifiedKfold



Gradient Boosting Model with Testing Data
on Covariate Features Classification Report

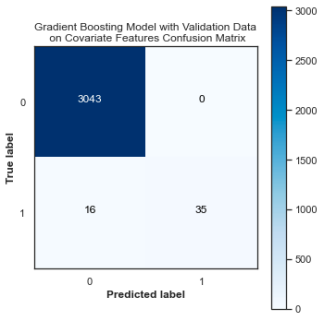
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3026
1	1.00	0.60	0.75	68
accuracy			0.99	3094
macro avg	1.00	0.80	0.87	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 46: Gradient Boosting Test Results on Covariate Features)

The gradient boosting model performed well. Although the F1 score of 0.75 was the lower than the random forest and logistic models, it managed to capture a significant amount of data from the 'Deprived' class. From grid search the best parameters were found to be 'criterion': 'friedman_mse', 'learning_rate': 0.05, 'loss': 'deviance', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 0.1, 'min_samples_split': 0.1, 'n_estimators': 100, 'subsample': 1.0.

5.8.13 Covariate Feature Importance: Gradient Boosting Validation Results

A validation set was implemented on the gradient boosting model.



Gradient Boosting Model with Validation Data
on Covariate Features Classification Report

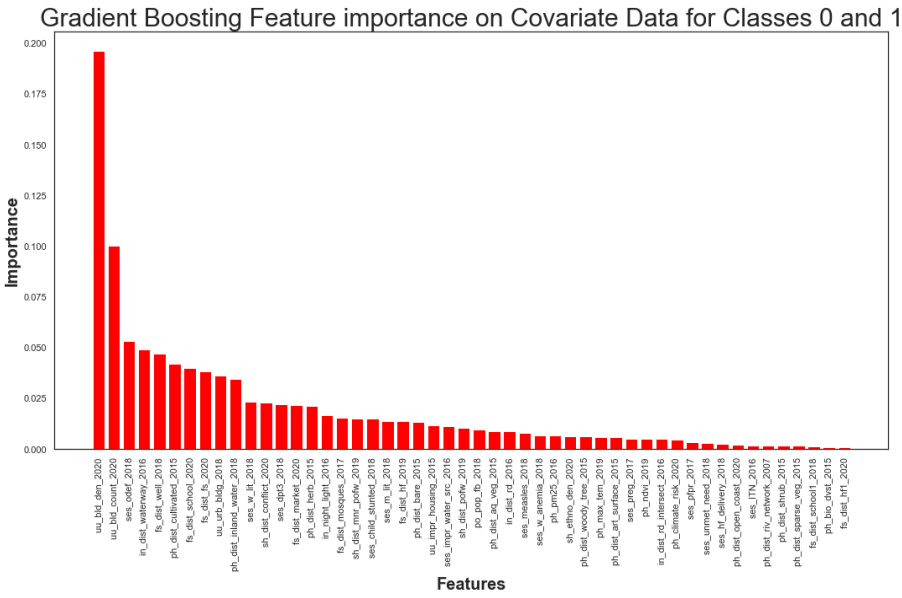
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3043
1	1.00	0.69	0.81	51
accuracy			0.99	3094
macro avg	1.00	0.84	0.91	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 47: Gradient Boosting Validation Results on Covariate Features)

The validation set performed better than the test set for gradient boosting. It yielded an F1 score of 0.81 for the 'Deprived' class.

5.8.14 Covariate Feature Importance: Gradient Boosting Feature Importance

Once testing and validation was completed for gradient boosting, feature importance was conducted on the 60 covariate features.



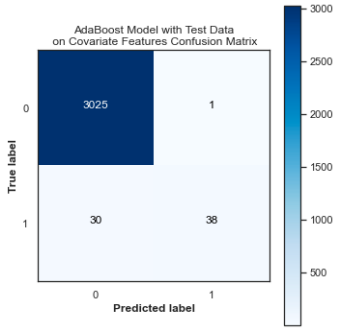
(Figure 48: Gradient Boosting Feature Importance on Covariate Features)

Gradient boosting feature importance found that the most significant covariate feature was 'uu_bld_den_2020'.

5.8.15 Covariate Feature Importance: AdaBoost Test Results

The final model tested on the covariate features was adaboost. Grid search was applied to this model with the following parameters:

- 'n_estimators' = [50, 100, 150, 200],
- 'learning_rate' = [0.01, 0.05, 0.025]



AdaBoost Model with Testing Data
on Covariate Features Classification Report

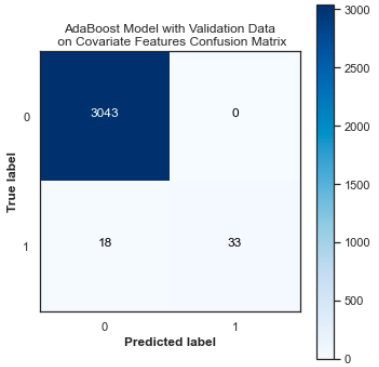
	precision	recall	f1-score	support
0	0.99	1.00	0.99	3026
1	0.97	0.56	0.71	68
accuracy			0.99	3094
macro avg	0.98	0.78	0.85	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 49: AdaBoost Test Results on Covariate Features)

The model did moderately well in identifying the 'Deprived' areas with an F1 score of 0.71. Grid search revealed that the most optimal parameters were a learning rate = 0.05 and n_estimators = 200.

5.8.16 Covariate Feature Importance: AdaBoost Validation Results

Once testing was complete, the AdaBoost model was assessed on validation data.



AdaBoost Model with Validation Data
on Covariate Features Classification Report

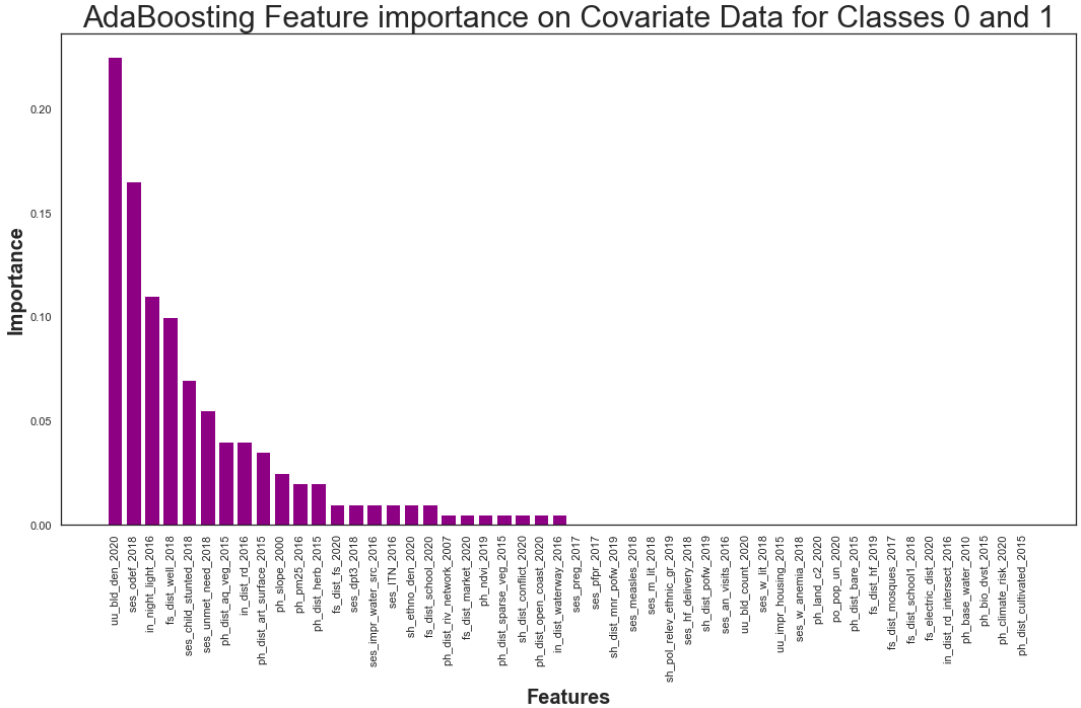
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3043
1	1.00	0.65	0.79	51
accuracy			0.99	3094
macro avg	1.00	0.82	0.89	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 50: AdaBoost Validation Results on Covariate Features)

The model performed well on the validation data. The model was able to identify 33 out of the 51 Deprived areas.

5.8.17 Covariate Feature Importance: AdaBoost Feature Importance

Once the testing and validation was completed for the AdaBoost model, feature importance was conducted.



(Figure 51: AdaBoost Feature Importance on Covariate Features)

The most significant feature found within feature selection was ‘uu_bld_den_2020’.

5.8.18 *Covariate Feature Importance: Comparing Models*

Model	Validation F1 for ‘Deprived’

Random Forest	0.91
Logistic Regression	0.87
Gradient Boosting	0.81
Adaboost	0.79

(Figure 52: Overview of Validation Results for Minority Class for Contextual Features)

The models for the covariate feature performed exceptionally well when compared to the contextual features.

Covariate Features	**Logistic Rank**	**Random Forest Rank**	**Gradient Boosting Rank**	**AdaBoost Rank**	**Mutual Information Rank**	**Overall Rank**
uu_bld_den_2020	24	1	1	1	4	1
ses_odef_2018	1	13	3	2	14	2
ses_impr_water_src_2016	2	9	24	15	21	3
uu_bld_count_2020	25	2	2	35	7	4
ses_dpt3_2018	18	17	13	14	17	5
ph_dst_aq_veg_2015	13	5	27	7	36	6

(Figure 53: Top Ranks of Covariate Features)

After all feature importance methods were finished, all the ranks were calculated and compared to one another. Most notably, the covariate feature that performed the best was found to be ‘uubld_den_2020’ within three of the five feature importance methods. Furthermore, ‘uu_bld_den_2020’, ‘ses_odef_2018’, ses_impr_water_src_2016’, ‘ses_dpt3_2018’, ‘ph_dist_herb_2015’, ‘ses_child_stunted_2018’, and ‘ses_measles_2018’ were found to be within the top thirty rank of all sixty covariate features when considering each of the five feature importance methods.

Covariate Features	**Logistic Rank**	**Random Forest Rank**	**Gradient Boosting Rank**	**AdaBoost Rank**	**Mutual Information Rank**	**Overall Rank**
Ph_grd_water_2000	41	49	59	55	37	55
Ph_base_water_2010	31	58	53	47	54	56
Fs_electric_dist_2020	39	54	54	45	60	57

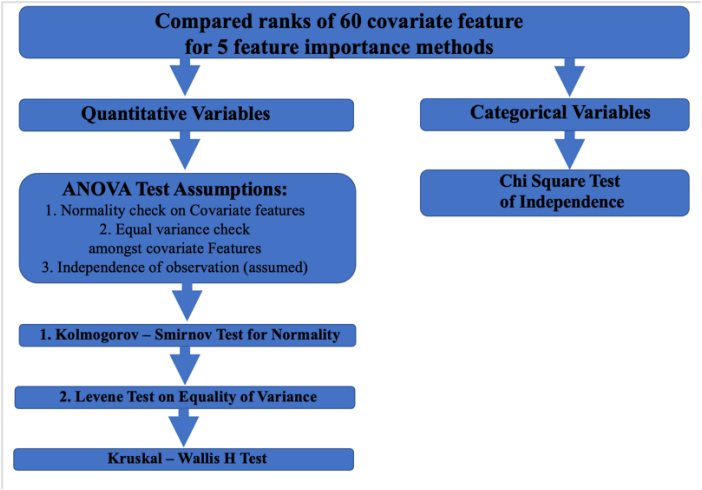
Fs_dist_hf1_2020	57	41	50	58	50	58
Ph_hzd_index_2011	44	55	55	56	56	59
ph_land_c1_2019	37	60	58	57	57	60

(Figure 54: Bottom Ranks of Covariate Features)

While feature importance was useful in identifying the covariate features that were important, it could also be used to identify features that had the least significant in predicting the target variable. A total of 11 covariate features were found to be consistently in the 30-60 range for all three covariate features. They were from least to most significant: 'ph_land_c1_2019', 'Ph_hzd_index_2011', 'Fs_dist_hfl_2020', 'Fs_electric_dist_2020', 'ph_base_water_2000, 'ph_grd_water_2000', 'ph_bio_dvst_2015', 'po_pop_un_2020', 'sh_pol_relev_ethnic_gr_2019', 'fs_dist_school1_2018', 'ses_preg_2017'. Through the five feature importance methodologies, the researchers were able to identify seven important and eleven non important features.

5.9. *Statistical Methods*

5.9.1 *Covariate Feature Importance: Statistical Testing*



(Figure 55: Statistical Testing on Covariate Features)

The methodology for conducting statistical tests on the Covariate features were visualized in a graphic. There were 54 continuous variables and 6 categorical variables.

5.9.2 *Kolmogorov – Smirnov test for Normality*

The KW test was constructed to see if the distribution of covariate features for the 'Deprived' and 'Built-up' areas followed a normal distribution. Testing revealed all the covariate features all had p-values lower than the alpha of 0.05. This led the researchers to conclude that the covariate features for either class did not follow a normal distribution. The parameter *kstets* from the Scipy library was utilized to conduct the normality test[29].

5.9.3 *Levene Test on Equality of Variance*

The Levene test was introduced to see if the continuous covariate variables had equal variance between the 'Deprived' and 'Built up' regions. It was found that five of the 54 covariate features did have equal variance. The parameter *levene* from the Scipy library was implemented to conduct an equality of variance assessment [30].

Levene Test Results: Covariate Features with Equal variance		
Covariate Feature	**P-value**	**Rank**
po_pop_fb_2018	0.2902	38
ph_ndvi_2019	0.5175	45
po_pop_un_2020	0.0804	52
ph_bio_dvst_2015	0.5039	55
ph_base_water_2010	0.6432	56

(Figure 56: Leven Testing Results on Covariate Features)

These five variables were all found to have low overall ranks.

5.9.4 *Kruskal Wallis H-test on Covariate Features*

The Kruskal Wallis test was implemented on the continuous covariate features that were not normally distributed. It was found that the top 21 ranked features were statistically significant in determining the difference between the areas of 'Deprived' and 'Built-up'. The function *Kruskal* from the Scipy library was implemented on the covariate features [31].

5.9.5 *Chi Square test on Covariate Features*

A Chi Square test of independence was conducted on the six variables. The function *chi2_contingency* from the Scipy library was implanted to test the independence between the categorical features and area description[32]

Chi Square Test Results			
	P-value	**Number of categories**	**rank**
fs_electric_dist_2020	0.7381	2	57
ph_hzd_index_2011	0.00	6	59
ph_land_c1_2019	0.00	11	60
ph_land_c2_2020	0.00	8	53
sh_pol_relev_ethnic_gr_2019	0.00	2	50
uu_urb_bldg_2018	0.00	3	48

(Figure 57: Chi Square Test of Independence Results on Covariate Features)

The results of the chi square test revealed that only 'fs_electric_dist_2020' showed there was no relationship between the categorical features. According to 'fs_electric_dist_2020', knowing if an area is 'Deprived' did not predicted if another value would be 'Built-up'. The other five features did indicate that there was a relationship between the categorical features of area description.

The researchers were able to identify the rank of the covariate features and proved statistically that the top 21 features were useful in identifying 'Deprived' and 'built-up' areas.

6 Results and Discussion

6.1 Experimentation protocol

As mentioned in the *5 Solution and Methodology* section above, the model types the researchers trained and tested on were: logistic regression, multi-layer perceptron (MLP) neural network, random forest, and gradient boosting classifiers. These models were chosen to give variety (linear, non-linear, & neural network models) in the model testing phase to see if any particular model type would best suit this classification problem.

The metric the researchers chose to identify as the indicator of model performance is the F1-score, particularly the micro F1-score for the deprived class and the macro F1-scrore for the entire model. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers and can be compared across models to determine which model performs best at correctly identifying and classifying instances [33]. The F1-score of a classification model is calculated as follows:

$$\frac{2(P * R)}{P + R}$$

P = the precision

R = the recall of the classification model

(Figure 58: F1-score Calculation)

The micro F1-score indicates how well a model was able to correctly classify the deprived instances since this class if very underrepresented, and the macro F1-score indicates how well a model was able to correctly classify all instances from every class.

Due to processing power and time limitations, the researchers limited the feature reduction tests to simply using the top 50 features in each feature set. The two feature importance methods used to create feature reduction sets were PCA and random forest feature importance. By iterating through all model, feature reduction sets, and included class combinations, the researchers totaled 24 distinct models per covariate and contextual dataset, summing to 48 models total that were trained and tested.

To keep consistent across all experiments done using the Lagos satellite imagery since models across experiments were ensembled in an effort to maximize performance when collaborating with other researchers, this experiment primarily focused on the ability to correctly classify deprived areas using a subset of the contextual and covariate datasets only consisting of the deprived and built-up labeled classes.

For model optimization and tuning, the researchers used Gridsearch CV [34] from the sklearn library and a hyper-parameter space for the MLP [35] and Logistic Regression [19] models. The standard base models for the Gradient Boosting [12] and Random Forest [9] models were used without any parameter tuning since the researchers ran into unresolved grid search issues with those models.

The hyper-parameter spaces and parameters iterated through when looking for the optimal parameter combinations are as follows:

MLP

- Hidden layer sizes: (60, 100, 60), (100, 100, 100), (50, 100, 50)
- Activation: identity, relu, logistic, tanh
- Solver: adam
- Alpha: 0.0001
- Learning Rate: invscaling

Logistic Regression

- Penalty: l1, l2, elasticnet, none
- Dual: True, False
- C: 0.001, 0.01, 1, 10
- Class Weight: dict, balanced, None
- Solver: newton-cg, lbfgs, liblinear, sag, saga

Due to processing power and time limitations, the researchers had to limit the number of parameter variables used in the grid search.

6.2 Contextual Features Results

Figure 59 compares the micro F1-scores for the deprived class and the macro F1-scores across all models trained and tested on a subset of contextual feature dataset including only classes 0 (Built-up) and 1 (Deprived).

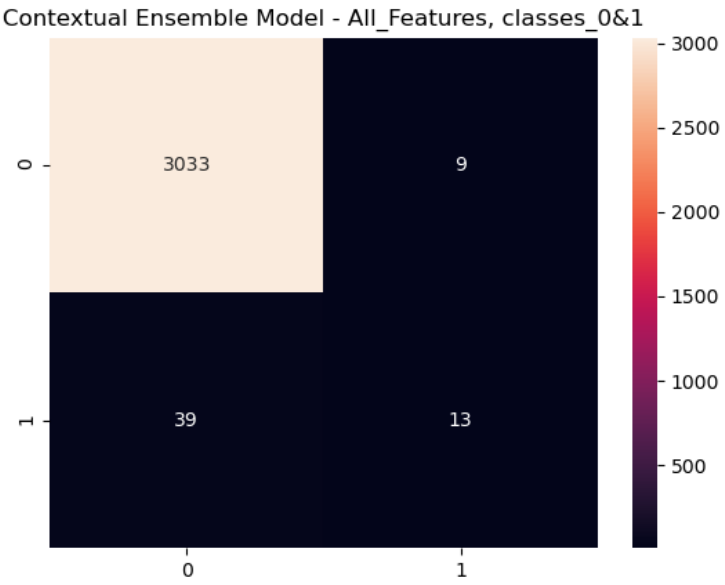
Dataset	Model	Feature Set	Feature Count	Classes	F1 - Class 1 (Deprived)	F1 - Macro
contextual	Ensemble	All_Features	144	classes_0&1	0.35	0.67
contextual	Gradient_Boosting	ADA_Features	50	classes_0&1	0.09	0.54
contextual	Gradient_Boosting	All_Features	144	classes_0&1	0.09	0.54
contextual	Random_Forest	ADA_Features	50	classes_0&1	0.07	0.53
contextual	Gradient_Boosting	Gradient_Boosting_Features	50	classes_0&1	0.07	0.53
contextual	MLP	Gradient_Boosting_Features	50	classes_0&1	0.04	0.51
contextual	MLP	All_Features	144	classes_0&1	0.04	0.51
contextual	MLP	ADA_Features	50	classes_0&1	0.04	0.51
contextual	MLP	Logistic_Features	50	classes_0&1	0.04	0.51
contextual	Logistic_Regression	All_Features	144	classes_0&1	0.04	0.51
contextual	Logistic_Regression	ADA_Features	50	classes_0&1	0.04	0.51
contextual	Logistic_Regression	Logistic_Features	50	classes_0&1	0.04	0.51
contextual	Random_Forest	All_Features	144	classes_0&1	0.04	0.51
contextual	Random_Forest	Logistic_Features	50	classes_0&1	0.04	0.51
contextual	Logistic_Regression	Gradient_Boosting_Features	50	classes_0&1	0.04	0.08
contextual	Gradient_Boosting	Minfo_Features	50	classes_0&1	0.03	0.51
contextual	Gradient_Boosting	Logistic_Features	50	classes_0&1	0.03	0.51
contextual	MLP	Random_Forest_Features	50	classes_0&1	0.03	0.50
contextual	MLP	Minfo_Features	50	classes_0&1	0.00	0.50
contextual	Gradient_Boosting	Random_Forest_Features	50	classes_0&1	0.00	0.49
contextual	Logistic_Regression	Random_Forest_Features	50	classes_0&1	0.00	0.50
contextual	Logistic_Regression	Minfo_Features	50	classes_0&1	0.00	0.50
contextual	Random_Forest	Random_Forest_Features	50	classes_0&1	0.00	0.50
contextual	Random_Forest	Gradient_Boosting_Features	50	classes_0&1	0.00	0.50
contextual	Random_Forest	Minfo_Features	50	classes_0&1	0.00	0.50

(Figure *59*: *Contextual Feature Model Results Using Only the Built-up (0) and Deprived (1) and Classes*)

The model that performed best on the subset of the contextual features dataset containing only instances for the Built-up (0) and Deprived (1) classes was a voting classifier ensemble model [36] model trained on the entire feature set. This ensemble model consisted of nine (9) MLP [15] models, all with the following parameters:

- Hidden layer sizes: (100, 100, 100)
- Activation: tanh
- Solver: adam
- Alpha: 0.0001
- Learning Rate: invscaling

These parameters were found to be the best parameters when using Gridsearch CV [34] on the independent MLP models and thus had the best performing parameters, meaning they would give the best performance for models used in the voting classification ensembled model.



(Figure 60: Confusion Matrix for the best performing model trained on the contextual features dataset containing only instances of the built-up (0) and deprived (1) classes.

Figure 61 compares the micro F1-scores for the deprived class and the macro F1-scores across all models trained and tested on the entire contextual features dataset, including all classes (built-up, deprived, and non-built-up).

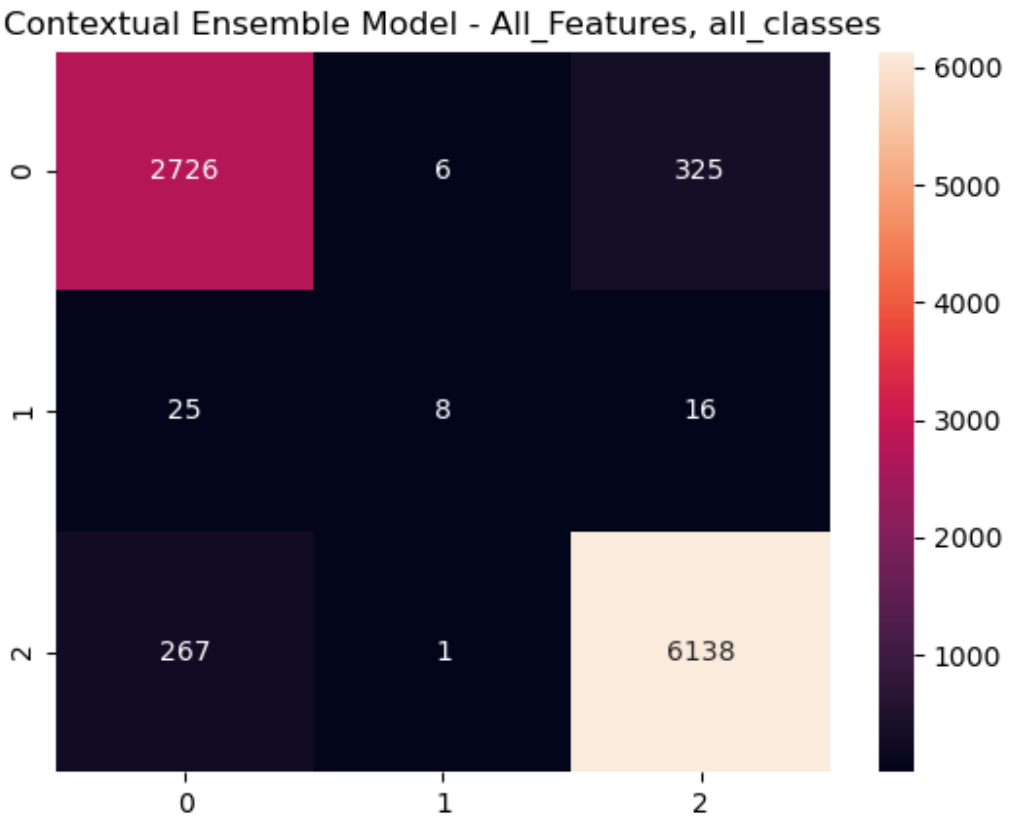
Dataset	Model	Feature Set	Feature Count	Classes	F1 - Class 1 (Deprived)	F1 - Macro
contextual	Ensemble	All_Features	144	all_classes	0.25	0.70
contextual	MLP	All_Features	144	all_classes	0.21	0.67
contextual	Gradient_Boosting	Minfo_Features	50	all_classes	0.18	0.67
contextual	MLP	ADA_Features	50	all_classes	0.10	0.64
contextual	MLP	Logistic_Features	50	all_classes	0.09	0.63
contextual	Random_Forest	ADA_Features	50	all_classes	0.08	0.63
contextual	Gradient_Boosting	ADA_Features	50	all_classes	0.07	0.62
contextual	Gradient_Boosting	All_Features	144	all_classes	0.07	0.63
contextual	Gradient_Boosting	Gradient_Boosting_Features	50	all_classes	0.06	0.61
contextual	Random_Forest	All_Features	144	all_classes	0.04	0.62
contextual	Gradient_Boosting	Logistic_Features	50	all_classes	0.03	0.61
contextual	Logistic_Regression	Random_Forest_Features	50	all_classes	0.00	0.33
contextual	Logistic_Regression	Minfo_Features	50	all_classes	0.00	0.10
contextual	Logistic_Regression	Gradient_Boosting_Features	50	all_classes	0.00	0.32
contextual	MLP	Random_Forest_Features	50	all_classes	0.00	0.38
contextual	MLP	Gradient_Boosting_Features	50	all_classes	0.00	0.44
contextual	MLP	Minfo_Features	50	all_classes	0.00	0.38
contextual	Gradient_Boosting	Random_Forest_Features	50	all_classes	0.00	0.57
contextual	Logistic_Regression	All_Features	144	all_classes	0.00	0.60
contextual	Logistic_Regression	ADA_Features	50	all_classes	0.00	0.59
contextual	Logistic_Regression	Logistic_Features	50	all_classes	0.00	0.59
contextual	Random_Forest	Random_Forest_Features	50	all_classes	0.00	0.44
contextual	Random_Forest	Gradient_Boosting_Features	50	all_classes	0.00	0.42
contextual	Random_Forest	Logistic_Features	50	all_classes	0.00	0.60
contextual	Random_Forest	Minfo_Features	50	all_classes	0.00	0.55

(Figure 61: Contextual Feature Model Results Using All Classes (Built-up, Deprived, Non-built-up)

The model that performed best on the subset of the contextual features dataset containing only instances for the Built-up (0) and Deprived (1) classes was a voting classifier ensemble model [36] model trained on the entire feature set. This ensemble model consisted of nine (9) MLP [35] models, all with the following parameters:

- Hidden layer sizes: (100, 100, 100)
- Activation: tanh
- Solver: adam
- Alpha: 0.0001
- Learning Rate: invscaling

These parameters were found to be the best parameters when using Gridsearch CV [16] on the independent MLP models and thus had the best performing parameters, meaning they would give the best performance for models used in the voting classification ensembled model.



(Figure 62: Confusion Matrix for the best performing model trained on the full contextual features dataset containing all instances of the Built-up (0), Deprived (1), and Non-built-up (2) classes)

With the best model focused on the Built-up (0) and Deprived (1) classes having a micro F1-score for the Deprived classes of 0.35 and macro F1-score of 0.67, and the best model focused on all classes having a micro F1-score of 0.25 and a macro F1-score of 0.70, it is determined that the contextual features are not great indicators of class type.

6.3 Covariate Features Results

Figure 63 compares the micro F1-scores for the deprived class and the macro F1-scores across all models trained and tested on a subset of covariate features dataset including only classes 0 (Built-up) and 1 (Deprived).

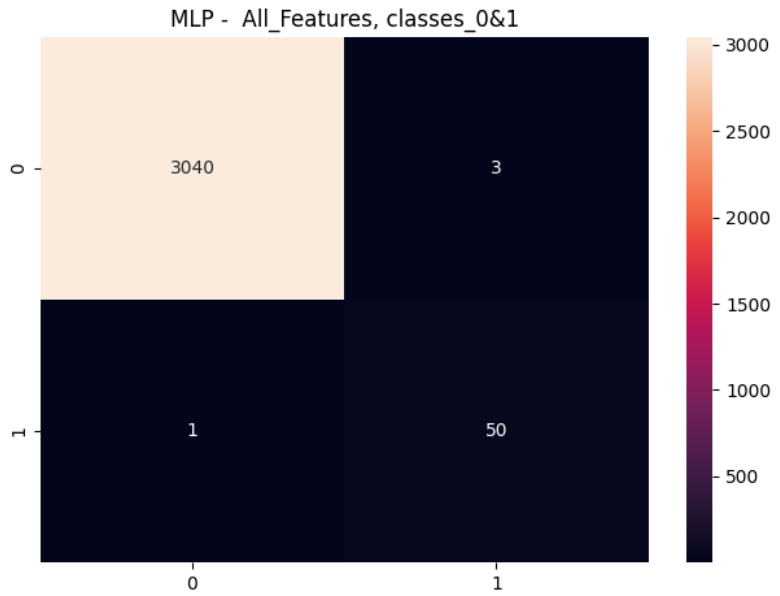
Dataset	Model	Feature Set	Feature Count	Classes	F1 - Class 1 (Deprived)	F1 - Macro
covariate	MLP	All_Features	60	classes_0&1	0.96	0.98
covariate	MLP	Gradient_Boosting_Features	50	classes_0&1	0.96	0.98
covariate	MLP	Logistic_Features	50	classes_0&1	0.94	0.97
covariate	Random_Forest	ADA_Features	50	classes_0&1	0.94	0.97
covariate	MLP	ADA_Features	50	classes_0&1	0.94	0.97
covariate	MLP	Minfo_Features	50	classes_0&1	0.93	0.96
covariate	Logistic_Regression	ADA_Features	50	classes_0&1	0.92	0.96
covariate	Random_Forest	Minfo_Features	50	classes_0&1	0.92	0.96
covariate	Random_Forest	All_Features	60	classes_0&1	0.92	0.96
covariate	Random_Forest	Gradient_Boosting_Features	50	classes_0&1	0.92	0.96
covariate	Random_Forest	Logistic_Features	50	classes_0&1	0.91	0.95
covariate	Logistic_Regression	Gradient_Boosting_Features	50	classes_0&1	0.90	0.95
covariate	Logistic_Regression	Minfo_Features	50	classes_0&1	0.90	0.95
covariate	Logistic_Regression	All_Features	60	classes_0&1	0.89	0.95
covariate	Gradient_Boosting	ADA_Features	50	classes_0&1	0.89	0.94
covariate	Gradient_Boosting	All_Features	60	classes_0&1	0.89	0.94
covariate	Gradient_Boosting	Gradient_Boosting_Features	50	classes_0&1	0.88	0.94
covariate	Gradient_Boosting	Minfo_Features	50	classes_0&1	0.86	0.93
covariate	Gradient_Boosting	Logistic_Features	50	classes_0&1	0.85	0.92
covariate	Logistic_Regression	Logistic_Features	50	classes_0&1	0.82	0.91

(Figure 63: Covariate Feature Model Results Using Only the Built-up (0) and Deprived (1) and Classes)

The model that performed best on the subset of the covariate features dataset containing only instances for the built-up (0) deprived (1) classes was an MLP model trained on the entire feature set. Through the use of Gridsearch CV and the parameter space mentioned in section 5.1, the following parameters were found to perform best for our top model highlighted above:

- Hidden layer sizes: (100, 100, 100)
- Activation: tanh
- Solver: adam
- Alpha: 0.0001
- Learning Rate: invscaling

This indicates that higher hidden layer sizes paired with the tanh activation outperforms lower hidden sizes and the other activations (identity, relu, logistic) for the dataset used to train and test these models.



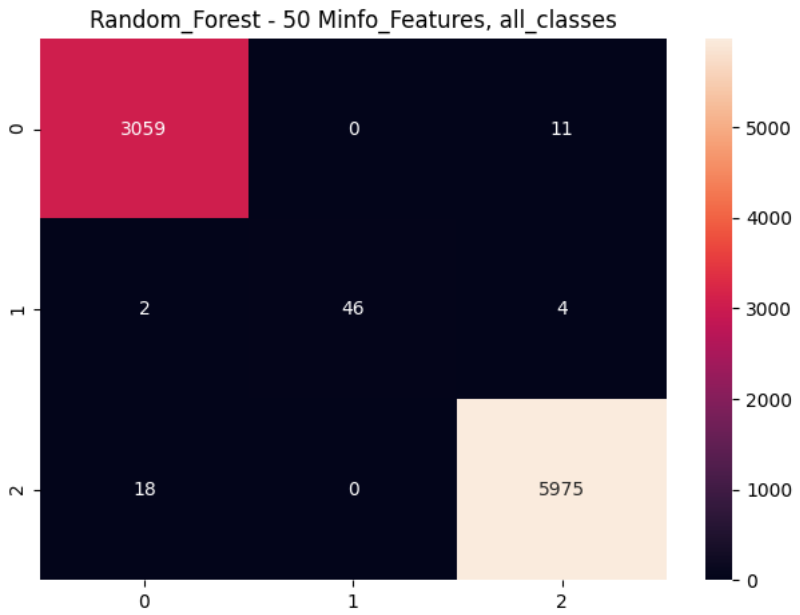
(Figure 64 – Confusion Matrix for the best performing model trained on the covariate features dataset containing only instances of the Built-up (0) and Deprived (1) classes)

Figure 65 compares the micro F1-scores for the deprived class and the macro F1-scores across all models trained and tested on the entire covariate features dataset, including all classes (Deprived, Built-up, and Non-built-up).

Dataset	Model	Feature Set	Feature Count	Classes	F1 - Class 1 (Deprived)	F1 - Macro
covariate	Random_Forest	Minfo_Features	50	all_classes	0.94	0.98
covariate	Random_Forest	All_Features	60	all_classes	0.91	0.97
covariate	Random_Forest	Gradient_Boosting_Features	50	all_classes	0.91	0.97
covariate	Random_Forest	Logistic_Features	50	all_classes	0.91	0.96
covariate	Random_Forest	ADA_Features	50	all_classes	0.90	0.96
covariate	MLP	Gradient_Boosting_Features	50	all_classes	0.90	0.96
covariate	MLP	ADA_Features	50	all_classes	0.89	0.96
covariate	MLP	Logistic_Features	50	all_classes	0.89	0.96
covariate	MLP	All_Features	60	all_classes	0.88	0.96
covariate	MLP	Minfo_Features	50	all_classes	0.88	0.96
covariate	Logistic_Regression	Logistic_Features	50	all_classes	0.80	0.92
covariate	Logistic_Regression	All_Features	60	all_classes	0.78	0.91
covariate	Logistic_Regression	Gradient_Boosting_Features	50	all_classes	0.78	0.91
covariate	Gradient_Boosting	ADA_Features	50	all_classes	0.78	0.92
covariate	Gradient_Boosting	Logistic_Features	50	all_classes	0.78	0.92
covariate	Logistic_Regression	ADA_Features	50	all_classes	0.78	0.91
covariate	Logistic_Regression	Minfo_Features	50	all_classes	0.78	0.90
covariate	Gradient_Boosting	Gradient_Boosting_Features	50	all_classes	0.74	0.91
covariate	Gradient_Boosting	All_Features	60	all_classes	0.73	0.90
covariate	Gradient_Boosting	Minfo_Features	50	all_classes	0.71	0.90

(Figure 65: Covariate Feature Model Results Using All Classes (Built-up, Deprived, Non-built-up))

The two models that performed best on the full contextual features dataset containing all instances for every class were both Random Forest models, one trained on the entire feature set and the other trained on the top 50 features in the mutual information feature importance feature set.



(Figure 66: Confusion Matrix for the best performing model trained on the full covariate features dataset containing all instances of the Built-up (0), Deprived (1), and Non-built-up (2) classes)

With the best model focused on the Built-up (0) and Deprived (1) classes having a micro F1-score for the Deprived classes of 0.96 and macro F1-score of 0.98, and the best model focused on all classes having a micro F1-score of 0.95 and a macro F1-score of 0.98, it is determined that the covariate features are extremely great indicators of class type.

7 Discussion

Unlike previous work done on utilizing feature importance methods to identify important feature in classification models[3], the researchers implemented five distinct methodologies (mutual information, random forest, logistic, gradient boosting, and AdaBoost) for feature importance. These feature methods were then combined with hyperparameter tuning to make the models even more robust. Unlike previous studies utilizing contextual features[2], the researchers did not find the contextual features significant. The major issue that the researchers encountered was attempting to find ways to make the contextual features useful in identifying 'Deprived' or 'Built-up' areas. Regardless of the feature importance methods, model choice, or hyperparameter tuning, the contextual features were not helpful. In future studies, other feature extraction methods for contextual features should be explored to confirm if these features are beneficial in identifying the target variable. Through statistical analysis, it was confirmed that the top 21 covariate features were useful in identifying the 'Deprived' and 'Built-up' areas.

In future analysis, more labeled data for the 'Deprived' area can be incorporated to address the issue of class imbalance. Furthermore, analysis can be conducted where outliers are removed to see if any of the statistical results change. Different statistical analysis can be applied to assessing the significance of the covariate features. Applying transformations to the data can be useful in deriving a normal distribution to fit the ANOVA assumption (**Figure 12**).

8 Conclusion

This project aimed to apply various deep learning, traditional machine learning classification techniques, and statistical analysis on low resolution and free satellite imagery as well as on calculated contextual and covariate features to detect deprived areas on a 10m^2 level. This report specifically focused on the use of traditional machine learning classification techniques on the contextual and covariate features.

Two distinct datasets were tested in this experiment, the contextual features dataset and the covariate features dataset. Each test was split into two subcategories: testing on the full dataset including all classes (Built-up, Deprived, and Non-built-up) and testing on a subset of each dataset only consisting of the Built-Up and Deprived classes. The researchers focused on the dataset subsets in order to keep data consistent with researchers working on the deep-learning portion of the project.

Figure 67 represents the best models per dataset and subset of classes including the micro F1-score for the Deprived class and the macro F1-score for the entire model.

Dataset	Model	Feature Set	Feature Count	Classes	F1 - Class 1 (Deprived)	F1 - Macro
covariate	MLP	All_Features	60	classes_0&1	0.96	0.98
covariate	Random_Forest	Minfo_Features	50	all_classes	0.94	0.98
contextual	Gradient_Boosting	All_Features	144	classes_0&1	0.11	0.55
contextual	MLP	All_Features	144	all_classes	0.21	0.67

(Figure 67: Best Models Results)

As can be seen in the table, the covariate features were extremely great indicators of class type while the contextual features were not. It was found that in most cases, the full feature sets outperformed the top 50 features per feature importance set indicating that there is a loss of useful information when conducting feature reduction.

9 Bibliography

[1] Kabaria, Carloine Wanjiku, "Integrated Deprived Area Mapping System (IDEAMAPS) Newtwork", *UK Research and Innovation,* 2021 https://gtr.ukri.org/project/0D56FCFB-5673-4AF2-8091-0A1710B6C3F4
[2] Chao, Steven, "Evaluating the Ability to Use Contextual Features Derived from Multi-Scale Satellite Imagery to Map Spatial Patterns of Urban Attributes and Population Distributions", *Remote Sensing,* 2021
[3] Jauhianinen, Saarela, 'Comparison of feature importance measures as explanations for classification models', https://link.springer.com/article/10.1007/s42452-021-04148-9
[4] sci-kit learn, StandardScalar, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[5] sci-kit learn, SelectKbest, https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
[6] sci-kit learn, mutual_info_classif, https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html
[7] Grassberger, Stögbauer, and Kraskov, 'Estimating mutal information, https://journals.aps.org/pre/abstract/10.1103/PhysRevE.69.066138
[8] Aznar, 'What is Mutual Information', https://quantdare.com/what-is-mutual-information/
[9]sci-kit learn, *Random Forest Classifie*r, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[10] sci-kit learn, 'Feature Importance with a Forest of Trees', https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
[11] sci-kit learn, *Logistic Regression Classifie*r, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[12] sci-kit learn, *Gradient Boosting Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[13] sci-kit learn, *AdaBoost Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier.feature_importances_
[14] sci-kit learn, *Stratified-KFold*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
[15] Techopedia, *Multi-Layer Perceptron (MLP),* https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp#:~:text=A%20multilayer%20perceptron%20is%20a,as%20a%20supervised%20learning%20technique.
[16] Hagan, Martin T., *Neural Network Design 2nd Edition,* https://hagan.okstate.edu/NNDesign.pdf
[17] Donges, Niklas, *Random Forest Algorithm: A Complete Guide*, https://builtin.com/data-science/random-forest-algorithm#how
[18] Hoare, Jake, "Gradient Boosting Explained – The Coolest Kid on The Machine Learning Block", *DisplayR*, https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/#:~:text=Gradient%20boosting%20is%20a%20type,order%20to%20minimize%20the%20error.

[19] Lawton, George, “Logistic Regression”, *TechTarget*, https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression#:~:text=Logistic%20regression%20is%20a%20statistical,or%20more%20existing%20independent%20variables.
[20] Varsheni, Shri, “perform Logistic Regression with Pytorch Seamlessly”, *Analytics Vidya,* https://www.analyticsvidhya.com/blog/2021/07/perform-logistic-regression-with-pytorch-seamlessly/, *2021*
[21] Technology Networks Informatics, “One-Way vs Two-Way ANOVA: Differences, ssumptions and Hypotheses’, https://www.technologynetworks.com/informatics/articles/one-way-vs-two-way-anova-definition-differences-assumptions-and-hypotheses-306553
[22] Engineering Statistics Handbook, ‘Kolmogorov-Smirnov Goodness of Fit Test’, https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
[23] Engieering Statistics Handbook, ‘Levene Test for Equlaity of Variance’ https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm
[24] Laerd Statistics, ‘One-way ANOVA (cont’d)’, https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide-3.php
[25] ScienceDirect, ‘Kruskal Wallis Test’, https://www.sciencedirect.com/topics/medicine-and-dentistry/kruskal-wallis-test
[26] Laerd Statistics, ‘Kruskal-Wallis H Test using SPSS’, https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php
[27] Penn State Eberly Colle of Science, ‘The Chi-Square Test of Independence’, https://online.stat.psu.edu/stat500/lesson/8/8.1
[28]sci-kit learn, *StratifiedKfolds*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
[29] Scipy, ktest, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html
[30] Scipy, levene, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.levene.html
[31]scipy, Kruskal, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html
[32] scipy, chi_contingency, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html
[33] Edpresso Team, "What is the F1-score?", *Educative*, https://www.educative.io/edpresso/what-is-the-f1-score
[34] sci-kit learn, *Gridsearch CV*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[35] sci-kit learn, *MLPClassifier, https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html*
[36] sci-kit learn, *Voting Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

10 Appendix

Link to researcher’s Github
https://github.com/mojahid/Mapping-Deprived-Areas-Using-Deep-Neural-Networks

Band Name	Description	Data type	Data Domain	Link of original data
fs_dist_fs_2020	Lagos Distance to Financial Services (HOTOSM Export)	numerical	Facilities & services	https://data.humdata.org/dataset/hotosm_gha_financial_services
fs_dist_hf_2019	Lagos Distance to health facilities (HOTOSM Export)	numerical	Facilities & services	https://doi.org/10.6084/m9.figshare.c.4399445.v1
fs_dist_hf1_2020	Lagos Distance to health facilities (Geopode)	numerical	Facilities & services	https://nga.geopode.world/geometry_export/0/?predefined&lgas=&format=FileGDB&layers=&states=ALL
fs_dist_market_2020	Lagos Distance to markets (Geopode)	numerical	Facilities & services	https://nga.geopode.world/geometry_export/0/?predefined&lgas=&format=FileGDB&layers=&states=ALL
fs_dist_mosques_2017	Lagos Distance to mosques (Geopode)	numerical	Facilities & services	https://nga.geopode.world/geometry_export/0/?predefined&lgas=&format=FileGDB&layers=&states=ALL
fs_dist_school_2020	Lagos Distance to Education Facilities (HOTOSM Export)	numerical	Facilities & services	https://data.humdata.org/dataset/hotosm_nga_education_facilities
fs_dist_school1_2018	Lagos Distance to schools (Geopode)	numerical	Facilities & services	https://nga.geopode.world/geometry_export/0/?predefined&lgas=&format=FileGDB&layers=&states=ALL
fs_dist_well_2018	Lagos Distance to wells (Geopode)	numerical	Facilities & services	https://nga.geopode.world/geometry_export/0/?predefined&lgas=&format=FileGDB&layers=&states=ALL
fs_electric_dist_2020	Lagos electrical distribution Grid Map. 1 = presence of MV grid infrastructure; 0 = no presence	categorical	Infrastructure	https://data.humdata.org/dataset/electricaldistributiongridmaps
in_dist_rd_2016	Lagos Distance to	numerical	Infrastructure	https://www.worldpop.org/geodata/summary?id=17457

	OSM major roads (2016)			
in_dist_rd_intersect_2016	Distance to OSM major road intersections (2016)	numerical	Infrastructure	https://www.worldpop.org/geodata/summary?id=17706
in_dist_waterway_2016	Distance to OSM major waterways (2016)	numerical	Infrastructure	https://www.worldpop.org/geodata/summary?id=17955
in_night_light_2016	VIIRS night-time lights of Lagos in 2016	numerical	Infrastructure	https://www.worldpop.org/geodata/summary?id=18693
ph_base_water_2010	Lagos baseline water stress score. collected and calculated in administrative unit	numerical	Physical hazards & assets	https://datasets.wri.org/dataset/aqueduct-global-maps-21-data
ph_bio_dvst_2015	Lagos Biodiversity (mean species abundance)	numerical	Physical hazards & assets	https://www.globio.info/globio-data-downloads
ph_climate_risk_2020	Average annual climate risk	numerical	Physical hazards & assets	https://data.chc.ucsb.edu/products/CHIRPS-2.0/africa_monthly/tifs/
ph_dist_aq_veg_2015	Distance to ESA-CCI-LC aquatic vegetation area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_art_surface_2015	Distance to ESA-CCI-LC artificial surface edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_bare_2015	Distance to ESA-CCI-LC bare area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_cultivated_2015	Distance to ESA-CCI-LC cultivated area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926

ph_dist_herb_2015	Distance to ESA-CCI-LC herbaceous area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_inland_water_2018	Distance to ESA-CCI-LC inland water (2000-2012)	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=24171
ph_dist_open_coast_2020	Distance to open-water coastline (2000-2020)	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=23922
ph_dist_riv_network_2007	Lagos Distance to River Network	numerical	Physical hazards & assets	https://hydrosheds.org/downloads
ph_dist_shrub_2015	Distance to ESA-CCI-LC shrub area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_sparse_veg_2015	Distance to ESA-CCI-LC sparse vegetation area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_dist_woody_tree_2015	Distance to ESA-CCI-LC woody-tree area edges 2015	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=22926
ph_gdmhz_2005	Lagos 2000 Global Multihazard frequency & distribution. *No original data in Lagos, but since it's a cross-city variable, whole lagos was identified as 0	numerical	Physical hazards & assets	https://sedac.ciesin.columbia.edu/data/set/ndh-multihazard-frequency-distribution/data-download
ph_grd_water_2000	Lagos ground water stress score. collected and	numerical	Physical hazards & assets	https://datasets.wri.org/dataset/aqueduct-global-maps-21-data

	calculated in administratic ve unit			
ph_hzd_index_2011	Lagos Global estimated risk index for multiple hazards. risk index from 1 (low) to 5 (extreme); 256 = no data value	categorica l	Physical hazards & assets	https://preview.grid.unep.ch/index.php?preview=data&events=multiple&evcat=1&lang=eng
ph_land_c1_2019	Lagos Land cover from Copernicius Global Land Service. 20 = Shrubs; 30 = Herbaceous vegetation; 40 = Cultivated and managed vegetation/a griculture; 50 = Urban / built up; 60 = Bare / sparse vegetation; 80 = Permanent water bodies; 90 = Herbaceous wetland; 112 = Closed forest, evergreen,br oad leaf; 126 = Open forest, unknown; 200 = Open sea; 256 means no data	categorica l	Physical hazards & assets	https://land.copernicus.eu/global/content/annual-100m-global-land-cover-maps-available
ph_land_c2_2020	Lagos Land cover from GlobeLand3	categorica l	Physical hazards & assets	http://www.globeland30.org/defaults_en.html?type=data&src=/Scripts/map/defaults/En/browse_en.html&head=browse

	0 - 2020. 10 = Cultivated land; 20 = Forest; 30 = Grassland; 40 = Shrubland; 50 = Wetland; 60 = Water bodies; 70 = Tundra; 80 = Atrificial Surfaces; 90 = Bare land ; 100 = Permanent snow and ice; 256 = no data			
ph_max_tem_2019	Lagos 2019 maximun ground temperature (?)	numerical	Physical hazards & assets	https://clim-engine.appspot.com/climateEngine
ph_ndvi_2019	Lagos 2019 max ndvi 30m, resampled to IDEAMAPS 100m grid (Climate Engine)	numerical	Physical hazards & assets	https://clim-engine.appspot.com/climateEngine
ph_pm25_2016	Lagos air pollution (PM2.5) - 2016	numerical	Physical hazards & assets	https://sedac.ciesin.columbia.edu/data/set/sdei-global-annual-gwr-pm2-5-modis-misr-seawifs-aod/data-download
ph_slope_2000	Nairobi SRTM-based slope (2000)	numerical	Physical hazards & assets	https://www.worldpop.org/geodata/summary?id=23175
po_pop_fb_2018	Lagos Estimated Population Count 2018 (HRSL-Facebook)	numerical	Population counts	https://data.humdata.org/dataset/highresolutionpopulationdensitymaps-nga
po_pop_un_2020	Lagos Estimated Population Count 2020 (WorldPop-	numerical	Population counts	https://www.worldpop.org/geodata/listing?id=79

	UNadj-constrained)			
ses_an_visits_2016	Lagos precentage of pregnant women who had 4+ antenatal visits	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_child_stunted_2018	Lagos precentage of children who are stunted	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_dpt3_2018	Nairobi precentage of children receiving DPT3 vaccine	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_hf_delivery_2018	Lagos precentage of pregnant women delivering at a health facility	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_impr_water_src_2016	Lagos precentage of population living in households using an improved water source	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_ITN_2016	Lagos precentage of population with access to an insecticide-treated mosquito net (ITN)	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_m_lit_2018	Lagos precentage of men who are literate	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_measles_2018	Lagos precentage	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS

	of children receiving measles vaccine			
ses_odef_2018	Lagos precentage of population living in households using open defecation	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_pfpr_2017	Lagos Plasmodium falciparum parasite rate in 2-10 year olds (2017)	numerical	SES (HH)	https://malariaatlas.org/explorer/#/
ses_preg_2017	Lagos pregnancies	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_unmet_need_2018	Lagos precentage of unmet need for family planning	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_w_anemia_2018	Lagos precentage of women with any anemia	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
ses_w_lit_2018	Lagos precentage of women who are literate	numerical	SES (HH)	https://spatialdata.dhsprogram.com/modeled-surfaces/#survey=NG|2018|DHS
sh_dist_conflict_2020	Lagos distance to armed conflict & event locations	numerical	Social hazards & assets	https://acleddata.com/#/dashboard
sh_dist_mnr_pofw_2019	Distance to minority religious facility (compared to city average). Points derived from national dataset that	numerical	Social hazards & assets	http://download.geofabrik.de/

	was cut to Lagos city extent.			
sh_dist_pofw_2019	Distance to place of worship. Dervied from national dataset that was clipped to IDEAMAPS Lagos city extent.	numerical	Social hazards & assets	http://download.geofabrik.de/
sh_ethno_den_2020	Number of ethno-linguistic groups in 100m cell. Derived from a global dataset and clipped to IDEAMAPS Lagos city extent.	numerical	Social hazards & assets	https://go-imb.opendata.arcgis.com/datasets/pgopen?layer=0
sh_pol_relev_ethnic_gr_2019	Politically relevant ethnic groups. Politically relevant ethnic groups from the EPR-Core 2019 dataset (identifies all politically relevant ethnic groups and their access to state power in every country of the world from 1946 to 2017). GeoEPR assignes	categorical	Social hazards & assets	https://icr.ethz.ch/data/epr/

	every politically relevant group one of six settlement patterns and, if possible, provides polygons describing their location on a digital map. Projected to GCS_WGS_1984 and clipped to Lagos city extent. 1= presence of politically relevant ethinic group. 0= absense of politically relevant ethnic group.			
uu_bld_count_2020	Lagos Building Count	numerical	Unplanned urbanization	ftp://ftp.worldpop.org.uk/repo/wopr/_MULT/buildings/v1.1/
uu_bld_den_2020	Lagos Building Density	numerical	Unplanned urbanization	ftp://ftp.worldpop.org.uk/repo/wopr/_MULT/buildings/v1.1/
ho_impr_housing_2015	Lagos Improved Housing 2015 Prevalence	numerical	Housing(HO)	https://malariaatlas.org/explorer/#/
uu_urb_bldg_2018	Lagos Urban Building Footprint. Urban/rural classification based on building patterns in that area. 1	categorical	Unplanned urbanization	ftp://ftp.worldpop.org.uk/repo/wopr/_MULT/buildings/v1.1/

	indicates urban, 0 indicates rural, -1 indicates no data.			

(Figure 68: Covariate Feature Information)