THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Data Science Program

Capstone Report
Jake's Section
Spring 2022

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

Jake Lieberfarb

supervised by
Amir Jafari

Table of Contents

1 What I worked on

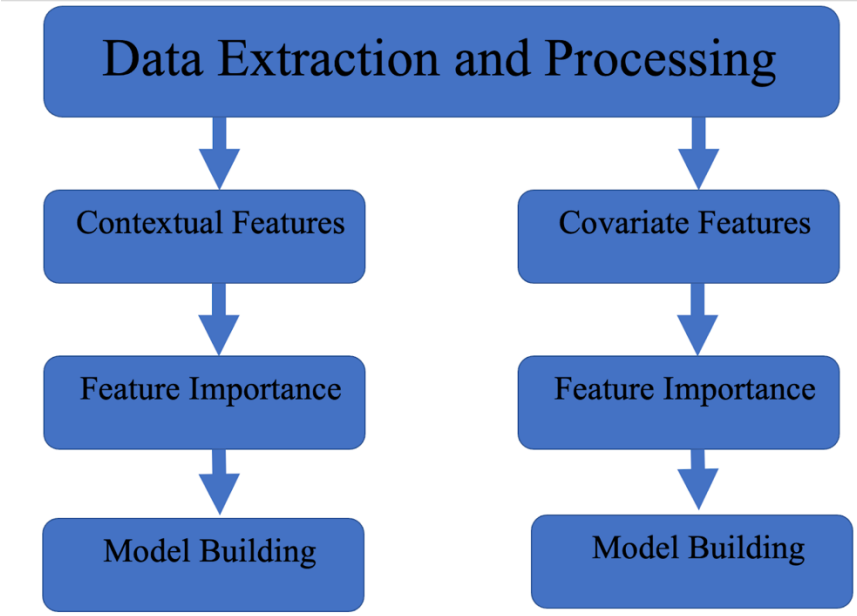
For the Capstone, I worked on processing and extracting feature importance for the contextual and covariate features. I created a pipeline for both feature methods that ranked the 144 contextual features and 60 covariate features based on the five feature importance methods: Mutual Information, Random Forest, Logistic Regression, Gradient Boosting, and Adaptive Boosting. I found this capstone project very interesting as I got to work with real world data and apply the machine learning models and statistical analysis that I had learned during my time the Data Science Program.

All my work can be found in the Contextual_and_Covariate_Features_Modeling folder (link) on Github. Below is the solution and methodology section I made for my Capstone.

2 Solution and Methodology

2.1 Overview of Section

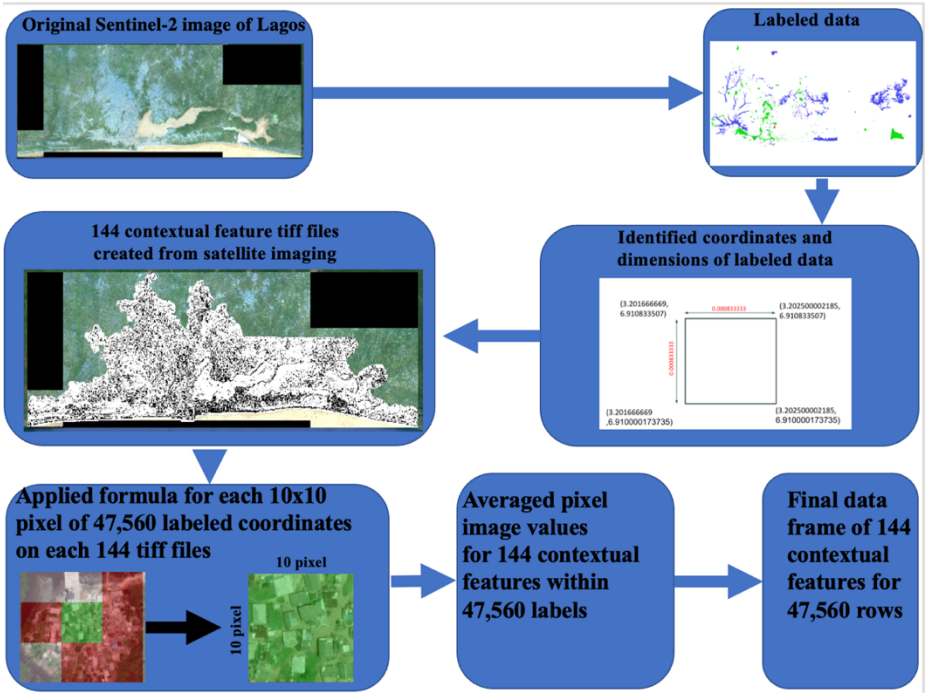
The solution and methodology sections were split up in two phases: contextual features and covariate features. The two sections underwent the same pipeline after the contextual and covariate features were processed.



(Figure 1: 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.

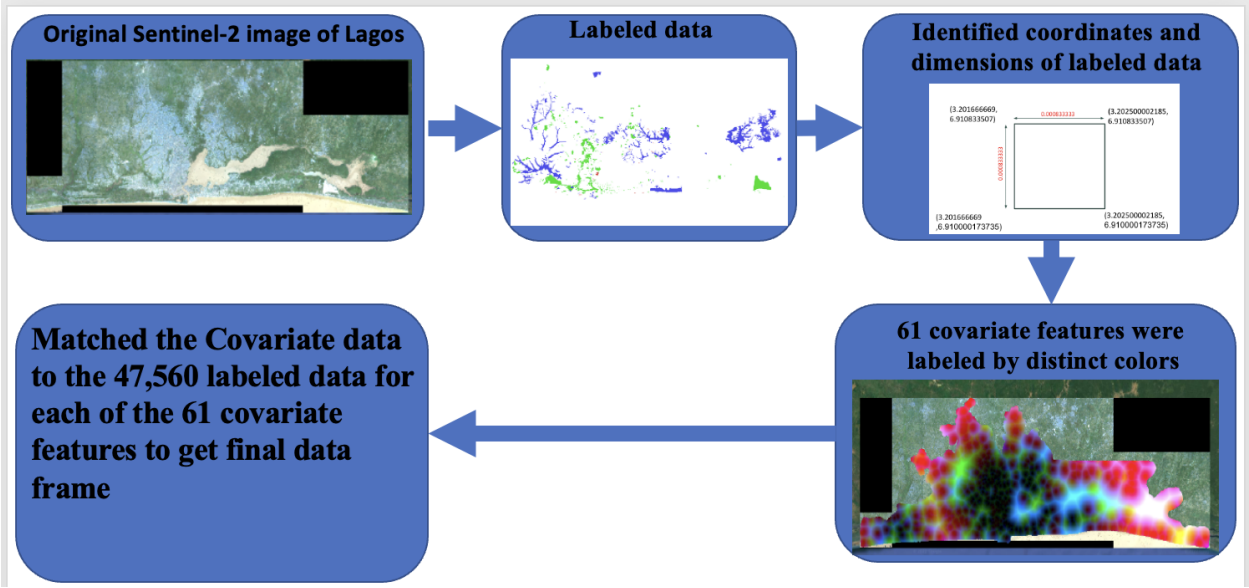
2.2.1 Data Extraction and Processing on Contextual Features



(Figure 2: 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.

2.2.2 Data Extraction and Processing on Covariate Features



(Figure 3: 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.

2.3 *Feature Standardization*

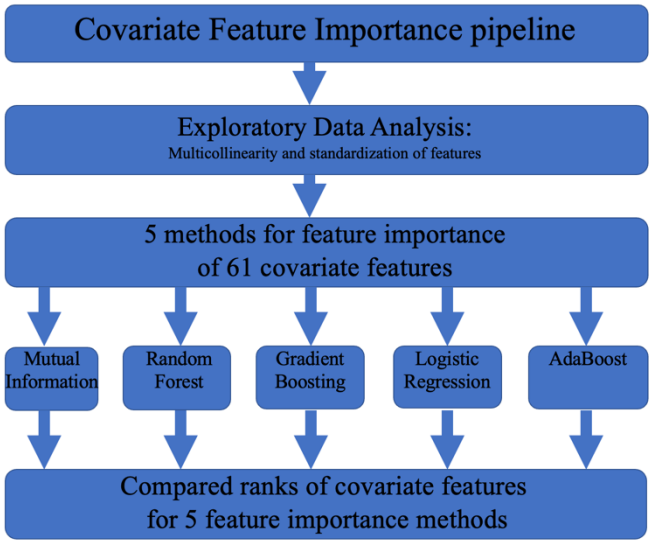
As part of preparing the contextual and covariate data for analysis. The features were standardized using *StandardScaler[1]*.

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

(Figure 4: Standardization Formula Used on Contextual and Covariate Features)

2.4 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 5: Pipeline for Feature Importance on Contextual and Covariate Datasets)

2.4.1 Mutual Information Extraction

To implement mutual information feature importance on the contextual and covariate features, the tools of *SelectKBest[2]* and *mutual_info_classif* [3]. 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[4]. Mutual information is closely related to entropy and provides results from the range of zero to 1[5].

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

(Figure 6: Entropy Calculation)

2.4.2 Random Forest Feature Importance

For Random Forest feature importance. The function *RandomForestClassifier*[6] 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[7].

2.4.3 Logistic Regression Feature Importance

To identify feature importance in logistic regression, full models were constructed for the contextual and covariate features. The function *LogisticRegression* [8] 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.

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

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

(Figure 7: Logistic Formula for Contextual and Covariate Models)

2.4.4 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) [9].

2.4.5 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 [10].

2.5 Statistical Assessments

2.5.1 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 [11]:

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

(Figure 8: 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 0.02.

2.5.2 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. [12]

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

(Figure 9: Kolmogorov-Smirnov Test Hypothesis test and Test Statistic Calculation)

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

2.5.3 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.[13]

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

(Figure 10: Levene Hypothesis test)

2.5.4 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 [14]. Mean rank refers to the average of ranks for observations within each. Furthermore, while previous research has recommended different non-parametric testing

procedures [15], the researchers decided to go with Kruskal Wallis H-test on the data for its ability to work with data of unequal variance.[16]

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

(Figure 11: 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.

2.5.5 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[17]:

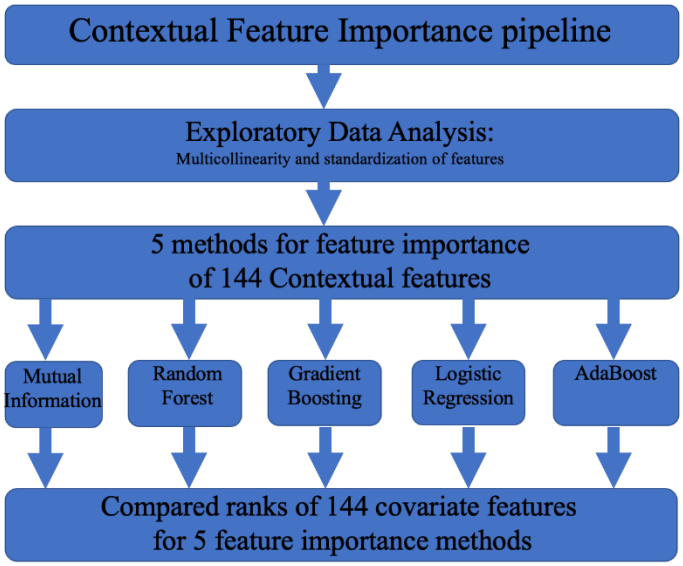
$$H_o: The\ categorical\ features\ are\ independent$$
$$H_a: The\ categorical\ features\ are\ not\ independent$$
$$X^{2^*} = \sum_{i=1} \frac{(O_i - E_i)^2}{E_i}$$

(Figure 12: 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.

2.6 Contextual Feature Analysis Overview

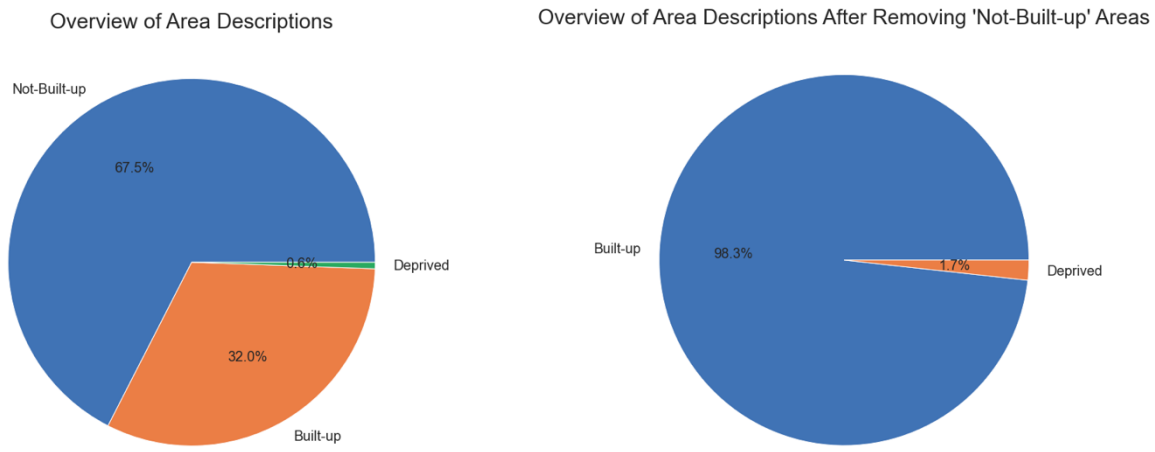
The contextual features were analyzed through exploratory data analysis and then feature importance. Once the contextual features were processed, exploratory data analysis was conducted on the entire data frame to identify any instances of multicollinearity. Also, the data was split on train/validation/test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



(Figure 13: pipeline for Contextual Features)

2.6.1 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.

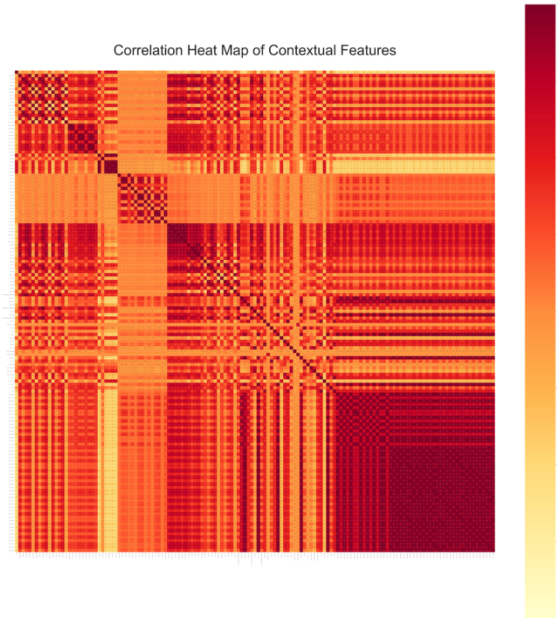


(Figure 14: Overview of Contextual Features)

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.

2.6.2 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.

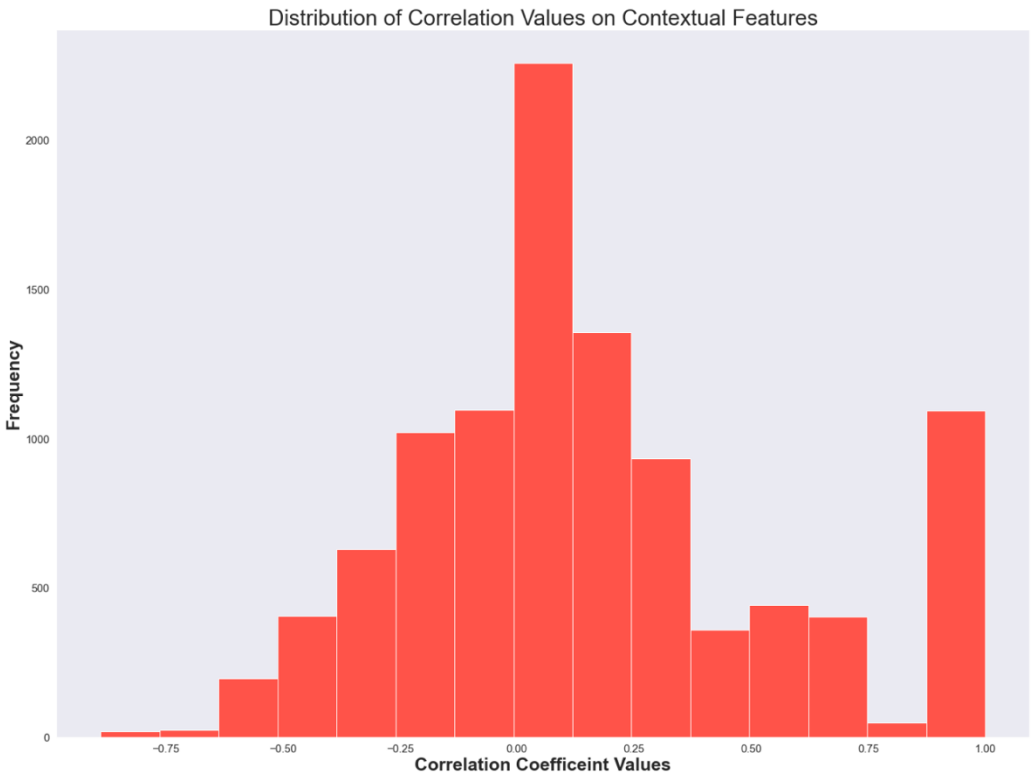


(Figure 15: Correlation Heat Map of Contextual Features)

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.

2.6.3 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.

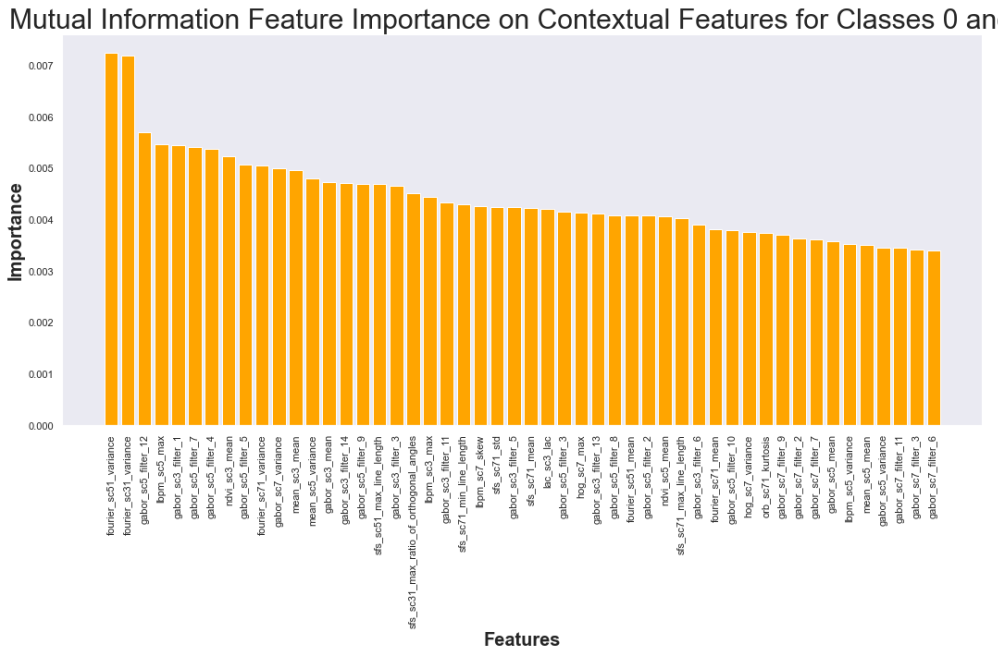


(Figure 16: Histogram of Correlation Coefficient of Contextual Features)

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.

2.6.4 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 17: 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'.

2.6.5 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()[18]

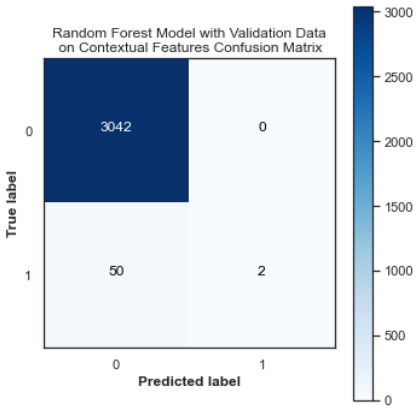

```
Random Forest Model with Test Data
 on Contextual Features Classification Report
              precision    recall  f1-score   support

           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 18: 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.

2.6.6 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

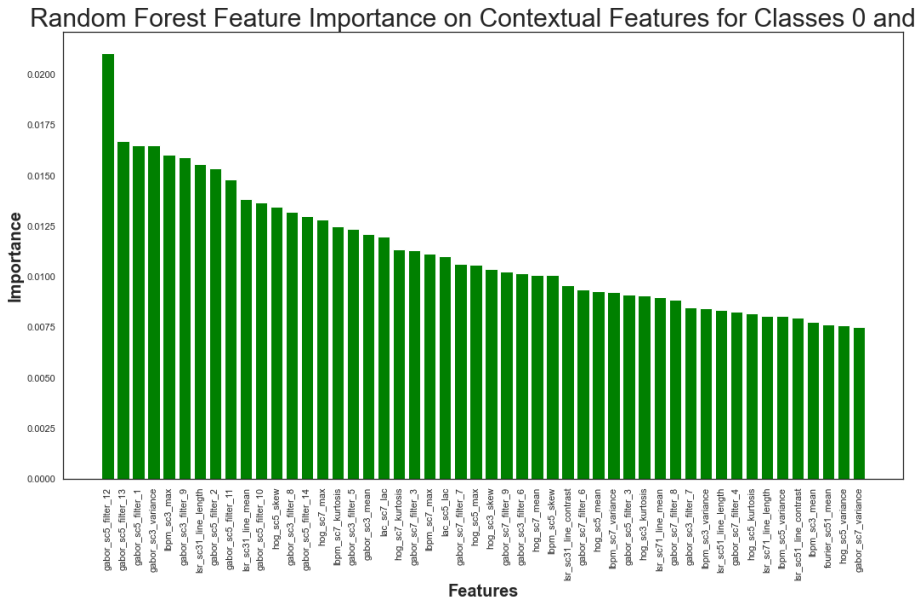
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	1.00	0.04	0.07	52
accuracy			0.98	3094
macro avg	0.99	0.52	0.53	3094
weighted avg	0.98	0.98	0.98	3094

(Figure 19: 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.

2.6.7 Contextual Feature Importance: Random Forest Feature Importance

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



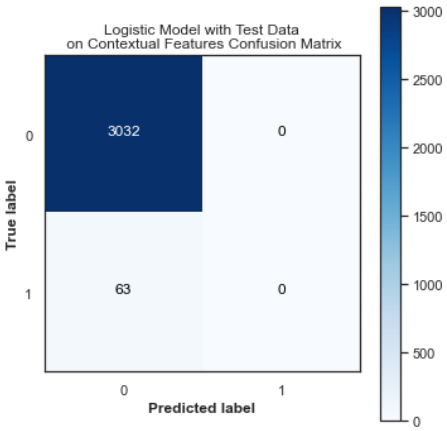
(Figure 20: Random Forest Feature Importance)

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'.

2.6.8 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

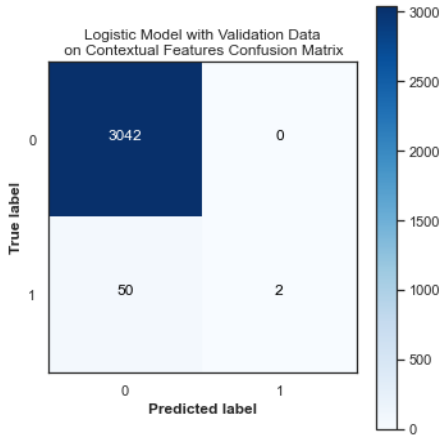
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 21: 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-02.

2.6.9 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

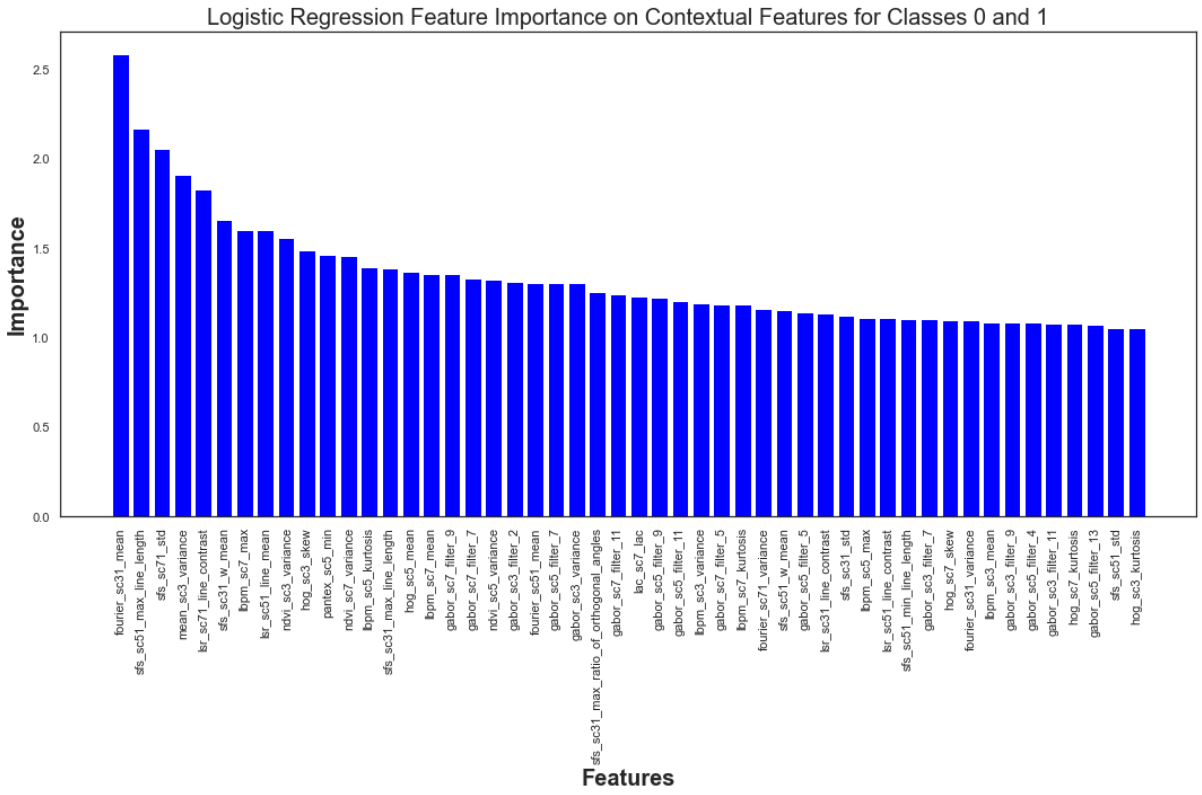
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	1.00	0.04	0.07	52
accuracy			0.98	3094
macro avg	0.99	0.52	0.53	3094
weighted avg	0.98	0.98	0.98	3094

(Figure 22: 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.

2.6.10 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.



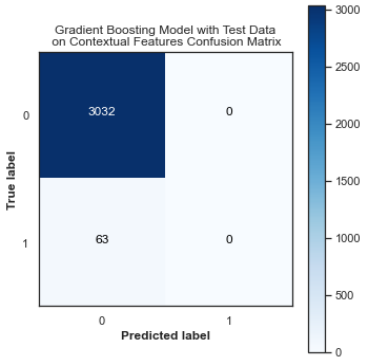
(Figure 23: Logistic Regression Feature Importance on Contextual Features)

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

2.6.11 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

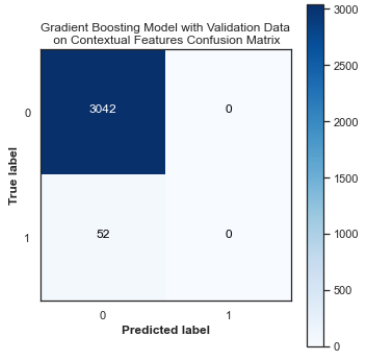
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 24: 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.

2.6.12 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

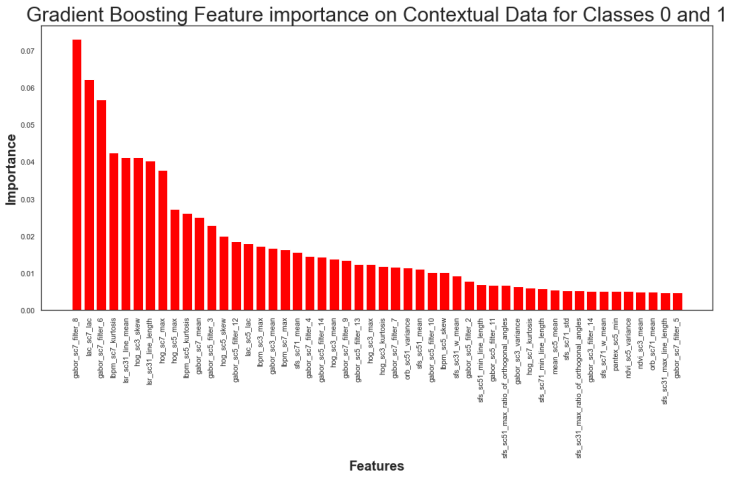
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	0.00	0.00	0.00	52
accuracy			0.98	3094
macro avg	0.49	0.50	0.50	3094
weighted avg	0.97	0.98	0.97	3094

(Figure 25: Gradient Boosting Validation Results Contextual Features)

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

2.6.13 Contextual Feature Importance: Gradient Boosting Feature Importance

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



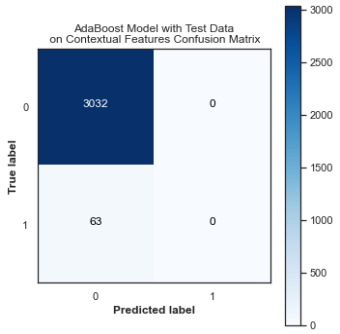
(Figure 26: Gradient Boosting Feature Importance on Contextual Features)

The contextual feature that performed the best was 'gabor_sc7_filter_8'. 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.

2.5.14 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

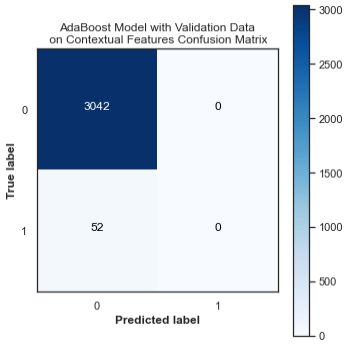
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3032
1	0.00	0.00	0.00	63
accuracy			0.98	3095
macro avg	0.49	0.50	0.49	3095
weighted avg	0.96	0.98	0.97	3095

(Figure 27: 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.

2.6.15 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

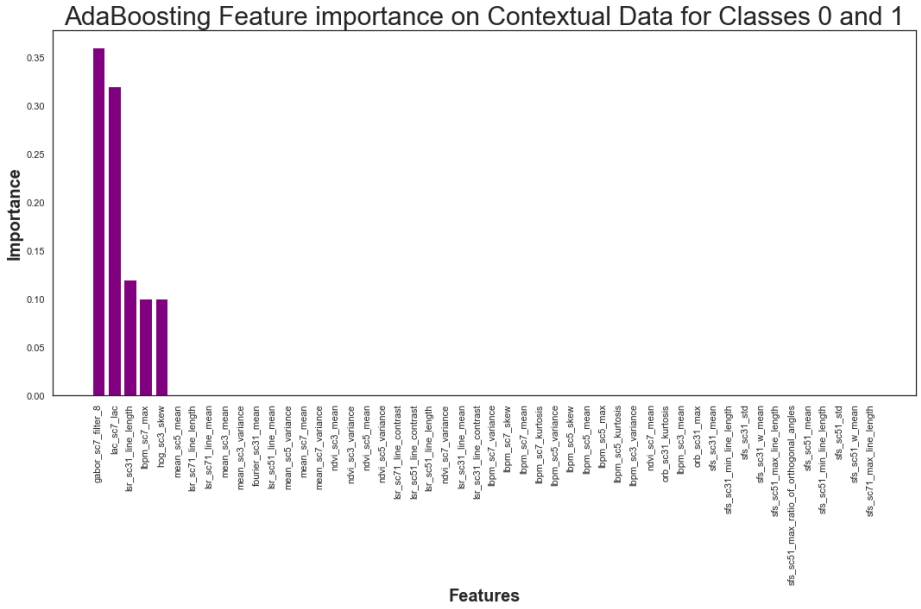
	precision	recall	f1-score	support
0	0.98	1.00	0.99	3042
1	0.00	0.00	0.00	52
accuracy			0.98	3094
macro avg	0.49	0.50	0.50	3094
weighted avg	0.97	0.98	0.97	3094

(Figure 28: 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.

2.6.16 Contextual Feature Importance: AdaBoost Feature Importance

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



(Figure 29: 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.

2.6.17 *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

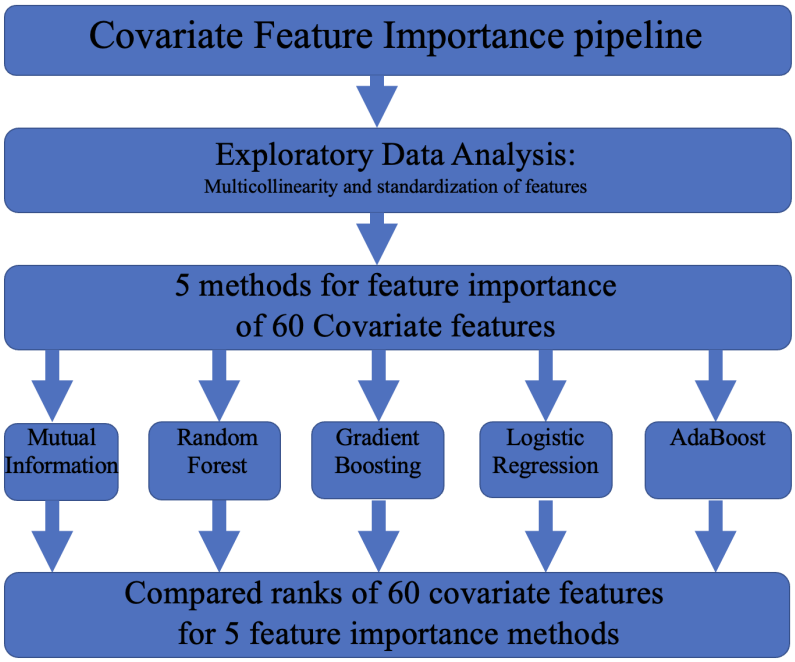
(Figure 30: Overview of Validation Results for Minority Class for Contextual Features)

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'. However, 'gabor_sc7_filter_8 did appear as the top feature for Gradient and Adaboost feature importance. 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.

2.7 Covariate Feature

2.7.1 Covariate Feature Analysis Overview

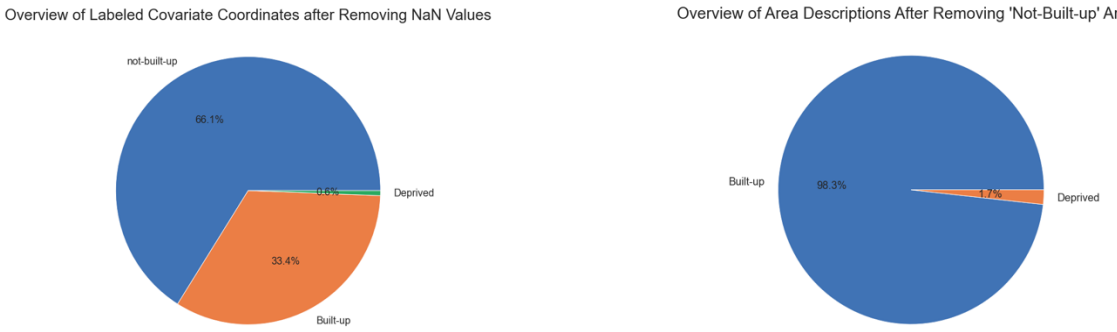
The covariate features were analyzed through exploratory data analysis and then feature importance. Once the covariate features were processed, exploratory data analysis was conducted on the entire data frame to identify any instances of multicollinearity. Also, the data was split on train/validation/test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



(Figure 31: pipeline for Covariate Features)

2.7.2 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.

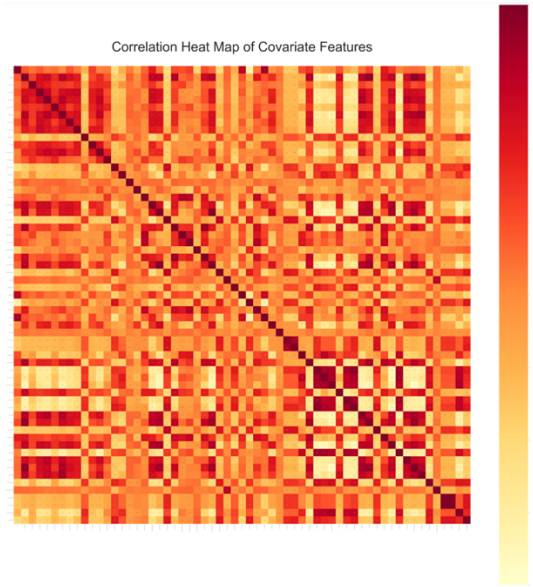


(Figure 32: Overview of Target Variable for Covariate Features)

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.

2.7.3 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.

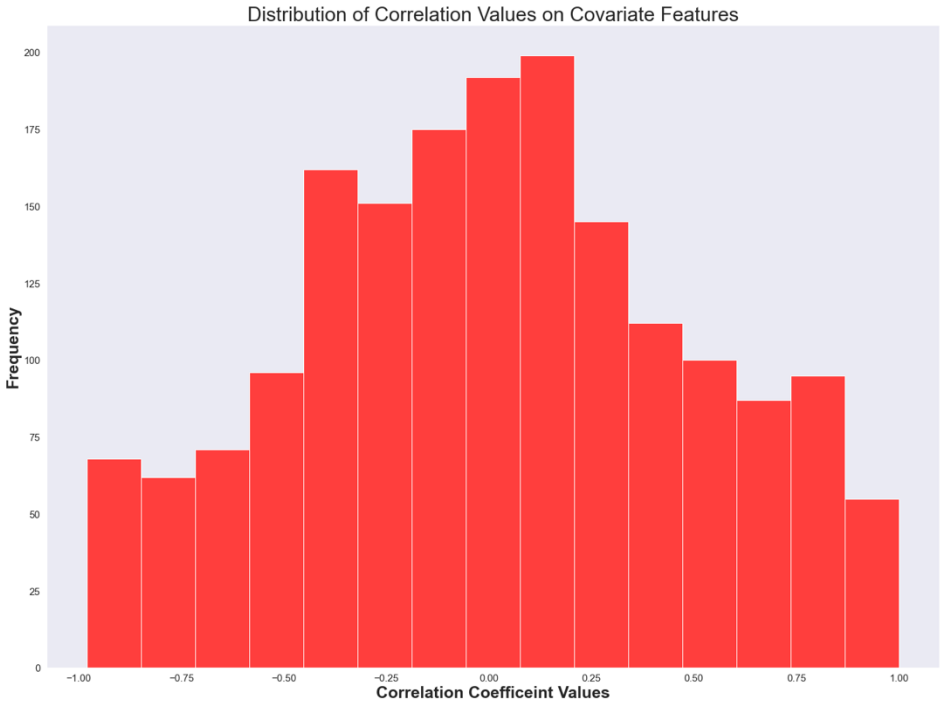


(Figure 33: Correlation Heat Map for Covariate Features)

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.

2.7.4 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.

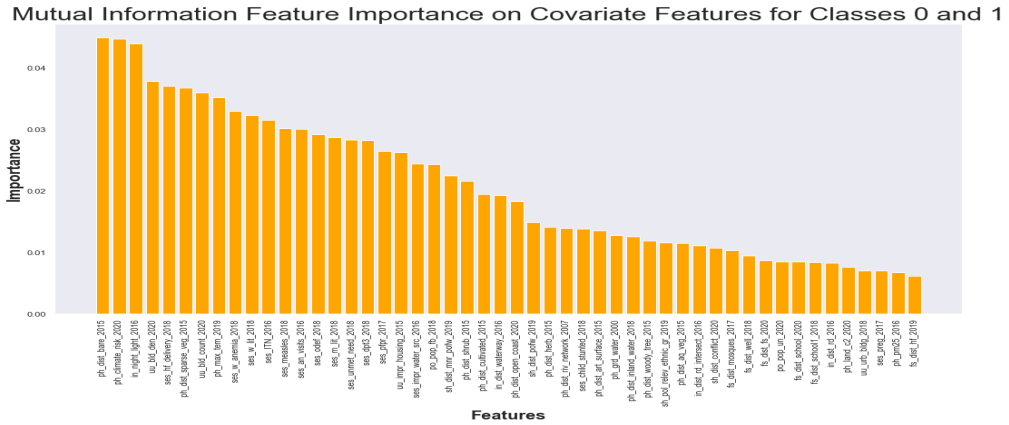


(Figure 34: Histogram of Correlation Coefficient of Covariate Features)

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.

2.7.5 Covariate Feature Importance: Mutual Information

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



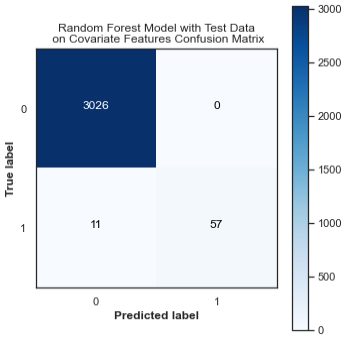
(Figure 35: Mutual Importance Feature Selection on Covariate Features)

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.

2.7.6 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



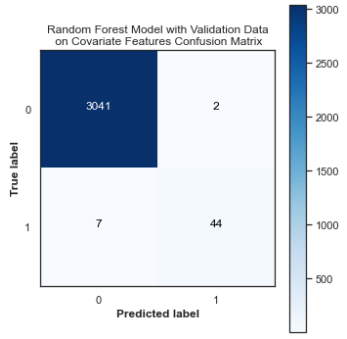
Random Forest Model with Test Data on Covariate Features Classification Report	precision	recall	f1-score	support
0	1.00	1.00	1.00	3026
1	1.00	0.84	0.91	68
accuracy			1.00	3094
macro avg	1.00	0.92	0.96	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 36: 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.

2.7.7 Covariate Feature Importance: Random Forest Validation Results

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



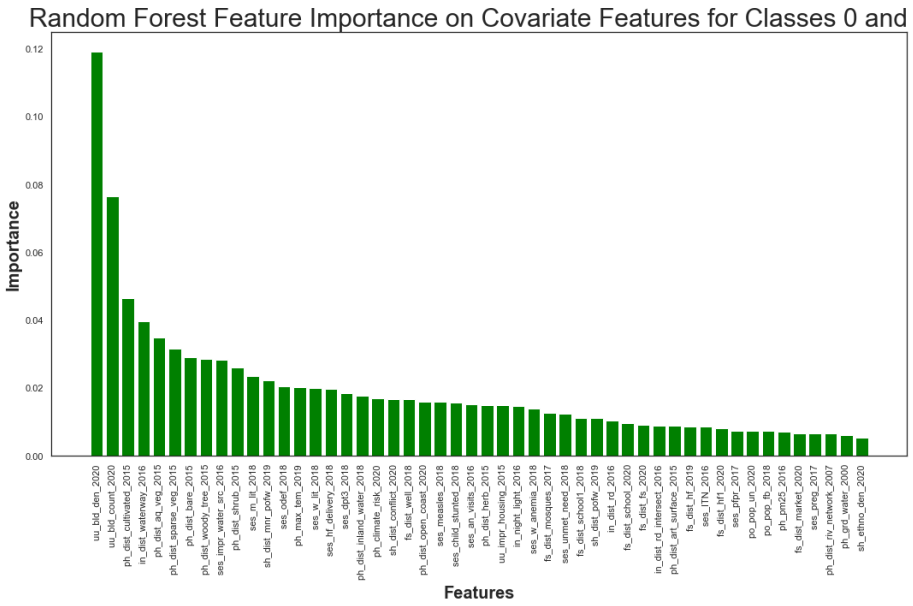
Random Forest Model with Validation Data on Covariate Features Classification Report	precision	recall	f1-score	support
0	1.00	1.00	1.00	3043
1	0.96	0.86	0.91	51
accuracy			1.00	3094
macro avg	0.98	0.93	0.95	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 37: 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.

2.7.8 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.



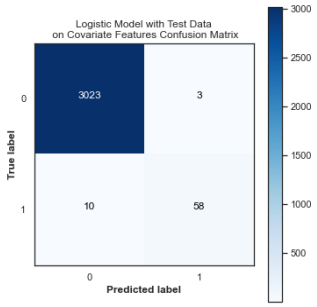
(Figure 38: Random Forest Feature Importance on Covariate Features)

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

2.7.9 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

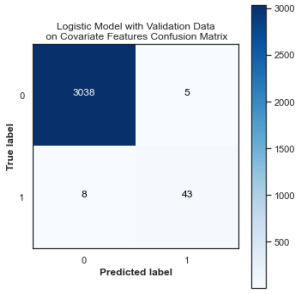
	precision	recall	f1-score	support
0	1.00	1.00	1.00	3026
1	0.95	0.85	0.90	68
accuracy			1.00	3094
macro avg	0.97	0.93	0.95	3094
weighted avg	1.00	1.00	1.00	3094

(Figure 39: 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-02.

2.7.10 Covariate Feature Importance: Logistic Regression Validation Results

A validation set was tested on the logistic model



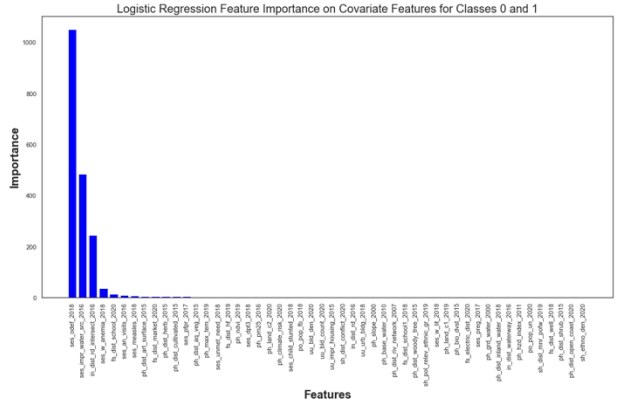

```
Logistic Model with Test Data
 on Covariate Features Classification Report
              precision    recall  f1-score   support

           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 40: 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.

2.7.11 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 41: Logistic Regression Feature Importance on Covariate Features)

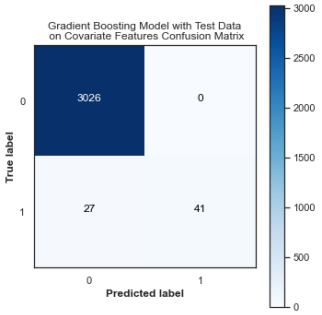
The value that was found to be the most important according to feature importance through the logistic regression method was 'ses_odef_2018'.

2.7.12 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

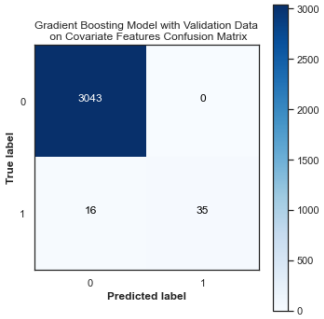
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3026
1	1.00	0.60	0.75	68
accuracy			0.99	3094
macro avg	1.00	0.80	0.87	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 42: 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.

2.7.13 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

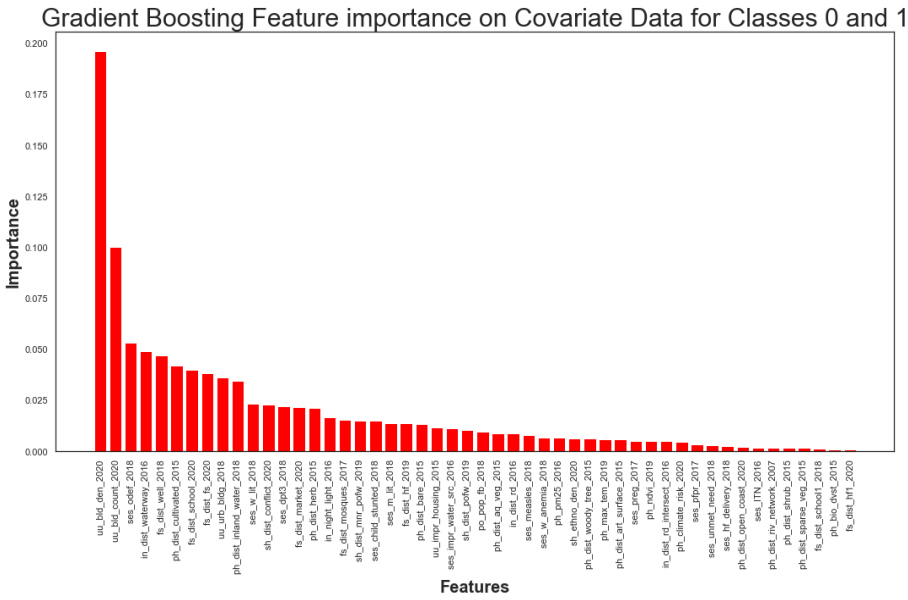
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3043
1	1.00	0.69	0.81	51
accuracy			0.99	3094
macro avg	1.00	0.84	0.91	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 43: 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.

2.7.14 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.



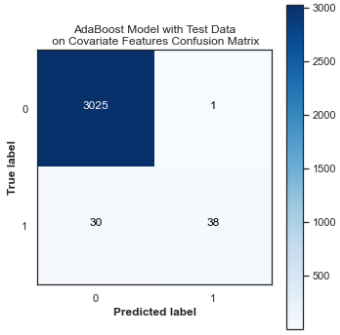
(Figure 44: Gradient Boosting Feature Importance on Covariate Features)

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

2.7.15 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

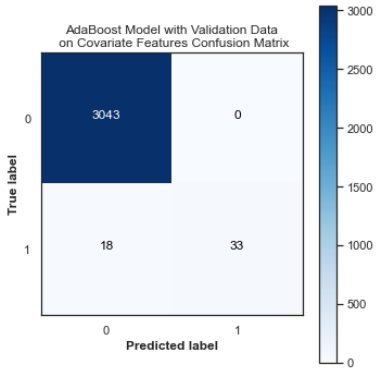
	precision	recall	f1-score	support
0	0.99	1.00	0.99	3026
1	0.97	0.56	0.71	68
accuracy			0.99	3094
macro avg	0.98	0.78	0.85	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 45: 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.

2.7.16 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

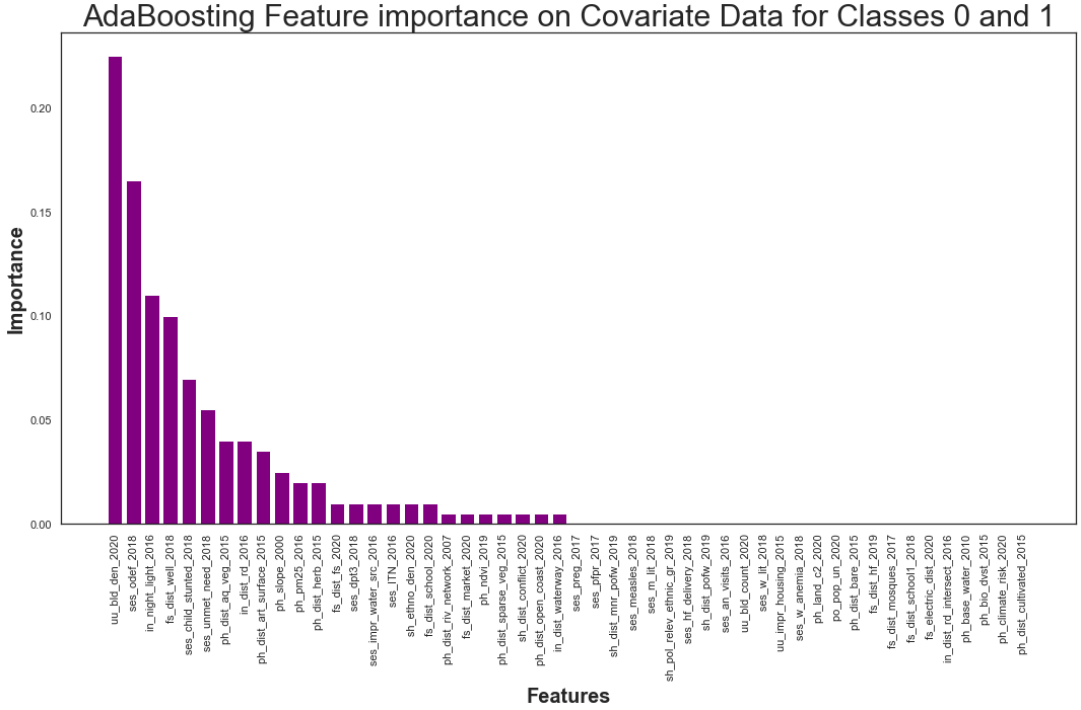
	precision	recall	f1-score	support
0	0.99	1.00	1.00	3043
1	1.00	0.65	0.79	51
accuracy			0.99	3094
macro avg	1.00	0.82	0.89	3094
weighted avg	0.99	0.99	0.99	3094

(Figure 46: 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.

2.7.17 Covariate Feature Importance: AdaBoost Feature Importance

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



(Figure 47: AdaBoost Feature Importance on Covariate Features)

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

2.7.18 *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 48: 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 49: 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.

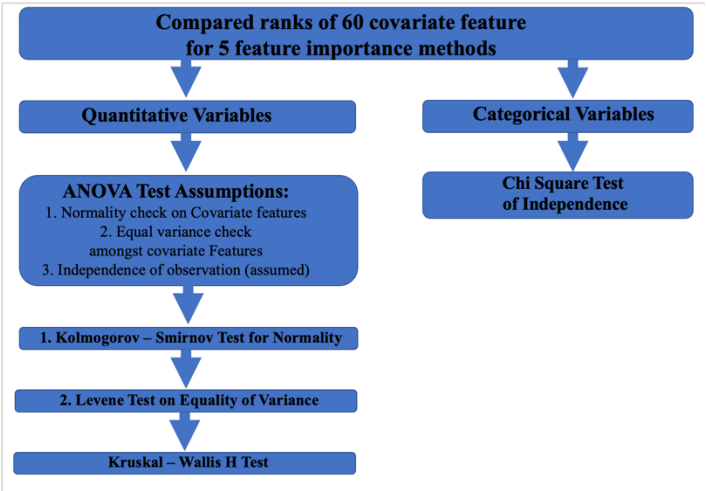
Covariate Features	Logistic Rank	Random Forest Rank	Gradient Boosting Rank	AdaBoost Rank	Mutual Information Rank	Overall Rank
Ph_grd_water_2000	41	49	59	55	37	55
Ph_base_water_2010	31	58	53	47	54	56
Fs_electric_dist_2020	39	54	54	45	60	57
Fs_dist_hfl_2020	57	41	50	58	50	58
Ph_hzd_index_2011	44	55	55	56	56	59
ph_land_c1_2019	37	60	58	57	57	60

(Figure 50: 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.

2.8. *Statistical Methods*

2.8.1 *Covariate Feature Importance: Statistical Testing*



(Figure 51: 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.

2.8.2 *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 0.02. This led the researchers to conclude that the covariate features for either class did not follow a normal distribution. It was observed that the 'Deprived' and 'Built-up' areas did come from the same distribution. The parameter *kstets* from the Scipy library was utilized to conduct the normality test[19].

2.8.3 *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 [20].

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 52: Leven Testing Results on Covariate Features)

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

2.8.4 *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 [21].

2.8.5 *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[22]

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 53: 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.

3 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**).

4 Bibliography

[1] sci-kit learn, StandardScalar,
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[2] sci-kit learn, SelectKbest,
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
[3] sci-kit learn, mutual_info_classif,
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html
[4] Grassberger, Stögbauer, and Kraskov, ‘Estimating mutal information,
https://journals.aps.org/pre/abstract/10.1103/PhysRevE.69.066138
[5] Aznar, ‘What is Mutual Information’, https://quantdare.com/what-is-mutual-information/
[6]sci-kit learn, *Random Forest Classifie*r,
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[7] sci-kit learn, ‘Feature Importance with a Forest of Trees’,
https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

[8] sci-kit learn, *Logistic Regression Classifie*r, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[9] sci-kit learn, *Gradient Boosting Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[10] sci-kit learn, *AdaBoost Classifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier.feature_importances_
[11] 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
[12]Engineering Statistics Handbook, 'Kolmogorov-Smirnov Goodness of Fit Test', https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
[13] Engieering Statistics Handbook, 'Levene Test for Equlaity of Variance' https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm
[14] Laerd Statistics, 'One-way ANOVA (cont'd)', https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide-3.php
[15] ScienceDirect, 'Kruskal Wallis Test', https://www.sciencedirect.com/topics/medicine-and-dentistry/kruskal-wallis-test
[16] Laerd Statistics, 'Kruskal-Wallis H Test using SPSS', https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php
[17] Penn State Eberly Colle of Science, 'The Chi-Square Test of Independence', https://online.stat.psu.edu/stat500/lesson/8/8.1
[18]sci-kit learn, *StratifiedKfolds*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
[19] Scipy, ktest, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html
[20] Scipy, levene, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.levene.html
[21]scipy, Kruskal, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html
[22] scipy, chi_contingency, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html

5 Appendix

Provided links to code below:

Link to researcher's Github (contains .py and .jpynb versions of analysis): https://github.com/mojahid/Mapping-Deprived-Areas-Using-Deep-Neural-Networks/tree/main/3.Contextual_and_Covariate_Feautres_Modeling/Code

Link to Jupyter notebook for contextual feature importance: https://github.com/mojahid/Mapping-Deprived-Areas-Using-Deep-Neural-Networks/blob/main/3.Contextual_and_Covariate_Feautres_Modeling/Code/Contextual_Feature_Importance_0_1.ipynb

Link to Jupyter notebook for covariate feature importance:
https://github.com/mojahid/Mapping-Deprived-Areas-Using-Deep-Neural-Networks/blob/main/3.Contextual_and_Covariate_Feautres_Modeling/Code/Covariate_Feature_Importance_0_1.ipynb