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

Ryan Engstrom
Data Science
George Washington University
Washington, D.C., United States
rengstro@gwu.edu

Amir Jafair
Data Science
George Washington University
Washington, D.C
ajafari@gwu.edu

Jake Lieberfarb
Data Science
George Washington University
Washington, D.C.
jakelieberfarb@gmail.com

Bradley Reardon
Data Science
George Washington University
Washington, D.C., United States
breardon7@gmail.com

Abstract—According to IDEAMAPS [1], at present, 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 and classical machine learning techniques 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.

Introduction

According to IDEAMAPS [1], at present, 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'.

Data Source

A. 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].

B. 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.

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, are contextual and/or covariate features useful in identifying 'Deprived' and 'Built-up' areas?

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 features 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.

Solution and Methodology

A. Overview of Data and Methodology

The contextual and covariate 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 dataset to identify any instances of multicollinearity. Also, the data was split on train/validation/test; 60/20/20. The data was standardized using *StandardScalar*[4]. After processing and exploring the data, five feature importance methods were utilized:

Table 1. Feature Importance methodologies

Feature Importance Method	***Function***	***Methodology***
Mutual Information	*SelectKBest[5], mutual_info_classif [6]*	*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].*
Random Forest	*feature_importances_[9]*	*The mean and standard deviation of accumulation of the impurity decrease within each decision tree of the random forest[10].*
Logistic Regression	*LogisticRegression [11]*	*odds ratio for coefficients*
Gradient Boosting	*feature_importances_[12]*	*rank features based on the total reduction of the criterion within each feature (Gini importance)*
Adaboost	*feature_importances_[13]*	*rank features based on the total reduction of the criterion within each feature (Gini importance)*

The feature importance methods were applied to the Contextual and covariate features on the validation data. The F1 score for the 'Deprived' class was reported. Hyperparameter tuning along with StratifiedKfold [14]cross validation was implemented in optimizing the models for feature importance.

B. Processing Contextual variables

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.

There was heavy class imbalance between the three classes of 'Not-Built-up'(67.5%), 'Built-up'(32.0%), and 'Deprived' (0.6%). After 'Not-Built-up' was removed, 'Built-up' consisted of 96.3% of the data and 'Deprived' was 1.7%

C. Contextual feature importance results

TABLE II. VALIDATION RESULTS ON CONTEXTUAL FEATURES

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

None of the models performed well in identifying the deprived areas. The most significant variable was found to be 'lbpm_sc7_max'.

D. Processing Covariate variables

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.

Upon processing the data, it was revealed that there were 1987 nan values and the covariate feature of 'ph_gdmhz_2005' contained only nan values. After cleaning was conducted, the labeled coordinates were 'Not-Built-up' (66.1%), 'Built-up' (33.4%), and Deprived (0.6%). Once 'Not-Built-up' was removed, 'Built-up' consisted of 98.3% and 'Deprived' was 1.7% of the data.

E. Covariate feature Importance Results

TABLE III. VALIDATION RESULTS ON CONTEXTUAL FEATURES

Function	*Validation F1 for 'Deprived'*
Random Forest	*0.91*
Logistic Regression	*0.87*
Gradient Boosting	*0.81*
Adaboost	*0.79*

The covariate features did an exemplary job at identifying the 'Deprived' areas. Random forest performed the best.

F. Covariate feature ranking

TABLE IV. TOP RANKED COVARIATE 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*

All of the top three covariate features had appeared in at least two of the 10 of the five feature importance methods.

Once the researchers had identified the top performing features within the covariate dataset, A statistical assessment was conducted through Analysis of Variance (ANOVA). This was done to see if the features could statistically derive a difference between 'Deprived' and 'Built-up' areas. An alpha of 0.05 was used for this analysis. No outliers were removed form this analysis has there was already a large class imbalance between 'Deprived' and 'Built-up'.

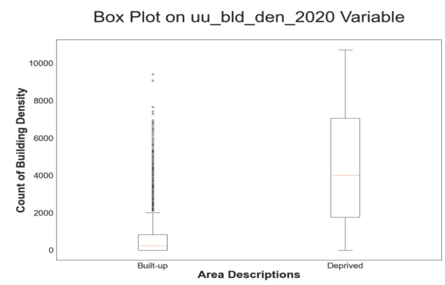
TABLE . STATISTICAL TEST IMPLEMENTED ON DATA

The assumptions of normality, equality of variance and independence of samples had to be tested before selecting the proper ANOVA test[15]. Independence of area descriptions was assumed.

Statistical test	*Scipy Function*	*Reasoning*
Kolmogorov-Smirnov test[16]	*ktest[17]*	*tests normality of data*
Levene Test[18]	*levene[19]*	*equality of variance*
Kruskal-Wallis H test[20]	*kruskal[21]*	*non parametric ANOVA test[22]*
chi square[23]	*chi2_contingency[24]*	*test of independence on categorical features*

Within the covariate features, there were 54 continuous variables and 6 categorical variables. The Kolomogorov - Smirov test revealed that the features were not normally distributed. The Levene test revealed that the majority of features did not exhibit equal variance. The researchers decided to utilize the Kruskal - Wallis H test for its ability to work with non parametric data. Although heteroscedasticity was present in the data, box plots of the variables still indicated a difference in covariate features for 'Deprived' and 'Built-up'.

FIGURE 1 . BOX PLOT OF HIGHEST RANKED COVARIATE FEATURE



The boxplot (Figure 1) illustrated that the quantiles between 'Built-up' and 'Deprived' did not overlap. there were noticeable outliers within the 'Built-up' boxplot. Upon inspection of the results from the non parametric Kruskal - Wallis H test. It was found that the top 21 ranked features were all statistically significant in identifying 'Deprived' and 'Built-up' areas.

For the six categorical features, only one covariate feature 'fs_electric_dist_2020' was found to be statistically independent. The other five were dependent.

Results

C. Experimentation Protocol

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-score 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 [25]. The F1-score of a classification model is calculated as follows:

$$2(P*R) / (P+R)$$

P = precision
R = recall

The micro F1-score indicates how well a model was able to correctly classify the deprived instances since this class is 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 models, feature reduction sets, and included class combinations, we totaled 25 distinct models per covariate and contextual dataset, summing to 50 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, we used GridsearchCV [26] from the sklearn library and a hyper-parameter space for the MLP [27] and Logistic Regression [28] models. The standard base models for the Gradient Boosting [12] and Random Forest [9] models were used without any parameter tuning since we 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:

TABLE I. MLP GRIDSEARCHCV HYPERPARAMETERS

Hidden Layers	*(60, 100, 60)*	*(100,100,100)*	*(50,100,50)*	
Activation	*identity*	*relu*	*logistic*	*tanh*
Solver	*adam*			
Alpha	*0.0001*			
Learning Rate	*invascaling*			

TABLE II. LOGISTIC REGRESSION GRIDSEARCHCV HYPERPARAMETERS

Penalty	*l1*	*l2*	*elasticnet*	*none*	
Dual	*True*	*False*			
C	*0.001*	*0.01*	*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.

D. Contextual Feature Results

BEST PERFORMING CONTEXTUAL MODEL ON CLASSES 0 & 1

Dataset	*Model*	*Feature Set*	*Feature Count*	*Classes*	*F1-Deprived*	*F1-Macro*
contextual	*Ensemble*	*All*	*144*	*classes 0&1*	*0.35*	*0.67*

a. Ensemble model consisted of nine (9) MLP models

Fig. 1. (Best Performing Contextual Model on Classes 0 & 1)

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 [29] model trained on the entire feature set. This ensemble model consisted of nine (9) MLP [27] 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 GridsearchCV [26] 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.

TABLE III. CONTEXTUAL CONFUSION MATRIX - CLASSES 0 & 1

True	*Built-up*	*3033*	*9*
	Deprived	*39*	*13*
		Built-up	*Deprived*
	Predicted		

Fig. 2. (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.)

Fig. 3. Best Performing Contextual Model on All Classes

Dataset	*Model*	*Feature Set*	*Feature Count*	*Classes*	*F1-Deprived*	*F1-Macro*
contextual	*Ensemble*	*All*	*60*	*all classes*	*0.25*	*0.70*

b. Ensemble model consisted of nine (9) MLP models

(Best Performing Contextual Model on all 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 [29] model trained on the entire feature set. This ensemble model consisted of nine (9) MLP [27] 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 GridsearchCV [26] 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.

TABLE IV. Contextual Confusion Matrix - All Classes

True	*Built-up*	*2729*	*6*	*325*
	Deprived	*25*	*8*	*16*
	Non-built-up	*267*	*1*	*6138*
		Built-up	*Deprived*	*Non-built-up*
	Predicted			

Fig. 4. (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.

E. Covariate Feature Results

Best Performing Covariate Model on Classes 0 & 1

Dataset	*Model*	*Feature Set*	*Feature Count*	*Classes*	*F1-Deprived*	*F1-Macro*
covariate	*MLP*	*All*	*60*	*classes 0&1*	*0.96*	*0.98*

c. MLP: Multilayer Perceptron

Fig. 5. (Best Performing Covariate Model on classes 0 & 1)

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 GridsearchCV[26] and the parameter space mentioned in the *Experimentation Protocol* section, 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.

TABLE V. Covariate Confusion Matrix - Classes 0 & 1

True	*Built-up*	*3040*	*3*
	Deprived	*1*	*50*
		Built-up	*Deprived*
	Predicted		

Fig. 6. (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.)

Fig. 7. Best Performing Covariate Model on All Classes

Dataset	*Model*	*Feature Set*	*Feature Count*	*Classes*	*F1-Deprived*	*F1-Macro*
covariate	*RF*	*Minfo*	*50*	*all classes*	*0.94*	*0.98*

d. RF: Random Forest

Fig. 8. (Best Performing Covariate Model on all classes)

The model that performed best on the full contextual features dataset containing all instances for every class was a Random Forest model trained on the top 50 features in the mutual information feature importance set.

TABLE VI. Covariate Confusion Matrix - All Classes

True	*Built-up*	*3059*	*0*	*11*
	Deprived	*2*	*46*	*4*
	Non-built-up	*18*	*0*	*5975*
		Built-up	*Deprived*	*Non-built-up*
	Predicted			

Fig. 9. (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.

Discussion

Unlike previous work done on utilizing feature importance methods to identify important features 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

Conclusion

This project as a whole aimed to apply various deep learning and traditional machine learning classification techniques 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 sub categories: 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.

TABLE VII. BEST PERFORMING MODELS

Dataset	*Model*	*Feature Set*	*Feature Count*	*Classes*	*F1-Deprived*	*F1-Macro*
covariate	*MLP*	*All*	*60*	*classes 0&1*	*0.96*	*0.98*
covariate	*RF*	*Minfo*	*50*	*all classes*	*0.94*	*0.98*
contextual	*GB*	*All*	*144*	*classes 0&1*	*0.35*	*0.67*
contextual	*MLP*	*All*	*144*	*all classes*	*0.25*	*0.70*

e. MLP: Multilayer Perceptron

f. RF: Random Forest

g. GB: Gradient Boosting

Fig. 10. (Best performing models per dataset)

As you can see in the table, the covariate features are extremely great indicators of class type while the contextual features are 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.

References

[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 Classifier*, 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 Classifier*, 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] Laerd Statistics, 'One-way ANOVA (cont'd)', https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide-3.php

[16] Engineering Statistics Handbook, 'Kolmogorov-Smirnov Goodness of Fit Test', https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm

[17] Scipy, ktest, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html

[18] Engieering Statistics Handbook, 'Levene Test for Equlaity of Variance' https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm

[19] Scipy, levene, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.levene.html

[20] ScienceDirect, 'Kruskal Wallis Test', https://www.sciencedirect.com/topics/medicine-and-dentistry/kruskal-wallis-test

[21] scipy, Kruskal, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html

[22] Laerd Statistics, 'Kruskal-Wallis H Test using SPSS', https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php

[23] Penn State Eberly Colle of Science, 'The Chi-Square Test of Independence', https://online.stat.psu.edu/stat500/lesson/8/8.1

[24] scipy, chi_contingency, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html

[25] Edpresso Team, "What is the F1-score?", Educative, https://www.educative.io/edpresso/what-is-the-f1-score

[26] sci-kit learn, *GridsearchCV*, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html