

STAT 6500

Statistical Machine Learning

Land Use Cover

Kendall Byrd - Atitarn Dechasuravanit - Alexys Rodriguez - Abdulaziz Alsugair

Final Report

Spring 2022
Monday, May 2

Contents

1	Introduction	2
1.1	Original Project Description	2
1.2	Data Description	2
1.3	Project Objectives	3
2	Problem statement	3
3	Methods	4
3.1	Available resolutions and variables	4
3.2	Classification methods	4
3.2.1	Neural Networks	4
3.2.2	Random Forest	4
3.2.3	K-Nearest Neighbor	5
4	Results	5
4.1	Neural Networks	5
4.2	Random Forest	6
4.3	Nearest Neighbours	7
5	Discussion and Conclusions	8
	References	9

1 Introduction

For this study, we propose implementing machine learning techniques on image segmented super-objects, to determine the effect scale has on accurately classifying land cover maps using super-object's features, extracted from urban satellite imagery at multiple image segmentation scales. The data set that will be analyzed in this project is the Urban Land Cover Data Set, see B. Johnson (2018). This data can be found in the UCI Machine Learning Repository and it was originally sourced by (B. A. Johnson 2012; B. Johnson and Xie 2013).

1.1 Original Project Description

The authors of the original project, B. Johnson and Xie (2013), classified a high-resolution image of an urban area using super-object information, i.e. the objects in the images (groups of pixels), representing real objects in the field (buildings, trees, cars, etc.) that a specific pixel belongs to. To extract those super-objects an image segmentation method to generate vectors (polygons) representing the real objects was used. The mentioned algorithm mainly depends of the input parameter *image pixel scale*, which represents the scale of the image used, relative to the original pixel size. Single pixels or fine-scale image segments, for instance in Figure 1, scale 20% left image (a), can generate multiple segments (polygons) for each super-object, and on the other side, a coarser scale, right image (c) at 140%, can mix multiple super-objects into one image segment. Depending on the real size of the original super-object, there is an optimum extraction scale, center image (b) at 80%. B. Johnson and Xie (2013) extracted the super-object for the multiple image segmentation scales, i.e. 20%, 40%, 60%, 80%, 100%, 120%, and 140%, and for each one, assigned **spectral** (vegetation index NDVI, average objects reflectance for each spectral band, etc.), **texture** (contrast, correlation, energy or homogeneity from the Gray-level co-occurrence matrix - GLCM), **size** (area, length, others) and **shape** (roundness, shape index, etc.) features, with the purpose of measuring the contribution of coarseness scales (individually and together) in the land cover use classification of the image, using machine learning techniques. In B. A. Johnson (2012), the authors created the land cover map from the original image, taking advantage of the improvements in model accuracy due to the additional feature information at different image segmentation scales.

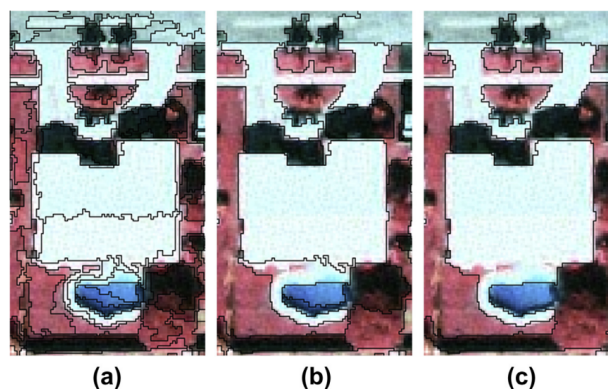


Figure 1: Image segmentation as a function of the pixel scale. Source: Johnson (2012)

1.2 Data Description

The multivariate Urban Land Cover data set has 168 rows and 148 columns. It has twenty-two attributes, that are repeated for seven different coarser scales. It contains training and testing data for classifying a high-resolution aerial image into nine classes (target classification variable) of urban land cover. The nine land cover classes are *concrete*, *trees*, *soil*, *grass*, *buildings*, *cars*, *asphalt*, *pools*, and *shadows*. There are a

low number of training samples for each class (14-30) and a high number of classification variables (148), so testing different feature selection methods will be interesting. The testing data set was generated from super-objects random sampling of the image. All attribute abbreviations and brief explanations can be seen in the Table 1.

Table 1: Variables Description

Variables	Variables
Class: Land cover class (nominal)	SD_R: Standard deviation of Red (texture variable)
BrdIndx: Border Index (shape variable)	SD_NIR: Standard deviation of Near Infrared (texture variable)
Area: Area in m2 (size variable)	LW: Length/Width (shape variable)
Round: Roundness (shape variable)	GLCM1: Gray-Level Co-occurrence Matrix (texture variable)
Bright: Brightness (spectral variable)	Rect: Rectangularity (shape variable)
Compact: Compactness (shape variable)	GLCM2: Another Gray-Level Co-occurrence Matrix attribute (texture variable)
ShpIndx: Shape Index (shape variable)	Dens: Density (shape variable)
Mean_G: Green (spectral variable)	Assym: Assymetry (shape variable)
Mean_R: Red (spectral variable)	NDVI: Normalized Difference Vegetation Index (spectral variable)
Mean_NIR: Near Infrared (spectral variable)	BordLngh: Border Length (shape variable)
SD_G: Standard deviation of Green (texture variable)	GLCM3: Another Gray-Level Co-occurrence Matrix attribute (texture variable)

Table 2 describes the features set for different coarser scales.

Table 2: Feature Sets by Image Segmentation Scales

Feature Set	Scale	Number of Variables	Variables Names	Variables Suffix
1	20	21	See Table 1	NA
2	40	21	See Table 1	40
3	60	21	See Table 1	60
4	80	21	See Table 1	80
5	100	21	See Table 1	100
6	120	21	See Table 1	120
7	140	21	See Table 1	140

1.3 Project Objectives

The practical end-goal is:

1. To reduce the cost of the image segmentation post-process by selecting a single coarseness level (resolution) that produces most accurate prediction, in order to create land cover maps from urban satellite images, and using multiple super-object's feature information at different scales,
2. To optimize object identification by selecting the best classification method in terms of (a) model accuracy, (b) model simplicity, (c) operational cost and (d) model interpretability.

2 Problem statement

Based on the study from (B. A. Johnson 2012; @ B. Johnson and Xie 2013), urban land-cover information is essential for numerous urban-planning applications, for instance, green space analysis, Lang et al. (2007), and urban land-use mapping, Herold, Liu, and Clarke (2003). Most land cover has traditionally been

obtained from satellite images using pixel-based image classification techniques. Nevertheless, in fine spatial resolution images with spectral variability within the same class can lead to low accuracy for classification using pixel-based image classification techniques. Therefore, B. A. Johnson (2012) presented the object-based classification methods involving segmentation of the image on different scales. The average size of the segment will vary depending on the specified scale parameter. These scales were ranging from 40 to 140 with 20 intervals in this study. For each image segment, features such as spectral (mean values and variance for each band), mean normalized differential vegetation index (NDVI), area, shape, texture, length and so on were calculated in one of different scales (21 features for each scale).

This project aims to use machine learning techniques and statistical tools that are different from the techniques used in the original study, to predict the target class of the object derived from segmentation at different scales of high-resolution urban-land cover image. Three machine learning techniques which are *neural networks*, *random forest* and *k-nearest neighbors* will be used for this project. The comparison of different classifiers for each scale and how well each classifier can perform at the different scales will be analyzed. In total, 24 machine learning models will be implemented in this study (seven scales with three machine learning techniques for each scale, plus the three methods applied to the whole data set)

3 Methods

Given mentioned objectives, the project team opted to perform concurrent analyses on the data where all proposed classification methods are used to classify objects across all coarseness levels; The combination of resolution and classification method that produces the most accurate predictions on the testing data will be selected as the one that meets the objectives.

3.1 Available resolutions and variables

The data was already collected and presented at 7 coarseness levels, each having 21 variable. The data will be processing will include: (a) eliminating highly correlated variables resulting in 12 independent variables (only for neural networks and nearest neighbors), (b) nominalizing all values to be used in the Neural Networks model, (c) compare original transformed (standard scaling) variables, (d) include machine learning models for independent feature sets (each feature set independently), (d) include machine learning models for all the variables (all feature sets integrated as a unit), and (e) analyze feature importance (random forest).

3.2 Classification methods

The team will be attempting 3 different classification methods:

3.2.1 Neural Networks

A supervised feed-forward neural networks model will be trained using the training data (168 points); and consists of 21 nodes in the input layer and 9 nodes in the output layer representing the available variables and the object classes. The team will attempt to optimize the models by varying the number of hidden layers and the number of nodes in each layer. The resultant models will be used to predict classes using the testing dataset to estimate each model's accuracy. For a overview, frameworks and challenges of Neural Networks see Prieto et al. (2016).

3.2.2 Random Forest

The random forest decision tree model is less computationally expensive to train and to implement, thus, serving the objective of reducing data collection and classification cost. The decision tree model is created by

recursively branching the data using the variable that adds most to the prediction of model; from that branch further branching is made using the best variable (the same variable may be used again). The recursion process stops when no further branching adds to the prediction value (e.g. when splitting the data results in 50/50 odds). The random forest creates multiple decision trees, each with different and random branching, regardless of the predictive power. The prediction can be performed using voting schema (majority) or averaging the results. More information applications of random forest classifier in remote sensors can be found in Belgiu and Drăguț (2016).

3.2.3 K-Nearest Neighbor

The simplest of models, where the class of an input is determined by its neighboring points. Either the euclidean or manhattan distance is used to determine the distance from other data points; and the class of the input point is predicted to be the same as the majority of its neighbors. The number of nearest neighbors can be tuned to produce more accurate predictions. See Taunk et al. (2019) for a clear review of this classifier for learning and classification purposes.

4 Results

4.1 Neural Networks

A total of 42 models were trained; 7 datasets (one for each coarseness level), 6 hidden layer configurations (1,2, or 3 layers; with 10 or 20 nodes per layer). The package ‘neuralnet’ was used to train the models (‘rprop+’ algorithm; Logistic activation function; minimizing SSE as error function; stopping criteria: reaching 100,000 epochs or SSE difference of 0.01 from previous run). All 42 trained models were used to predict the test data (507 observations) from their corresponding dataset (i.e. training a model on data from coarseness level 80, and predicting the classes on testing data 80). The test error rate, time to train, and number of epochs to reach a stopping criteria were recorded. The test error rate ranged from 28% - 46.7%; Compute time ranged from 0.37 - 1.64 seconds; and number of epochs ranged from 294 - 1917 steps. Originally, the team was planning to use compute time/epochs as decision criteria for selecting the best model, however, training time difference was only 1.27 seconds between best and worst models. Therefore, it was decided to consider test error rate solely in choosing the best model.

The best model (the one with least test error rate) is a model trained on dataset 80, with 2 hidden layers 20 nodes each. The plot for the best model is presented in figure 2, the actual vs. predicted confusion matrix is presented in figure 3, and the top 6 models are presented in figure 4.

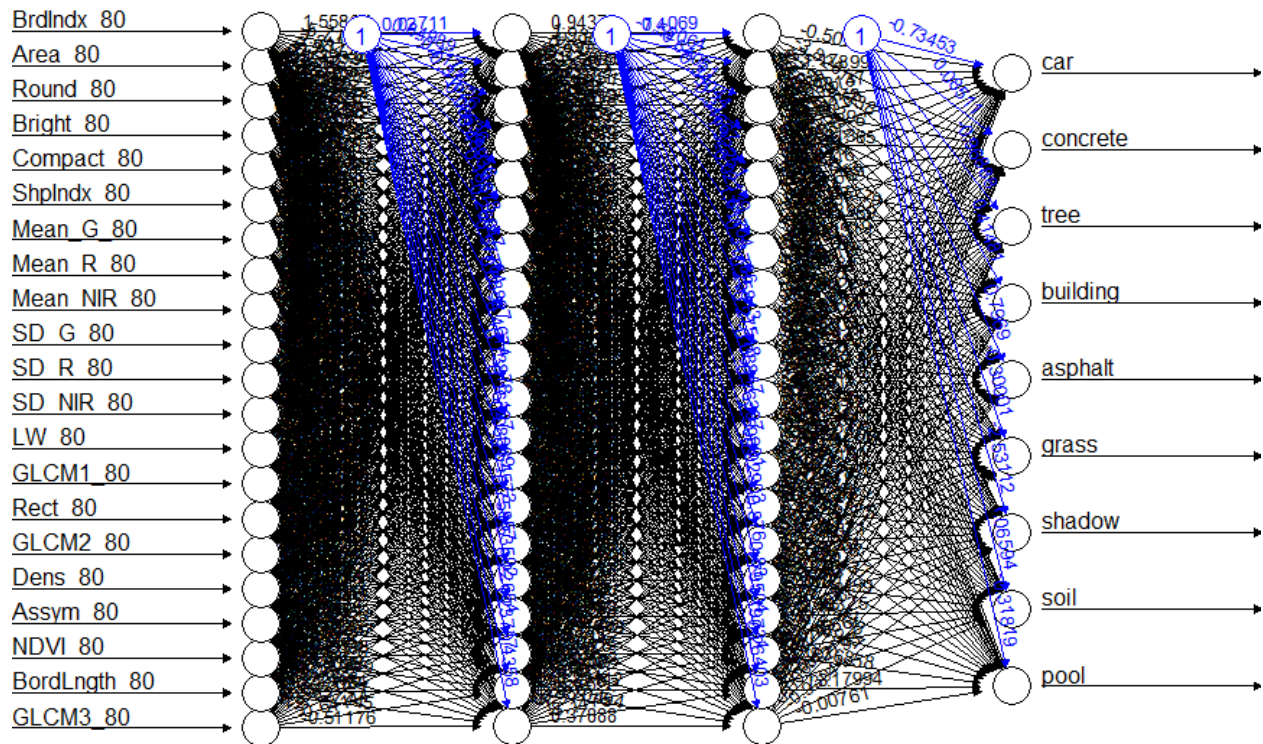


Figure 2: Best NN model (2 Hidden layers, 20 nodes each)

Table 3: Best NN model: Actual vs. Predicted

X	asphalt	building	car	concrete	grass	pool	shadow	soil	tree
asphalt	24	1	5	1	0	2	12	0	0
building	0	73	3	16	0	1	1	3	0
car	0	0	19	0	0	0	0	1	1
concrete	0	5	2	56	1	0	0	27	2
grass	0	0	3	1	59	0	0	4	16
pool	0	1	3	0	0	9	0	0	1
shadow	1	0	0	0	1	0	42	0	1
soil	0	1	1	0	10	0	0	8	0
tree	0	0	3	0	7	0	4	0	75

Table 4: Best NN models (arranged by test error rate)

Data.Set	configurations	Test.Error.Rate	Time	Steps
Data 80	c(20, 20)	28.0%	0.64	314
Data 80	c(20, 20, 20)	28.2%	0.91	367
Data 40	c(20, 20)	28.6%	0.72	418
Data 80	20	29.8%	0.65	632
Data 80	c(10, 10)	30.0%	0.61	642
Data 100	c(10, 10, 10)	30.8%	0.98	718

4.2 Random Forest

Transforming (standard scaling) the variables did not improve the model accuracy either for individual feature sets or for all feature set analyses. Super-objects at segmentation scale of 40% perform better in

order to create land cover maps using urban satellite images, for this specific case study with model accuracy of 79.3%. Once the coarseness scale increments, the accuracy of the model decrease in more that 10%, comparing with the broader scale of 140%. See Figure 3.

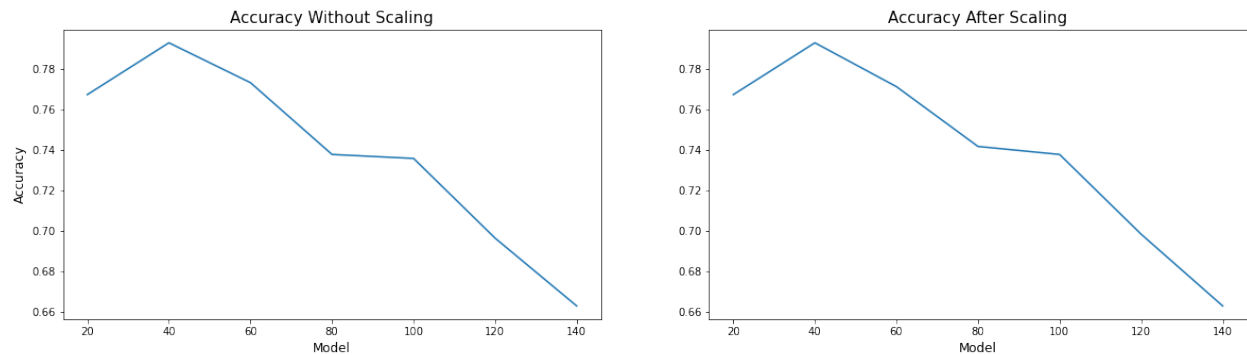


Figure 3: Random Forest individual feature sets (original - transformed). Model Accuracy.

For the best Random Forest model (scale 40%) an analysis of feature importance was performed. In the Figure 4, two types of features importance analysis is summarized, and in both, the most important variables are represented by taller bars (the black solid line represents the standard deviation). If a variable is not important for the model, the blue wide bar disappears. In the left graphic, the analysis consider the total amount of decrease in node impurity due to the splits over the predictors (Mean Decrease in Impurity - MDI), and the top five variables in importance are (a) NDVI, (b) super-object mean red, (c) mean near infrared, (d) mean green, and (e) bright. The right graphic represents a permutation variable importance analysis, with the 36% out-of-bag (OOB) samples not used in the bootstraps aggregating (bagging) sampling of random forest, and the most important variables are (a) NDVI, (b) area, (c) mean red, (d) mean near infrared, and (e) super-object near infrared standard deviation.

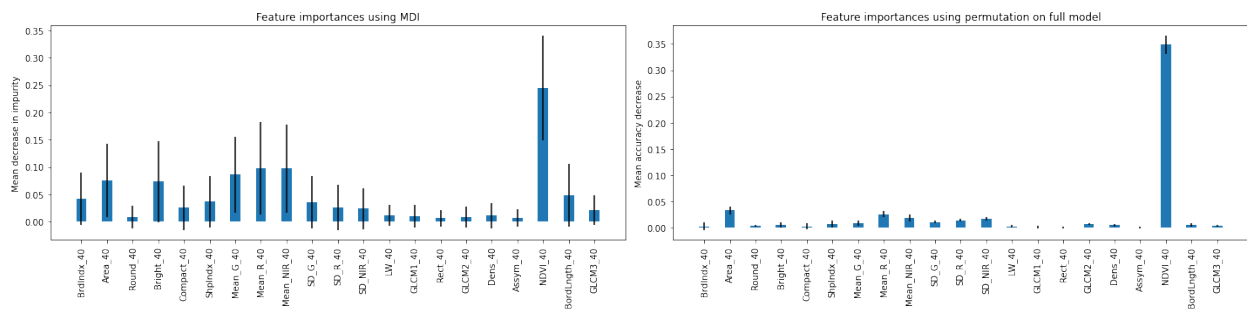


Figure 4: RF Feature Importances (MDI vs Permutation).

Including all feature sets as a unit for random forest analysis, improved the land cover classification from 79.3% to 82.2%.

4.3 Nearest Neighbours

K Nearest Neighbor was applied using all features without scaling, all features with scaling, individual feature sets without scaling, and individual features sets with training. For K nearest neighbor implementation hyperparameters were tuned using the package GridSearchCV. This package allowed for a random search of best parameters using 5-fold cross validation, iterated over 1000 times. For the best model, it was found that the optimal number of neighbors was 11, Manhattan distances should be used, and all neighbors should be weighted equally. Unlike the random forest model, transforming (standard scaling) of the variables did

improve the model's accuracy for both individual feature sets and all feature set analyses (see figure 5). Super objects at coarseness scale of 80 provided the best results for predicting land cover types from high resolution satellite imagery, with an accuracy score of 71.8% in this project. The next best coarseness scale was 60, with just a slightly lower prediction accuracy of 71.6%. For coarseness scales above 80, the accuracy of the model decreases with each incremental increase in coarseness scale. The best KNN model by coarser scale classified asphalt and buildings with the highest precision, 0.83 and 0.82 respectively, and classified cars and soil with the lowest precision, 0.31 and 0.46 respectively. In addition to the above results, when all features were applied the KNN model's prediction accuracy increased from 54% to 77.7% (see figure 6).

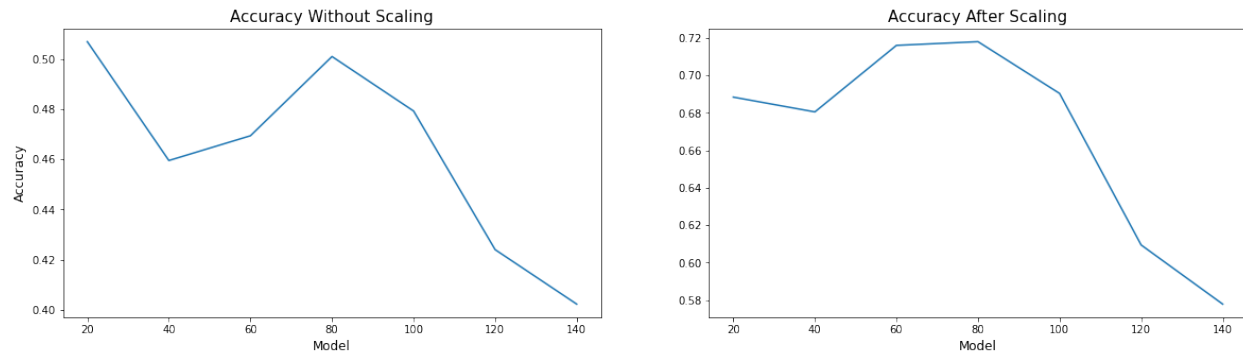


Figure 5: KNN individual feature sets (original - transformed). Model Accuracy.

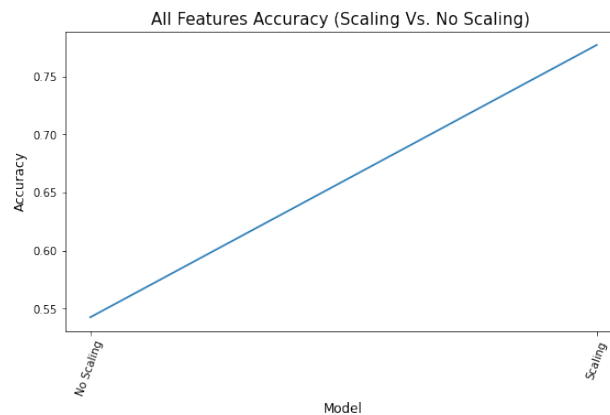


Figure 6: KNN all features (original - transformed). Model Accuracy.

5 Discussion and Conclusions

The Table 5, shows the best model for all methods, indicating if the data was transformed (standard scaling) and the image segmentation scale of the corresponding feature set.

Table 5: Best Model. All Machine Learning Methods

Method	Individual Feature Sets			All Feature Sets
	Data	Segmentation Scale	Accuracy	Accuracy
Neural Network	Not Transformed	80	72	63.3
Random Forest	Not Transformed	40	79.3	82.2
Nearest Neighbours	Transformed (Scaled)	80	71.8	77.7

As was mentioned in the project proposal, there are highly correlated variables across all feature sets, because each variable represents the same feature for the same super-object, but is extracted at a different image segmentation scale. Nonetheless, some of the main advantages of random forest are (a) decreasing auto-correlation in the trees by using only a random selection of variables, and (b) basing the prediction on the most important variables, therefore, this type of analysis with repeated correlated variables at multiples scales, when considered as a unity, is ideal for random forest classifier because it itself uses the most important inputs and selects the most representative elements to increase the accuracy of the final prediction without the need for an exhaustive exploratory analysis prior. Random forest was the best predictor in this study, with all variables, using both individual feature sets (79.3% of accuracy) or all together (82.3% of accuracy).

The Figure 7, shows the classification report for the best model (random forest at scale 40%). The best performance is achieved in classes *asphalt* and *building*, and the worse results are for classes *soil* and *tree*.

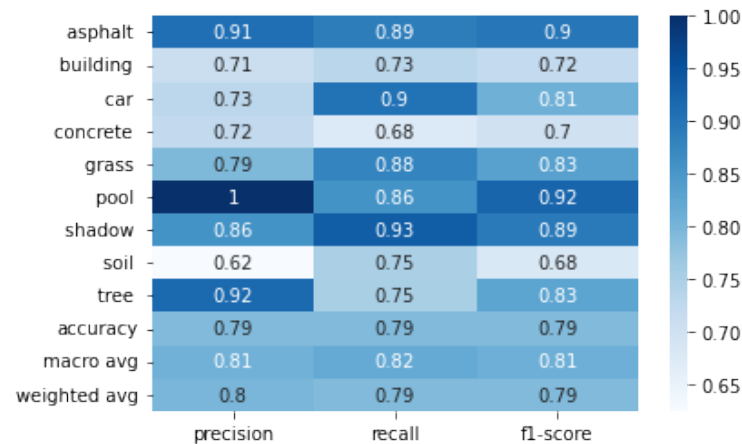


Figure 7: Classification Report - Best Model. RF Scale 40. Feature Set 2 (not transformed)

References

- Belgiu, Mariana, and Lucian Drăguț. 2016. "Random Forest in Remote Sensing: A Review of Applications and Future Directions." *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (April): 24–31. <https://doi.org/10.1016/j.isprsjprs.2016.01.011>.
- Herold, Martin, Xiao Liu, and Keith Clarke. 2003. "Spatial Metrics and Image Texture for Mapping Urban Land Use." *Photogrammetric Engineering and Remote Sensing* 69 (September): 991–1001. <https://doi.org/10.14358/PERS.69.9.991>.
- Johnson, Brian. 2018. "Urban Land Cover Data Set." Kamiyamaguchi, Hayama, Kanagawa, 240-0115 Japan: Institute for Global Environmental Strategies; Digital Repository. November 2018. [https://archive.ics.uci.edu/ml/datasets/Urban Land Cover#](https://archive.ics.uci.edu/ml/datasets/Urban+Land+Cover#).
- Johnson, Brian A. 2012. "High-Resolution Urban Land-Cover Classification Using a Competitive Multi-Scale Object-Based Approach." *Remote Sensing Letters* 4 (2): 131–40. <https://doi.org/10.1080/2150704x.2012.705440>.
- Johnson, Brian, and Zhixiao Xie. 2013. "Classifying a High Resolution Image of an Urban Area Using Super-Object Information." *ISPRS Journal of Photogrammetry and Remote Sensing* 83: 40–49. <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2013.05.008>.
- Lang, Stefan, Elisabeth Schoepfer, Daniel Hölbling, Thomas Blaschke, Matthias Möller, Thomas Jekel, and Elisabeth Schauppenlehner-Kloyber. 2007. "Quantifying and Qualifying Urban Green by Integrating Remote Sensing, GIS, and Social Science Method." In, 93–105. https://doi.org/10.1007/978-1-4020-6594-1_6.
- Prieto, Alberto, Beatriz Prieto, Eva Martinez Ortigosa, Eduardo Ros, Francisco Pelayo, Julio Ortega, and Ignacio Rojas. 2016. "Neural Networks: An Overview of Early Research, Current Frameworks and New

Challenges.” *Neurocomputing* 214 (November): 242–68. <https://doi.org/10.1016/j.neucom.2016.06.014>.
Taunk, Kashvi, Sanjukta De, Srishti Verma, and Aleena Swetapadma. 2019. “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification.” In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE. <https://doi.org/10.1109/iccs45141.2019.9065747>.