FINAL REPORT

AUGUST 05 2021

URBAN INVENTORIES
GABRIELLA BARRETT & SEAN ANDREW CHEN

NYU

Contents

Title

Analyzing Computational Methods for Urban Object Classification using LiDAR Data

GitHub

https://github.com/UrbanInventories/UrbanInventories

Abstract

As LiDAR technology becomes cheaper and more readily accessible, point cloud data of urban areas in tandem become not only more abundant but also of higher fidelity and resolution. This data can be of great use to city officials and planners for many purposes, including the inventorying and cataloging of objects within the urban and built environments – objects such as buildings, street furniture, traffic lights, and lampposts. To accomplish this inventorying, data scientists can utilize state of the art as well as conventional machine and deep learning techniques for urban object classification. Faced with a multitude of methods, the problem as presented to the data scientist is which technique to use. This project tests multiple machine and deep learning techniques on urban object LiDAR data comparing them in order to determine which is most adept and accurate at the task of object classification within the urban context. While there is heightened interest in deep learning, machine learning may very well perform as well if not better with greater interpretability and fewer resource requirements. Such a finding implies greater accessibility to urban object classification as machine learning in contrast to deep learning may not require as much expertise or computational resources.

Problem Statement

Cities would benefit greatly from being able to identify and catalog objects within the urban and built environments, whether it be for the purposes of guiding autonomous vehicles through haphazard urban streets or the preservation of the architectural heritage of the city. Ever evolving and advancing LiDAR technologies – such as terrestrial or aerial sensing – offer the potential for the realization of such endeavors. Researchers and data scientists have approached analyzing LiDAR data for the specific purpose of urban object classification with various methods, most notably machine and deep learning.

Urban object classification using artificial intelligence requires a deeper look into the varying methods and techniques. Machine learning can comprise multiple algorithms while three-dimensional LiDAR point cloud data offer a plethora of features from which to learn. Deep learning as well can be separated into numerous different network architectures. Before one can create an automated process of inventorying the built environment using LiDAR point cloud data and artificial intelligence, one must select the best method – where best can be measured by varying metrics such as computational efficiency or final accuracy.

With advancements in LiDAR technologies, it is imperative that researchers test which object classification methods work best. The main problem and objective for this research project is to determine which algorithm and computational method performs the task of urban object classification best. From our research we will be able to make better choices to classify urban objects with LiDAR data.

Literature Review

Classifying objects from LiDAR point cloud data is an involved process still under intensive investigation by many researchers. This is largely due to the fact that cloud data are unique in their challenges: they are irregular, unstructured, and unordered [1]. The density of points may change from place to place – hence they are irregular. Point clouds only have implicit and not explicit neighborhood relations – hence they are unstructured [2]. Permutations of the order of points are insignificant – hence they are unordered. These all differ from – for example – a two-dimensional image where pixels are regular (their density doesn't change from place to place), structured (neighborhoods are explicit), and ordered (a permutation of pixels would result in different images). These all lead to various challenges in typical computer vision problems such as object classification.

All machine learning algorithms need features from which to learn. In machine learning – unlike deep learning – these features must be explicitly specified. Thus, feature selection is an important part of machine learning for point cloud data object classification. With 3D features, shape descriptors related to eigenvalues are one option: linearity, planarity, scattering, omnivariance, anisotropy, eigenentropy, sum of eigenvalues and chance of curvature [3]. Once feature extraction and specification has been performed, it is time to choose a specific classifier. Support Vector Machines, Decision Trees, Random Forest Ensembles, and AdaBoost are just a few to name [4], [5].

Deep learning approaches can be broken down into whether the network can work on a structured grid-based dataset or on a raw point cloud dataset. The former is when point cloud datasets are voxelized while the latter relies on unaltered original data. Structured grid based deep learning architectures include VoxNet - a voxel based shallow 3D convolutional neural network (CNN). For raw point cloud data, the original architecture is PointNet [6].

Each of these architectures have their unique attributes which define unique performance metrics. For instance, voxel-based networks suffer from high memory consumption due to the need for voxelization but overall have very high performance [1]. What one determines as better depends on what one is ultimately defining as better. If one were working with autonomous vehicles, speed and computational efficiency may have more importance than full accuracy. Since our project relies on aerial LiDAR data that is immediately atemporal – we are not measuring data as a vehicle moves through time – we can discount computational intensities more than other researchers would.

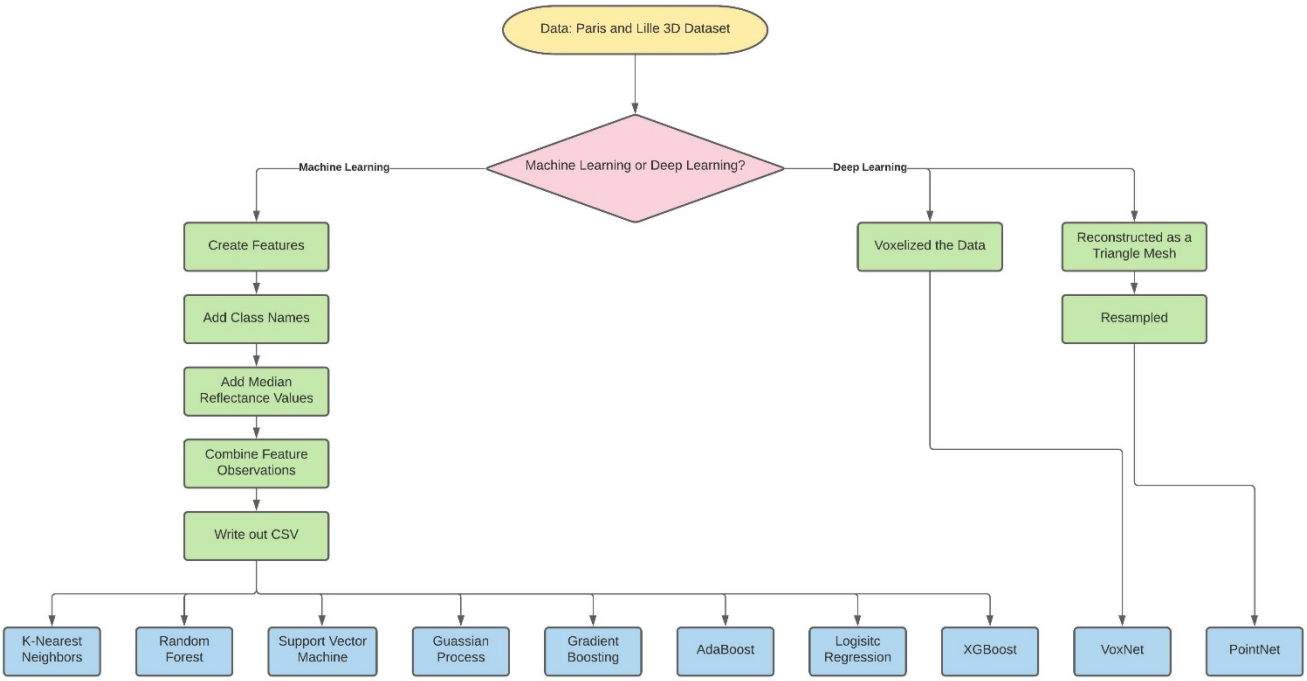


Figure 1 Process Flow Chart

Data

The Paris-Lille-3D Dataset was selected based on numerous factors. The dataset consists of 50 different classes of more than 140 million objects that were entirely labeled by hand. The large number of classes allows the classification of urban objects of various sizes and shapes.

Data processing was a critical step in the entire project. For the machine learning approach, features had to be calculated by calculating the covariance matrix of an object's point coordinates and then the matrix's eigenvalues. These eigenvalues were then themselves used to calculate:

1. Linearity
2. Planarity
3. Sphericity
4. Omnivariance
5. Anisotropy
6. Eigenentropy
7. Sum of Eigenvalues
8. Change of Curvature

In addition, reflectance values from the original data were kept.

For the deep learning approach, data was also highly processed. VoxNet required voxelization of the data and the creation of a three dimensional occupancy grid. PointNet on the other hand required the reconstruction of a triangle mesh through the ball pivoting method for each object so that a Poisson disk sampling method could provide an equal number of points for each object to be analyzed in the PointNet network.

Geometric Features	
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Sphericity	λ_3/λ_1
Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_1)^{1/3}$
Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
Eigenentropy	$-\sum \lambda_i \cdot \ln(\lambda_i)$
Sum of eigenvalues	$\lambda_1 + \lambda_2 + \lambda_3$
Change of curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$

Table 1 Geometric Feature Equations

Class	Number of Samples and Points							
	Lille1 # Samples	Lille1 # Points	Lille2 # Samples	Lille2 # Points	Paris # Samples	Paris # Points	Total # Samples	Total # Points
unclassified	1	54.88k	1	16.47k	1	60.79k	3	132.1k
other	16	70.15k	11	14.10k	11	30.82k	38	115.1k
road	1	34.80M	1	11.82M	1	19.68M	3	66.30M
sidewalk	18	6.566M	8	4.97M	5	4.6M	31	16.67M
island	7	458.8k	0	0	9	82.46k	16	541.2k
vegetation	32	562.2k	22	512.4k	6	157.1k	60	1.232M
building	34	18.2M	8	7.923M	5	9.421M	47	35.39M
post	9	13.51k	0	0	0	0	9	13.51k
bollard	122	34.80k	64	7.982k	84	28.33k	270	71.10k
floor lamp	72	232.9k	13	23.69k	21	113.2k	106	369.7k
traffic light	14	25.82k	11	27.41k	16	80.76k	41	133.10k
traffic sign	82	113.6k	29	69.89k	17	30.75k	128	214.3k
signboard	20	68.34k	12	43.14k	3	13.41k	35	124.9k
mailbox	0	0	0	0	1	4.739k	1	4.739k
trash can	11	14.72k	6	17.43k	22	30.35k	39	62.50k
meter	0	0	0	0	2	2.840k	2	2.480k
bicycle terminal	10	1.736k	0	0	13	6.451k	23	8.189k
bicycle rack	8	744	0	0	14	8.665k	22	9.439k
statue	2	4.158k	0	0	0	0	2	4.158k
barrier	45	83.94k831.0k	5	9.286k	7	56.10k	57	149.3k
roasting	6	14.18k	1	20.82k	4	1.677M	11	2.529M
wire	34	840.2k	8	9.325k	0	0	42	23.51k
low wall	58	849.2k	2	26.99k	9	951.4k	69	1.819k
shelter	0	0	0	0	3	83.45l	3	83.45k
bench	5	2.911k	11	364	2	1.391k	18	4.666k
distribution box	19	50.28k	8	14.89k	3	53.4k	30	118.2k
lighting console	78	7.25k	60	9.731l	9	4.393k	147	21.38k
windmill	1	10.17l	0	0	0	0	1	10.17k
pedestrian	17	24.81l	7	11.60k	61	150.2k	85	186.6k
parked bicycle	15	9.74k	0	0	33	81.67k	48	90.75k
mobile scooter	0	0	0	0	1	131	1	131
parked scooter	0	0	0	0	31	169.1k	31	169.1k
mobile motorbike	0	0	0	0	1	1.613k	1	1.613k
parked motorbike	2	2.428k	0	0	4	14.37k	6	16.79k
mobile car	21	175.0k	4	40.96k	5	66.35k	30	282.3k
stopped car	0	0	1	28.27k	1	9.375k	2	37.64k
parked car	182	2.266M	47	853.7k	65	1.610M	294	4.730M
mobile van	3	97.27k	1	41.6k	0	0	4	138.3k
parked van	5	84.75k	5	85.20k	9	357.6k	19	527.6k
stopped truck	0	0	0	0	1	235.7k	1	235.7k
parked truck	2	40.32k	0	0	1	53.44k	3	93.76k
stopped bus	0	0	0	0	1	78.41k	1	78.41k
parked bus	1	9.623k	0	0	0	0	1	9.623k
table	0	0	0	0	2	576	2	576
chair	0	0	0	0	8	4.482k	8	4.842k
trash can	138	148.8k	80	115.9k	0	0	218	264.7k
waste	5	2.307k	0	0	0	0	5	2.307k
natural	149	1.233M	47	296.9k	36	1.610M	232	3.240M
tree	72	2.310M	23	565.10k	101	4.755M	196	7.631M
potted plant	32	72.80k	5	26.9k	0	0	37	99.76k
Total	**1349**	**71.36M**	**501**	**26.84M**	**629**	**45.80M**	**2479**	**143.10M**

Table 2 Dataset Description

Methods

Our task for this research project is to classify objects in the urban environment using three dimensional point cloud data as gathered by aerial and terrestrial LiDAR. In order to do so, our primary method for classification involves machine and statistical learning methods.

We have used the following machine learning classifiers:

1. Logistic Regression Classifier
2. Naïve Bayes Classifier
3. K Nearest Neighbors Classifier
4. Gaussian Process Classifier
5. Support Vector Machine Classifier
6. Random Forest Ensemble Classifier
7. AdaBoost Classifier
8. Gradient Boosting Classifier
9. XGBoost Classifier
10. Voting Classifier

These methods have been implemented and executed using Python through the Scikit-Learn library. Using this package, we have developed a pipeline for data scaling and transformation, hyperparameter tuning, and model evaluation through metrics. XGBoost, however, was performed using the XGBoost library.

All method hyperparameters were tuned using a randomized search with repeated and stratified k fold cross validation with five splits repeated twice. A greater number of splits and repeats were desired but slowed processing down significantly. A randomized search was utilized due to this processing time. Bayesian Optimization was attempted with the Scikit-Optimization package but failed to converge. As classes are imbalanced in the data, metrics were focused on the F1 score, particularly using the “micro” algorithm where true positives, false negatives, and false positives are calculated globally rather than for each label.

Alongside the machine learning, we implemented two neural network architectures:

1. VoxNet
2. PointNet

Both implementations relied on the Keras interface for the TensorFlow library. In comparison to PointNet, VoxNet has a simpler architecture and is in fact a three dimensional convolutional neural network. Two three dimensional convolutional layers followed by a pooling layer and finally two dense layers make up the VoxNet architecture. PointNet, however, relies on a combination of multilayer perceptrons and T-Net transformers.

Results

The project findings show that new techniques are not always better. An analysis of different techniques finds that XGBoost, a machine learning method, had the highest accuracy score of 0.9393. The next two highest scores were also with machine learning methods: AdaBoost with .931 and Gradient Boosting with .925. The scores for the deep learning methods, VoxNet and PointNet were 0.7300 and 0.4523 respectively, significantly lower than the scores for the machine learning methods. One can take away from this analysis that deep learning does not always produce better results. It also points to the fact that boosting and ensemble methods perform the best, particularly ones which use decision trees. The Random Forest Ensemble classifier itself scored at .911.

The greatest improvement through hyperparameter tuning is with AdaBoost, going from a baseline score of 0.1311 to an optimized score of 0.9311. In contrast, Gradient Boosting had a very small improvement in the score from 0.9197 to 0.9246.

Method	Baseline Score	Number of Hyperparameters	Optimized Score
Naïve Bayes	0.4344	NA	NA
K Nearest Neighbors	0.7770	4	0.9311
Random Forest	0.9115	6	0.9148
Support Vector Machine	0.368	4	0.811
Gaussian Process	0.7443	1	0.8000
Gradient Boosting	0.9197	3	0.9246
AdaBoost	0.1311	2	0.9311
Logistic Regression	0.1328	2	0.6885
XGBoost	0.9393	NA	NA
VoxNet	0.7300	NA	NA
PointNet	0.4523	NA	NA

Table 1 Method Accuracy Scores

Conclusions

Classifying objects in the urban environment from LiDAR point cloud data is an area rich in possibilities. How many park benches are in neighborhoods populated with marginalized populations? How many Victorian style buildings are left in the city and where? Where exactly are manholes so that utilities may better understand their networks? These are just a few questions that an urban object inventory created with LiDAR data and object classification could potentially answer.

But the rise of high resolution LiDAR data must be met with the ability to properly analyze that data. Object classification with three dimensional point cloud data is - in many ways - non conventional. The data itself is unordered, unstructured, and irregular. This problem has given rise to unique deep learning network architectures such as PointNet. But the transformation of these points into geometric features such as eigenentropy, sphericity, planarity, etc. offer an alternative through more conventional machine learning.

This project's analysis of different methods - both machine and deep learning - has shown that ensemble and boosting methods outperform other machine learning methods as well as deep learning architectures.

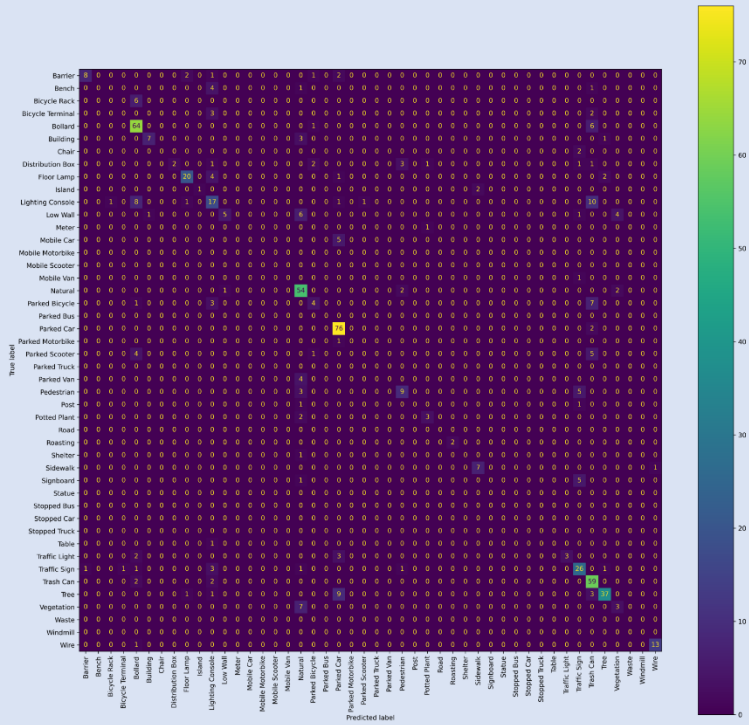
An important point of context is that PointNet and VoxNet in the past have been tested on datasets such as ShapeNet and ModelNet40. In these cases, objects tend to be formalized in surface volumes which are then sampled to create point clouds. Thus, one can say that these architectures were tested on objects outside of the urban context - in a clean, laboratory setting rather than the wonderfully messy and seemingly disorganized urban environment.

Future work will include greater testing of deep learning methods, most importantly PointNet++ - the successor to PointNet. There are still a multitude of deep learning methods to test. Data augmentation will also need to be utilized.

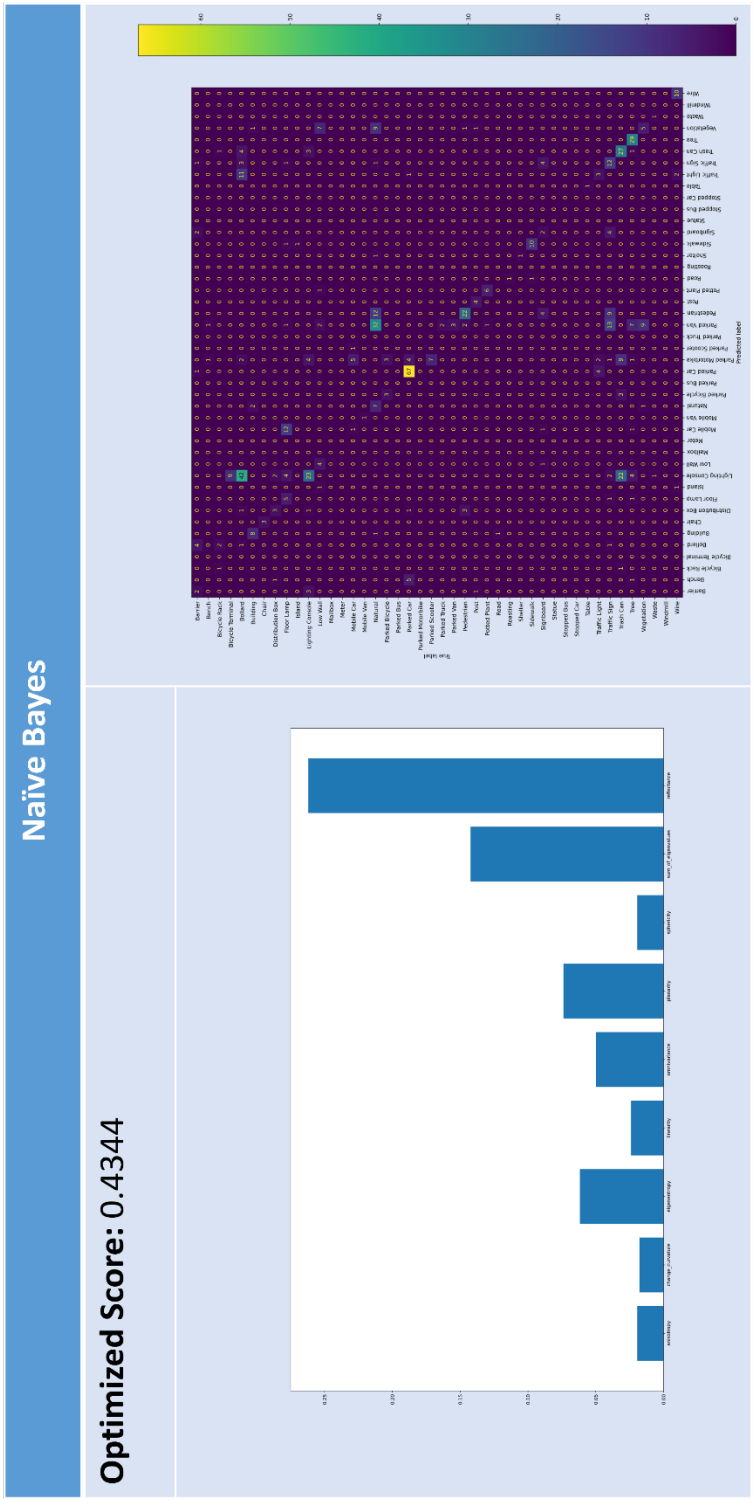
Creating highly accurate digital doubles of the urban environment and cataloging the objects found within the streetscape are exciting new possibilities brought about by not just advances in urban sensing technologies such as LiDAR but the use of machine and deep learning as well. This is an area that will necessitate continued study as data becomes finer and new neural network architectures are invented.

Appendix

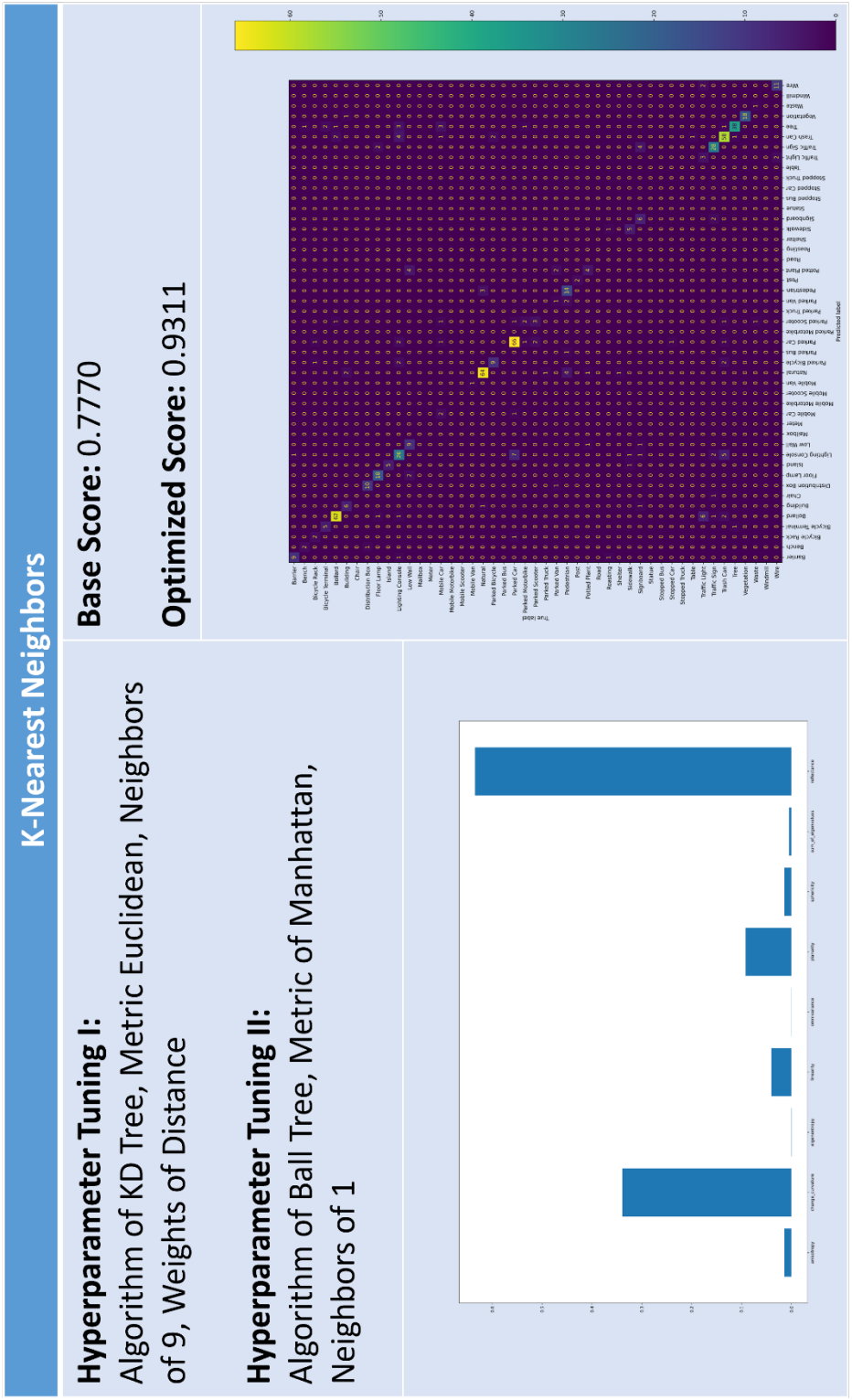
Logistic Regression

Logistic Regression
Hyperparameter Tuning I: C of 10, Solver of Newton CG **Base Score:** 0.1328 **Optimized Score:** 0.6885


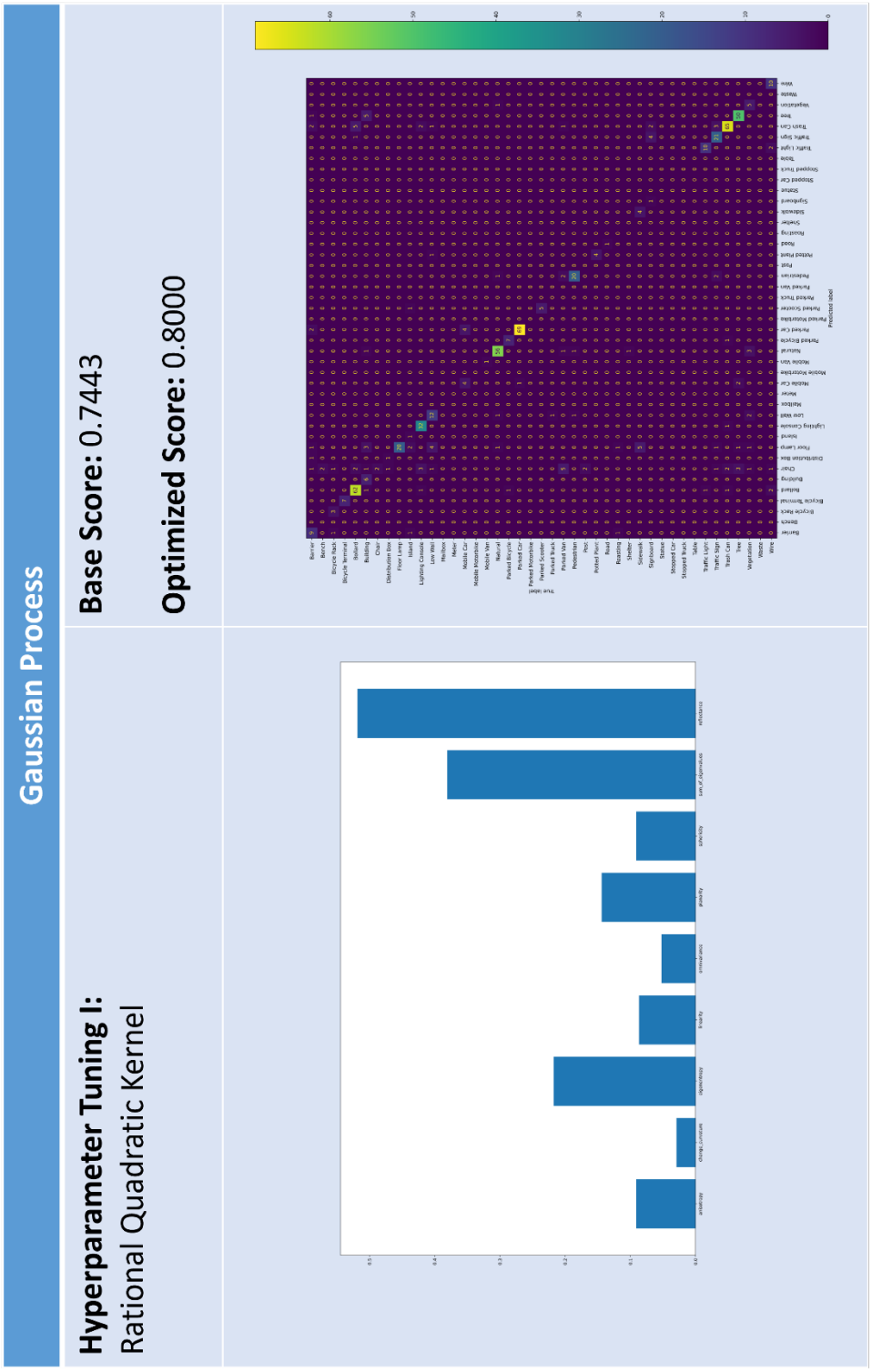
Naïve Bayes



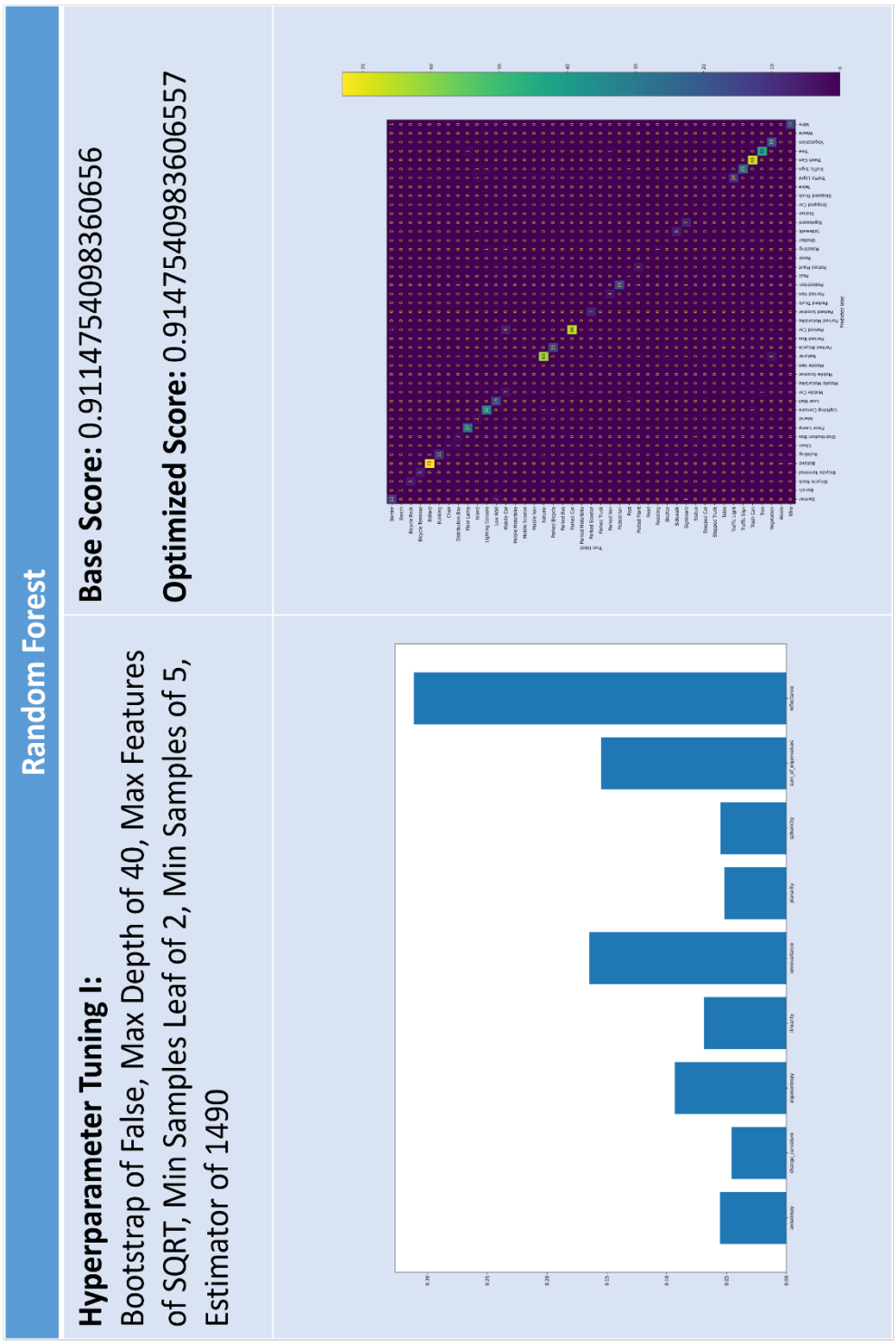
K-Nearest Neighbors



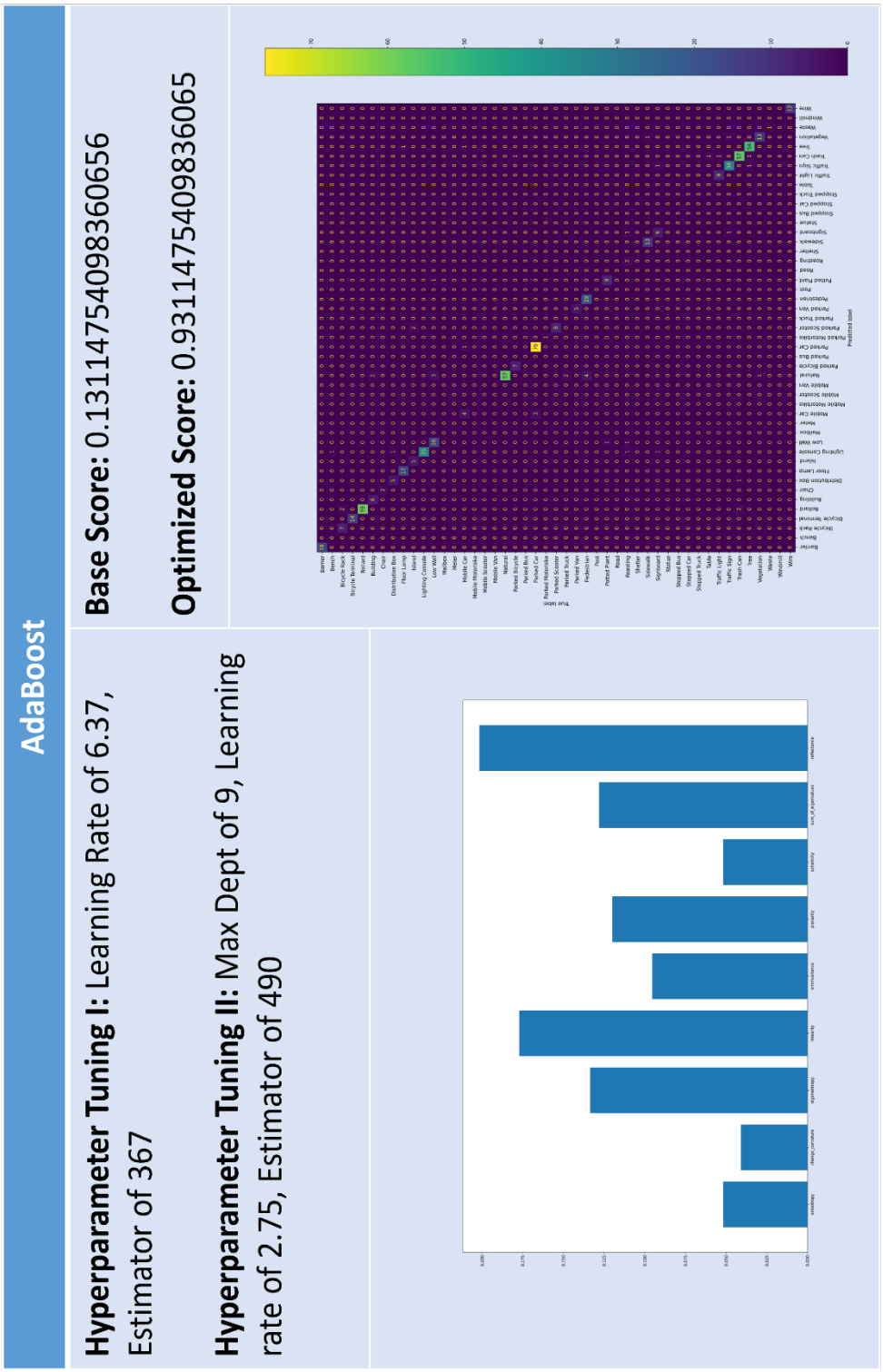
Gaussian Process



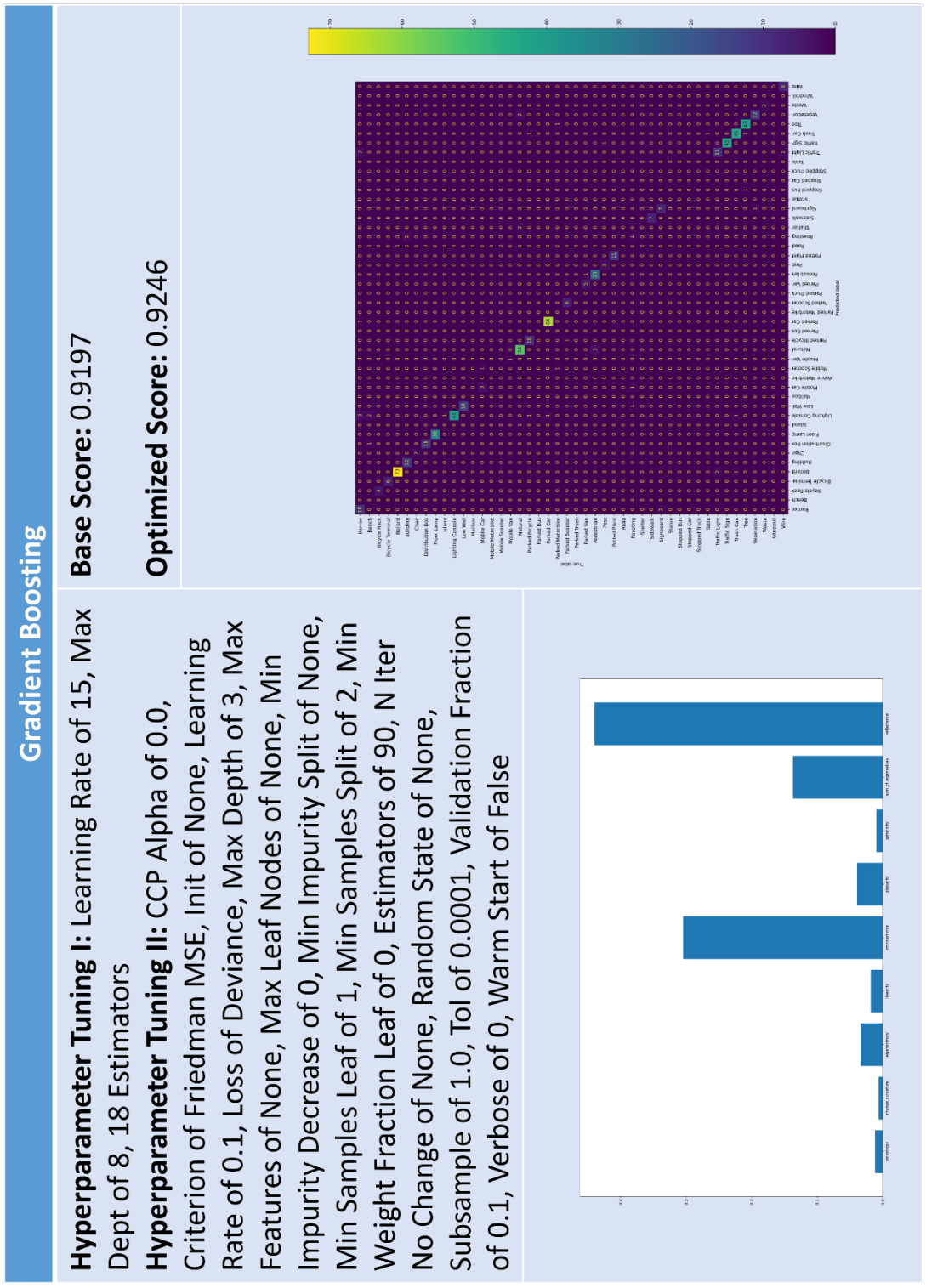
Random Forest



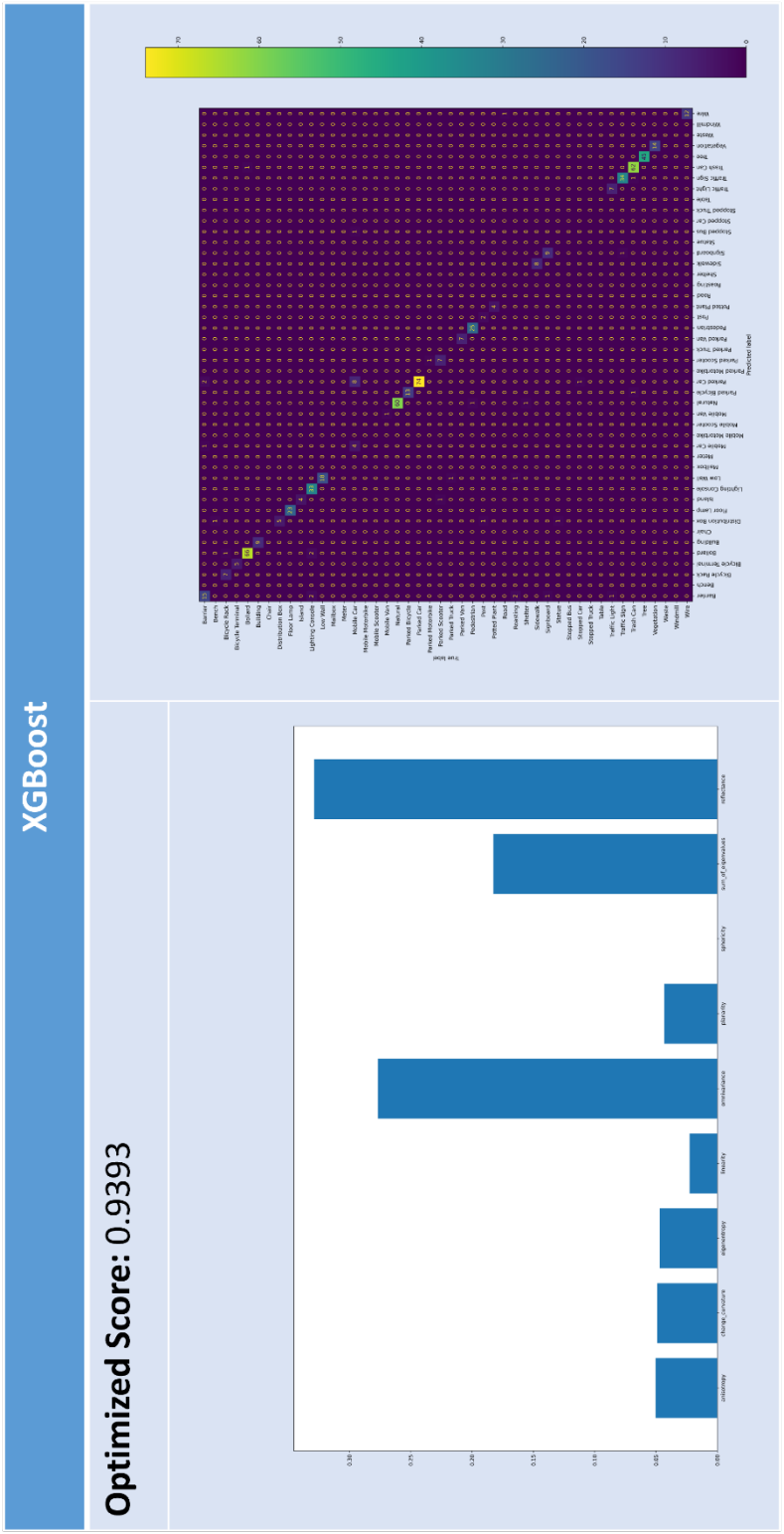
AdaBoost


AdaBoost
Hyperparameter Tuning I: Learning Rate of 6.37, Estimator of 367
Hyperparameter Tuning II: Max Dept of 9, Learning rate of 2.75, Estimator of 490
Base Score: 0.13114754098360656
Optimized Score: 0.9311475409836065

Gradient Boosting



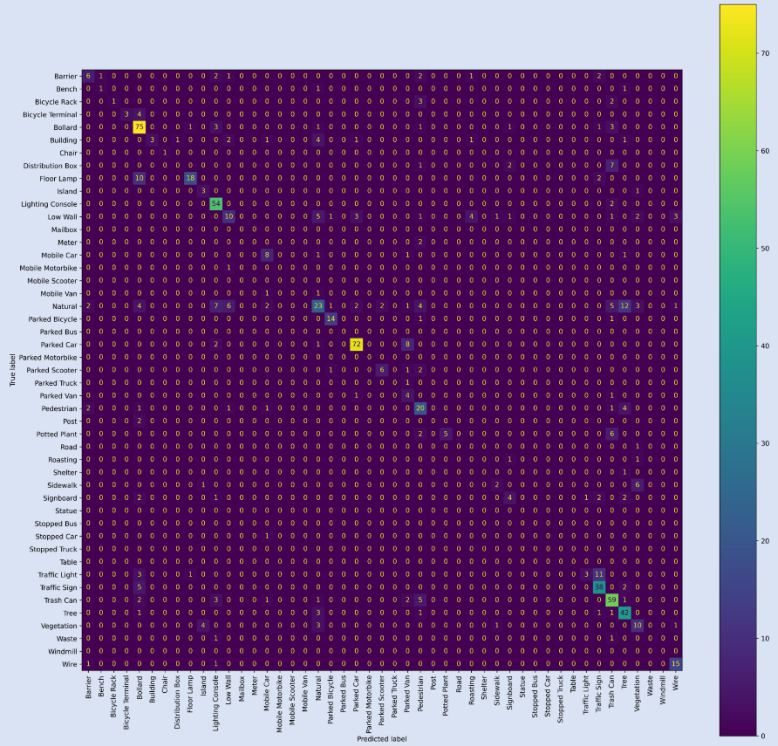
XGBoost



VoxNet

VoxNet

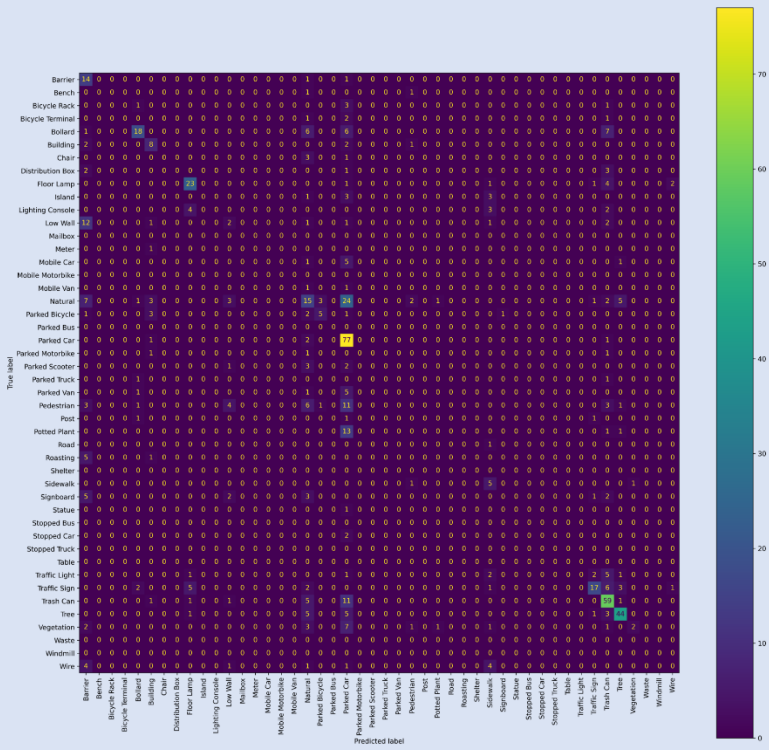
Optimized Score: 0.7300
Macro: 0.3711
Micro: 0.6501
Weighted: 0.6217



PointNet

Point Net

Optimized Score: 0.4523
Macro: 0.1615
Micro: 0.4523
Weighted: 0.3870



References

[1] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, “Review: Deep learning on 3D point clouds,” Remote Sensing, vol. 12, no. 11. MDPI AG, Jun. 01, 2020, doi: 10.3390/rs12111729.

[2] D. Griffiths and J. Boehm, “A Review on Deep Learning Techniques for 3D Sensed Data Classification,” Remote Sens., vol. 11, no. 12, p. 1499, Jun. 2019, doi: 10.3390/rs11121499.

[3] Q. Li and X. Cheng, “Comparison of different feature sets for TLS point cloud classification,” Sensors (Switzerland), vol. 18, no. 12, Dec. 2018, doi: 10.3390/s18124206.

[4] E. Özdemir, F. Remondino, and A. Golkar, “Aerial point cloud classification with deep learning and machine learning algorithms,” in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, Oct. 2019, vol. 42, no. 4/W18, pp. 843–849, doi: 10.5194/isprs-archives-XLII-4-W18-843-2019.

[5] F. Rottensteiner, “Advanced methods for automated object extraction from LiDAR in urban areas,” in International Geoscience and Remote Sensing Symposium (IGARSS), 2012, pp. 5402–5405, doi: 10.1109/IGARSS.2012.6352385.

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” arXiv Prepr. arXiv1612.00593, vol. 2017-Janua, pp. 77–85, Nov. 2016, doi: 10.1109/CVPR.2017.16.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer Verlag, 2016.

[8] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge, United Kingdom: Cambridge University Press, 2020.

[9] T. Hastie, J. Friedman, and R. Tisbshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2017.