THE GEO PROJECT
GEOSPATIAL EDUCATION & OUTREACH
www.geoproject.msstate.edu

GEO TUTORIAL

#QGIS

Dealing with Coastal Flooding, part 3A:
USING UNSUPERVISED MACHINE LEARNING FOR LAND USE LAND COVER CLASSIFICATION

Krzysztof Raczynski
Flavia Xavier
John Cartwright

Geosystems Research Institute
Mississippi State University

MAY 2025

This work was supported through funding by the National Oceanic and Atmospheric Administration Regional Geospatial Modeling Grant, Award # NA19NOS4730207.

GEOSYSTEMS RESEARCH INSTITUTE, MISSISSIPPI STATE UNIVERSITY, BOX 9627, MISSISSIPPI STATE, MS 39762-9652

The Geospatial Education and Outreach Project (GEO Project) is a collaborative effort among the Geosystems Research Institute (GRI), the Northern Gulf Institute (a NOAA Cooperative Institute), and the Mississippi State University Extension Service. The purpose of the project is to serve as the primary source for geospatial education and technical information for Mississippi.

The GEO Project provides training and technical assistance in the use, application, and implementation of geographic information systems (GIS), remote sensing, and global positioning systems for the geospatial community of Mississippi. The purpose of the GEO Tutorial series is to support educational project activities and enhance geospatial workshops offered by the GEO Project. Each tutorial provides practical solutions and instructions to solve a particular GIS challenge.

USING UNSUPERVISED MACHINE LEARNING FOR LAND USE LAND COVER CLASSIFICATION

Krzysztof Raczynski [1, 2, 4, 5, 6, 8] chrisr@gri.msstate.edu
Flavia Xavier [3, 7] flavia.xavier@msstate.edu
John Cartwright [7, 9, 10, 11] johnc@gri.msstate.edu

Geosystems Research Institute
Mississippi State University

CRediT: 1: Conceptualization; 2: Methodology; 3: Verification; 4: Resources; 5: Data Curation; 6: Writing - Original Draft; 7: Writing - Review; 8: Visualization; 9: Supervision; 10: Project administration; 11: Funding acquisition

REQUIRED RESOURCES

- QGIS 3+

FEATURED DATA SOURCES

- Click here to access the dataset used in this tutorial (2.973 MB).

OVERVIEW

Coastal areas across the United States face increasing challenges from changing water levels, which can lead to more frequent flooding and infrastructure strain. In communities like Bay St. Louis, Mississippi, rising water can make roads impassable, damage property, and disrupt daily life—posing serious concerns for homeowners and local economies.

As part of a planning team, your role is to assess how changing sea levels may impact the safety, infrastructure, and long-term growth of this Gulf Coast community. The focus is on protecting property, ensuring economic stability, and strengthening community resilience. This is the theme of the *Dealing with Coastal Flooding* tutorial series, which includes the following topics:

- Part 1: Creating Raster DEM from LiDAR Data
- Part 2: Spatial Predicates: Preparing Residential Data
- **Part 3A: Using Unsupervised Machine Learning for Land Use Land Cover Classification**
- Part 3B: Using Supervised Machine Learning for Land Use Land Cover Classification
- Part 4: Hydrologic Raster Preparation: Resampling and Burning Stream Network
- Part 5: Generating Flooding Extent with Raster Calculator
- Part 6: Calculating Spatial Statistics of Inundated Areas
- Part 7: Creating 3D Maps of Flooding Projections
- Part 8: 3D Map Animations
- Part 9: Creating and Animating Timeseries

In the previous part, we processed spatial information to prepare residential data for further analysis. In this part, we will use machine learning to classify Land Use Land Cover (LULC) within our studied region. To determine different LULC classes, we will need a satellite image with information from different bands, representing varying wavelengths of light captured by the remote sensors. There are two approaches we can take here to determine LULC classes automatically, that is, supervised and unsupervised machine learning. In the former case, we need to provide the algorithm with a training set, where we will tell it which areas are of which predetermined classes. The algorithm will use this information to extrapolate the dependencies over the area and classify each pixel according to similarities to the data we have prepared. In the second case, we do not predefine classes but let the algorithm decide which pixels are similar to each other. We will later decide what each detected class represents. In general, a supervised approach tends to provide more accurate results, while an unsupervised approach is useful if we don't know the final categories we will use or if we want to perform a quick categorization. In this tutorial, we will apply the unsupervised approach. Make sure to check the remaining tutorials in the series to learn more about the entire analysis process.

DATA

For this tutorial, we will use *Landsat* satellite imagery that you can download from USGS EarthExplorer (requires free account) or from the Featured Data Sources link above. If you chose the former approach, set the *spatial extent* to the area of **Bay St. Louis, MS,** and the *cloud cover* limit to ***0%*** to obtain only a clear image. For the datasets, use Landsat Level 1. You can choose any of the dates, but it is recommended to use some from the ***spring*** or ***summer*** time to obtain bigger differences in vegetation, as some of the winter images may lead to lower precisions of the result. You should download either the entire dataset (which might be around 1GB) or at least **bands 1 to 7**. The data provided in the *Featured Data Sources* is .zip archived (remember to extract before use) and cropped to the 3-kilometer spatial extent of the building boundary data (upper left coordinate: 789990.8953, 316815.7082; lower right coordinate: 839412.6408, 266091.7101 in EPSG:6507) (Fig. 1). Once you have downloaded the necessary data from either source add all the seven bands (in .tif files) into your project in QGIS.

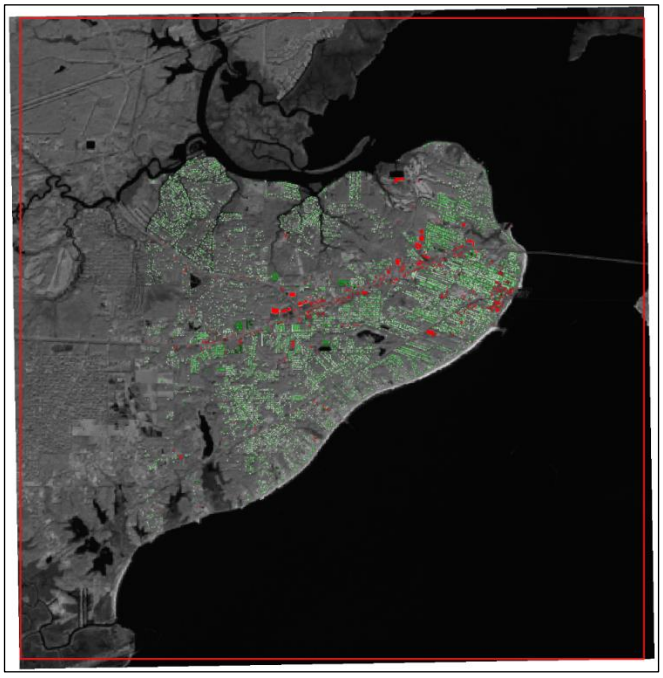

Fig. 1. *Landsat band 5 image against the 3-kilometer buffered building bounding box (in red) and the study area buildings generated in part 2.*

SAGA TOOLS

Some of the QGIS versions do not include *SAGA* tools by default. To verify if this applies to you, launch QGIS and open the *Processing Toolbox.* Navigate through the main level tabs and look for either ***SAGA*** or ***SAGA Next Gen***. If found, you can proceed to the next section. If not, open the *Plugins* menu and select *Manage and Install Plugins*. In the *Installed* tab, verify if the *SAGA* plugin is present. Occasionally, the *SAGA* may not activate by default. If you see the *SAGA* on the list, mark the checkbox next to the plugin, then restart QGIS. If *SAGA* is not visible in the list, switch to the *All* tab and type *saga* in the search box, then select either the ***Processing Saga NextGen Provider*** or, the ***SAGA GIS provider*** (depends on the QGIS version, generally, if you have NextGen available, we recommend choosing it over the legacy tool). Click *Install Plugin* and restart QGIS. The *SAGA* tools should now be present in the *Processing Toolbox*.

UNSUPERVISED MACHINE LEARNING FOR LULC CLASSIFICATION

Once *SAGA* is installed, expand either *SAGA* or *SAGA Next Gen* in the *Processing Toolbox* (use Next Gen if available). Under *Image Analysis* (or *Imagery—Classification* in older versions), select *K-means clustering for grids* (*for Raster* in the older versions). This tool allows you to run unsupervised classification on raster data. This means that the algorithm will grab the data for each pixel and will perform proximity analysis, grouping pixels into classes that have similar parameters. We do not predefine classes, as we let the algorithm define them; however, we will need to provide how many classes we want to use. In our case, a reasonable estimate for this areas' LULC categories is 7 classes representing *water*, *wetland*, *low-density urban*, *high-density urban*, *high vegetation*, *low vegetation*, and *cropland*. This is just our estimation, and the final classes will be determined (but not interpreted!) by the algorithm.

For a first approximation we will use our seven bands from the Landsat image. To do so, select the button on the right of the *grids input* box and select all seven ***Landsat_band*** layers provided with the tutorial (or downloaded manually). Set *clusters* to ***7***. There are three approaches we can use as clustering *Method*s: *minimum distance*, *hill climbing*, or *combined*. The differences between them are presented in the table below (Table 1).

Table 1. Overview of the three methods for unsupervised k-means clustering in SAGA

	Hill-climbing	Minimum-Distance	Combined Approach
Focus	Optimizing the clustering objective function	Assigning points to the nearest centroid	Alternates between optimization and assignment for better clustering
Process	Iterative improvement of the cost function	Simple assignment based on distance	Combines iterative cost minimization with nearest centroid assignment
Cluster Assignment	Adjusts centroids and clusters simultaneously	Assigns points to clusters only	Assign points first, then optimize centroids
Centroid Update	Centroids are adjusted during optimization	Centroids are recalculated after assignment	Centroids are updated iteratively between steps
Objective	Focuses on minimizing the overall cost	Focuses on minimizing local distances	Focuses on both global cost and local assignments
Risk of Local Minima	High, as it depends on initialization	Does not explicitly address global optimization	Balances between local and global optimization but still sensitive to initialization
Primary Role	Strategy for overall optimization	Assignment step within clustering	The full iterative process of K-means
Convergence	May stagnate in a local minimum	Stops when assignments stabilize	Typically faster and more robust than hill-climbing alone
Complexity	More computationally expensive	Less computationally intensive	Balanced; computational cost depends on iterations
Initialization Sensitivity	Highly sensitive to initial centroids	Sensitive to centroid initialization indirectly	Improved by using smarter initialization like K-means++
Strengths	Finds better cluster configurations when successful	Simple and efficient for assigning points	Efficient and effective, combining strengths of both methods
Weaknesses	Risk of suboptimal results due to local minima	No global optimization, just local assignments	Still requires good initialization and is prone to spherical cluster assumptions

Given the differences between all approaches, let's change the *method* to ***Combined*** and set the *maximum iterations* to a higher number, e.g., ***30***. Mark the option to ***normalize the inputs***. Ensure the *start partition* is set to ***random***, and you are not using the old version (SAGA Next Gen only). Once all is set (Fig. 2), click *Run*. Depending on the speed of your computer and the size of the input raster (unclipped raster will process slower), the process might take from a few seconds to a couple of minutes.

Once the classification algorithm is done, you will need to compare the resulting clusters against a background layer like *Google Satellite* or *OpenStreetMap* to determine which category represents type of LULC class. Remember, that in unsupervised learning, classes are generated based on the similarity and are not interpreted. It is up to you, as the analyst, to determine what each generated class represents. You should also set the color and label individually each class. To do so, right-click in the classified raster and select *Properties*. In the *Symbology* tab, change the render type to *Paletted/Unique values* and click *Classify* to generate a random color scheme. Then you can double-click on each color and label to rename it according to the LULC type that you have determined that it represents (example final setup is presented on Fig. 3).

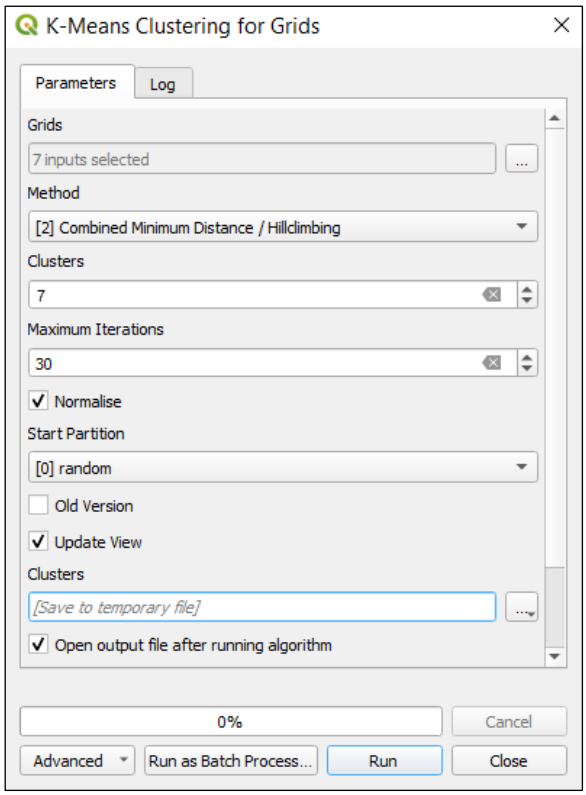

Fig. 2. *Settings used for unsupervised classification using K-means.*

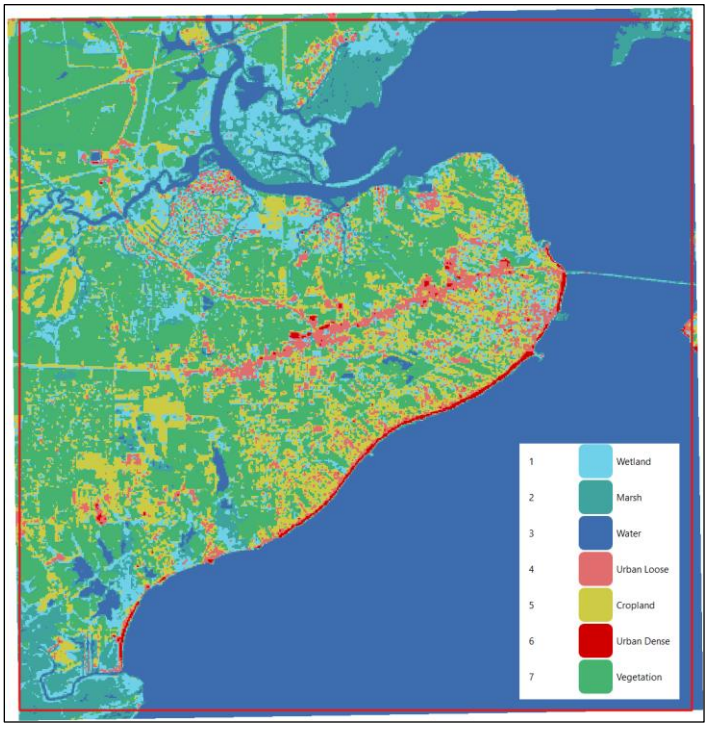

Fig. 3. *Remember that unsupervised classification executes on pixel similarity, therefore what each class represents must be decided manually and color palette assigned accordingly.*

If you are not satisfied with the results, you can rerun the algorithm or change its parameters. Note that **with each run the class numbers won't be consistent**, and you might receive varying results, therefore be cautious when interpreting the raw numbers.

You can compare the quality of the results by opening the *statistics* table that was added to the layers panel with the cluster raster. The goal of the algorithm is to minimize the variance of the output classes; therefore, ideally, the lower the values you see in the *standard deviation* attribute, the better your data is represented. Try modifying the algorithm parameters to see how well you can classify LULC.

CONCLUSION

In this GEO tutorial, we applied unsupervised machine learning using the K-means clustering method to classify LULC in Bay St. Louis, Mississippi. By leveraging multispectral data from Landsat Imagery and the SAGA tools in QGIS, we grouped similar pixel characteristics into distinct classes representing various land cover types. This approach enables a data-driven understanding of the region's landscape, thus providing a foundation for further analysis of flood impacts in the future. The ability to categorize LULC efficiently is essential for informed planning, decision making, and resilience building in communities. This classification process also allows for iterative refinement, thus enhancing accuracy and supporting better decision making in future analyses.