



SMARTFACTOR

BootCamp: zajęcia 13

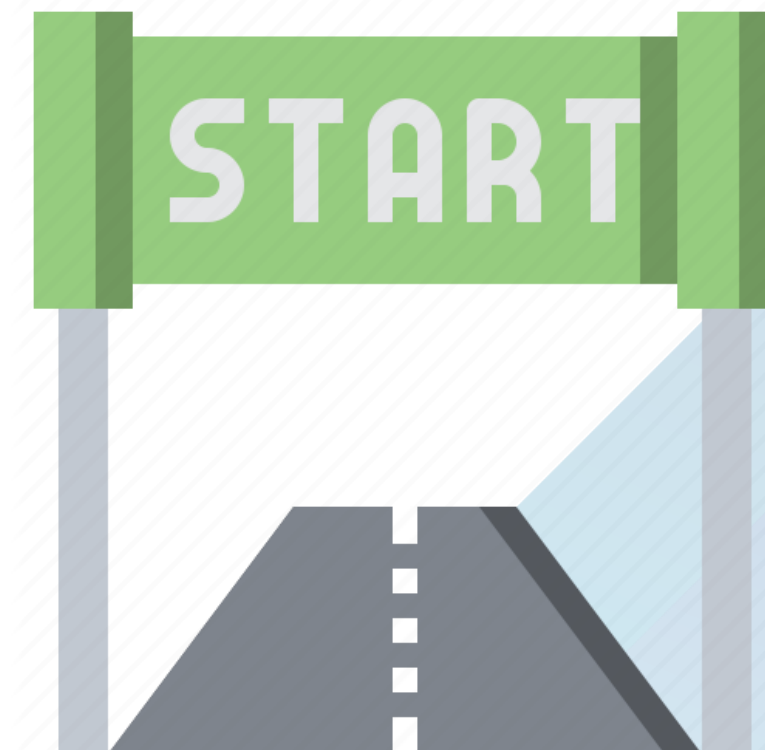
„Nie zawsze potrafię przewidywać,
ale potrafię **kłaść podwaliny**.
Bo **przyszłość** jest czymś,
co się **buduje**”

Antoine de Saint-Exupery

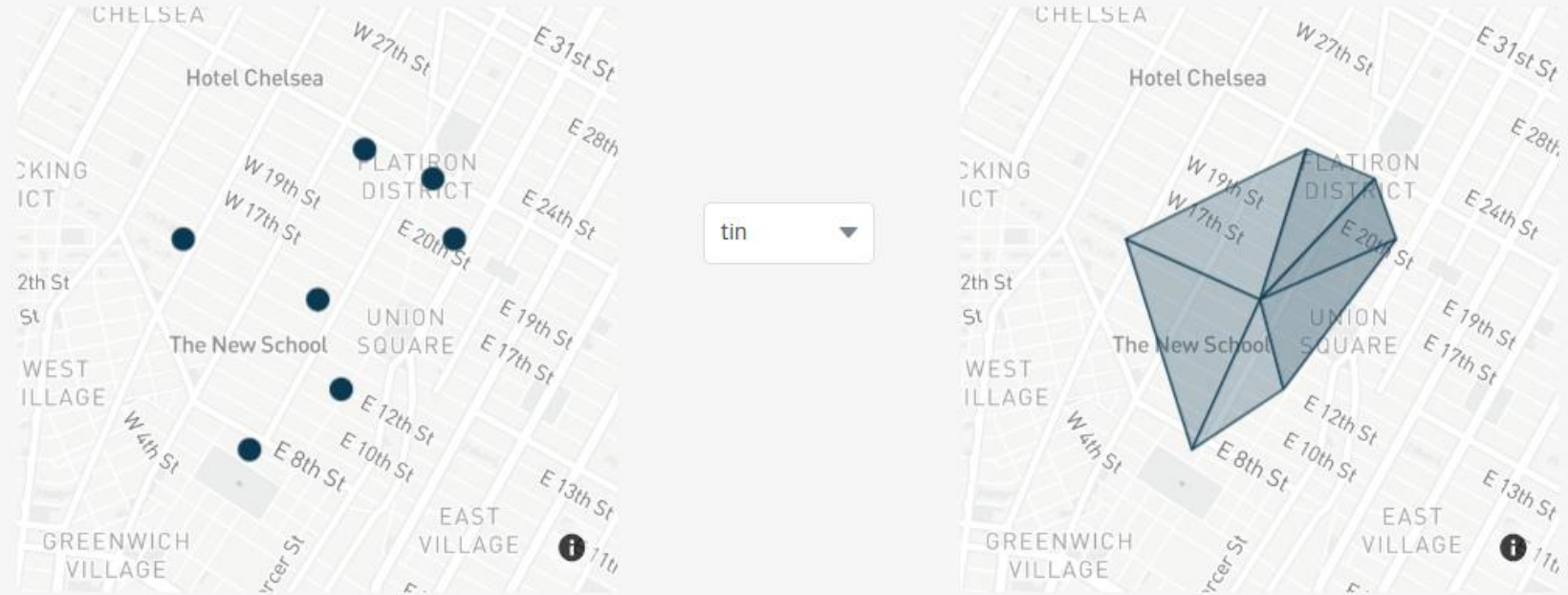


Spis treści

1. Przegląd dokumentacji biblioteki Turf
2. Generowanie losowych punktów
3. Interpolacja
4. Tworzenie bufora
5. Obliczanie przecięcia (*intersect*)
6. Wyznaczanie geometrycznego środka
7. Wyznaczenie fragmentu koła wielkiego
8. Upraszczanie geometrii
9. Wybór punktów przez lokalizację poligonu
10. Pokrycie obszaru siatką kwadratów
11. Klasyfikacja metodą k-średnich
12. Nie tylko Leaflet i Turf



Welcome to Turf.js



tin

Advanced geospatial analysis for browsers and Node.js

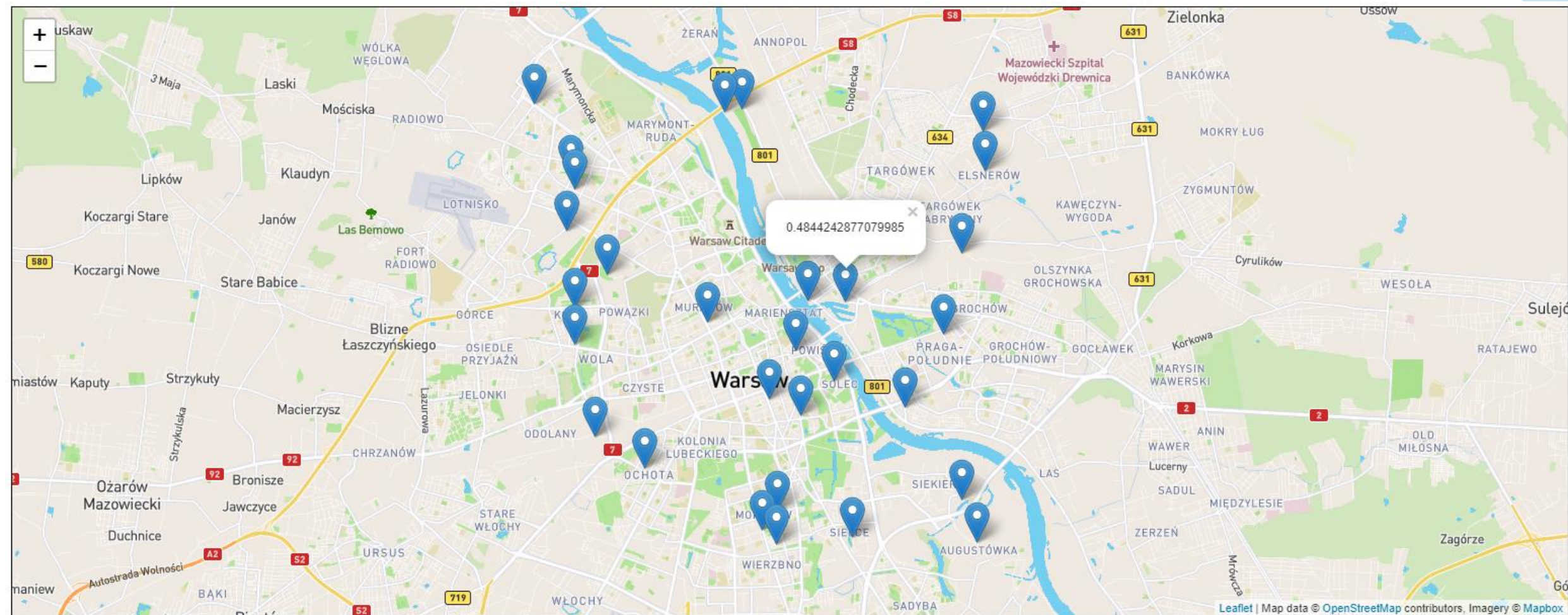
Simple	Modular	Fast
Modular, simple-to-understand JavaScript functions that speak GeoJSON	Turf is a collection of small modules, you only need to take what you want to use	Takes advantage of the newest algorithms and doesn't require you to send data to a server

Generowanie losowych punktów

```
var points = turf.randomPoint(30, {bbox: [20.92950381,52.195038,21.094045,52.295152]});

turf.featureEach(points, function(point) {
  point.properties.solRad = Math.random() * 10;
});

L.geoJSON(points, {
  onEachFeature: function(feature, layer){
    layer.bindPopup('<p>'+feature.properties.solRad+'</p>');
  }
}).addTo(map);
```

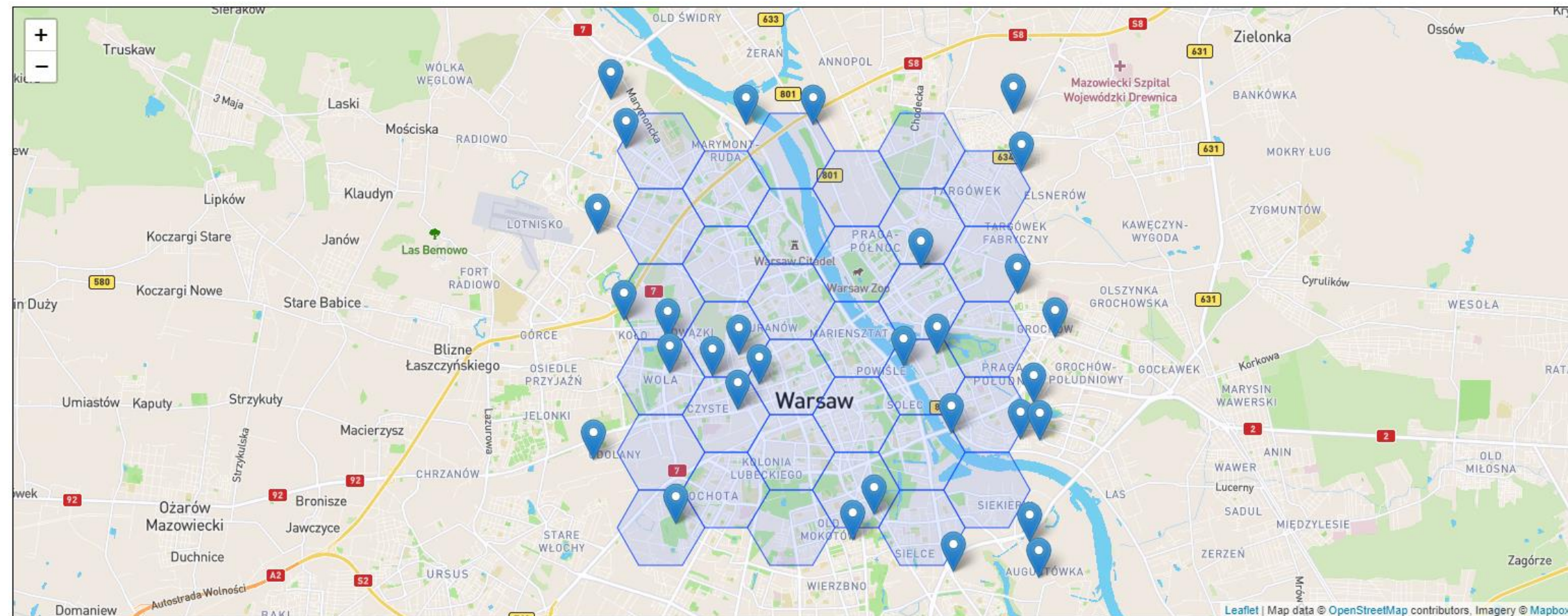


Rys. 1. Losowo wygenerowane punkty

Interpolacja

```
var options = {gridType: 'hex', property: 'solRad', units:
'kilometers'};
var grid = turf.interpolate(points, 1, options);
console.log(grid);
```

```
L.geoJSON(grid, {
  style: function (feature) {
    return {color: '#1a53ff',
      weight: 1,
      fillColor: '#809fff'};
  },
  onEachFeature: function(feature, layer){
    layer.bindPopup('<p>'+feature.properties.solRad+'</p>');
  }
}).addTo(map);
```



Tworzenie bufora

```
L.geoJSON(ptwp, {
  style: function(feature){
    return {color: '#05c9f5', weight: 1}
  }
}).addTo(map);
```

```
var buffered = turf.buffer(ptwp, 0.2, {units: 'kilometers'});
```

```
L.geoJSON(buffered, {
  style: function(feature){
    return {color: '#f58d05', weight: 1}
  }
}).addTo(map);
```



Obliczanie przecięcia (*intersect*)

```
L.geoJSON(rezerwat, {
  style: function(feature){
    return {color: '#046e1d', weight: 1}
  }
}).addTo(map);

L.geoJSON(teren, {
  style: function(feature){
    return {color: '#c95cff', weight: 1}
  }
}).addTo(map);

var intersection = turf.intersect(rezerwat, teren);

L.geoJSON(intersection, {
  style: function(feature){
    return {color: 'red', weight: 3, fillColor: '#f08080'}
  }
}).addTo(map);
```



Rys. 4. Przecięcie (inaczej część wspólna lub ang. *intersection*) dwóch obiektów poligonowych

Wyznaczanie geometrycznego środka

```
L.geoJSON(rezerwat, {  
  style: function(feature){  
    return {color: '#046e1d', weight: 1}  
  }  
}).addTo(map);  
var centroid = turf.centroid(rezerwat);  
L.marker([centroid.geometry.coordinates[1], centroid.geometry.coordinates[0]]).addTo(map);
```

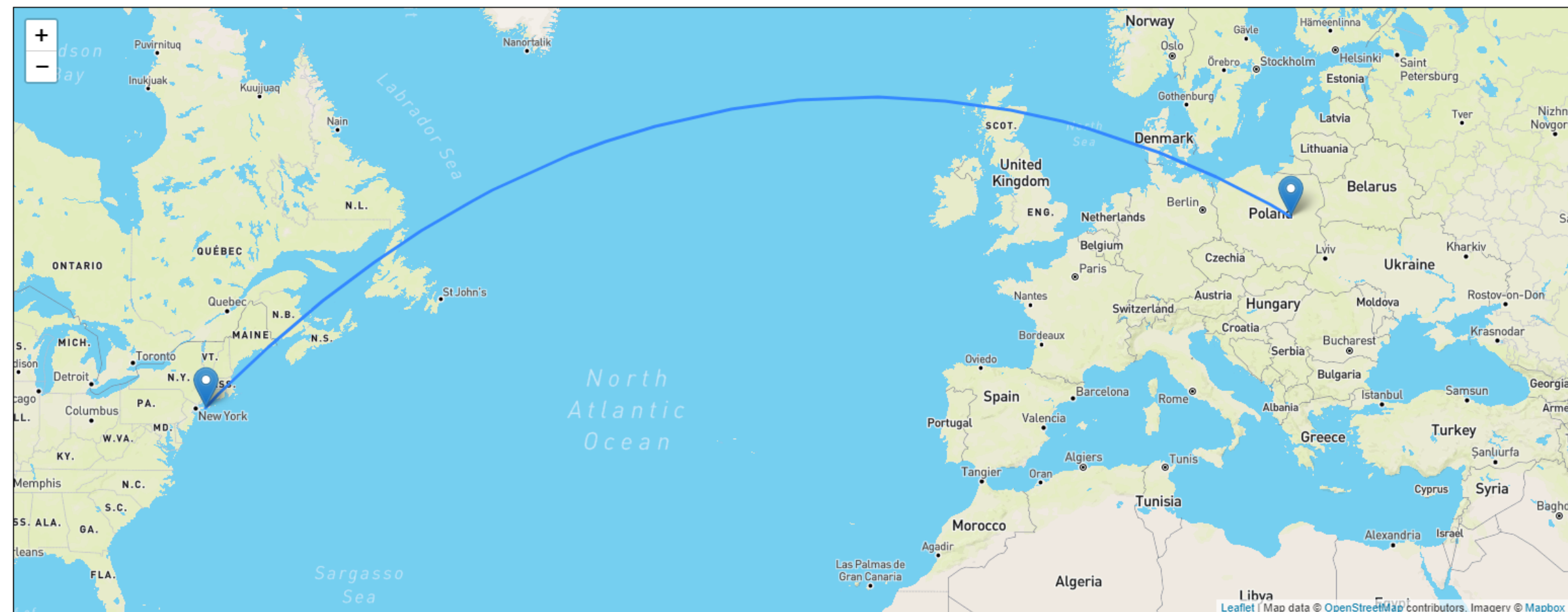


Wyznaczenie fragmentu koła wielkiego

```
var lotniskoChopina = turf.point([20.96637725830078,  
52.167720082876045]);  
var longIslandAirport = turf.point([-73.10062408447266,  
40.79899672300568]);
```

```
L.marker([lotniskoChopina.geometry.coordinates[1],  
lotniskoChopina.geometry.coordinates[0]]).addTo(map);  
L.marker([longIslandAirport.geometry.coordinates[1],  
longIslandAirport.geometry.coordinates[0]]).addTo(map);
```

```
var greatCircle = turf.greatCircle(lotniskoChopina, longIslandAirport);  
L.geoJSON(greatCircle).addTo(map);
```



Upraszczenie geometrii

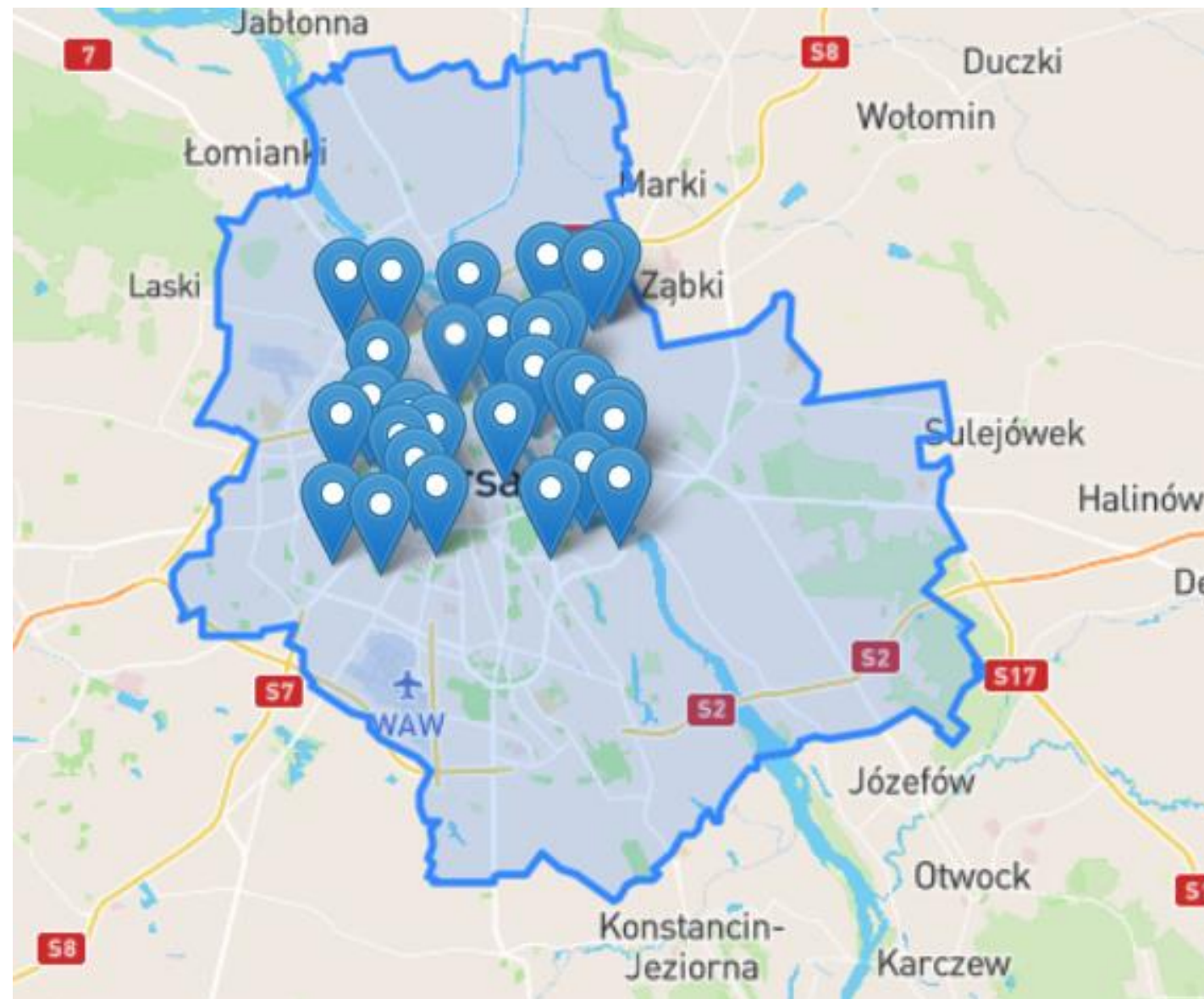
```
L.geoJSON(rezerwat, {
  style: function(feature){
    return {color: '#046e1d', weight: 1}
  }
}).addTo(map);
var options = {tolerance: 0.001, highQuality: true};
var simplified = turf.simplify(rezerwat, options);
L.geoJSON(simplified, {
  style: function(feature){
    return {color: '#26d902', weight: 1}
  }
}).addTo(map);
```



Rys. 7. Oryginalny oraz uproszczony poligon

Wybór punktów przez lokalizację poligonu

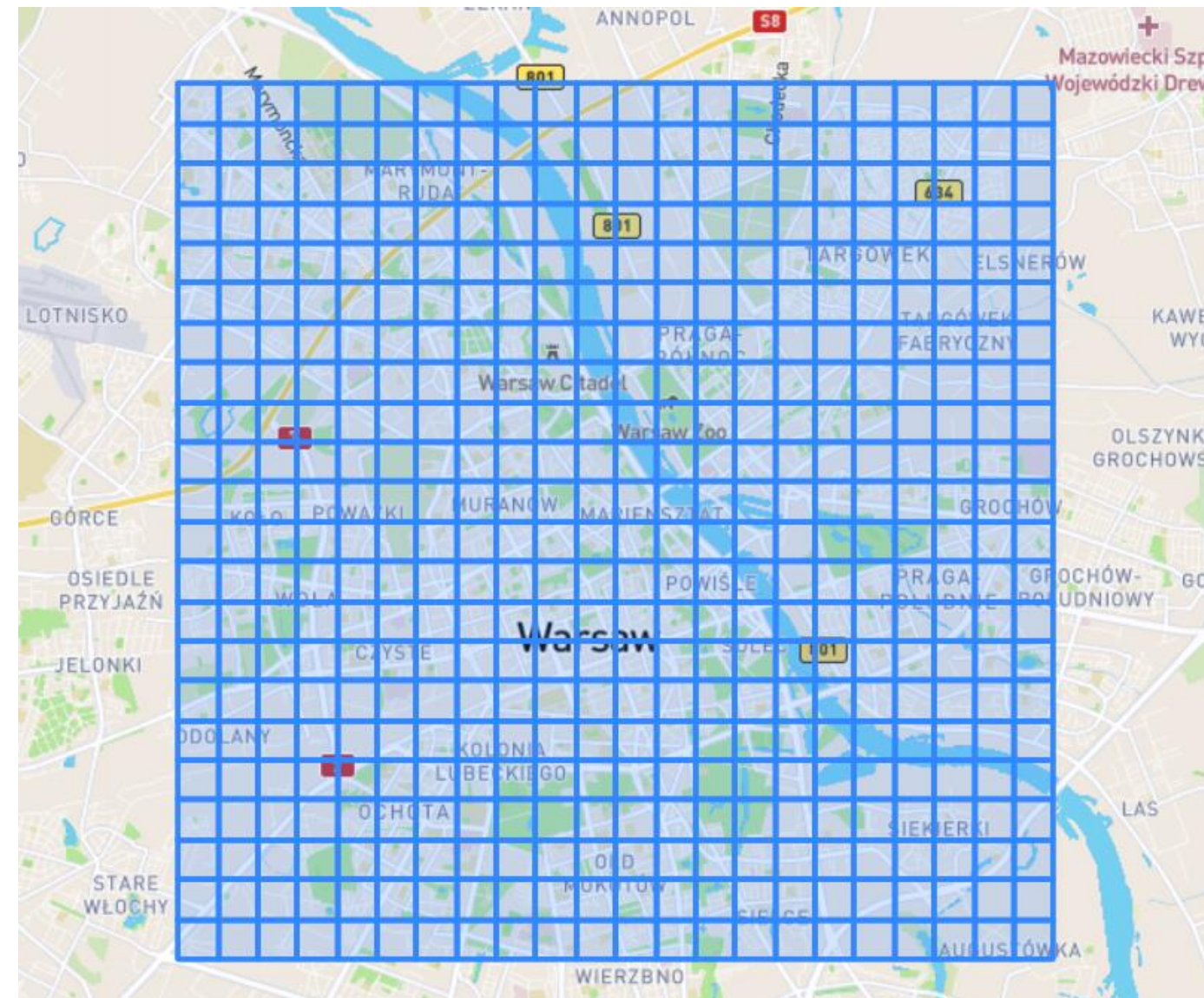
```
var points = turf.randomPoint(30, {bbox: [20.92950381,52.195038,21.094045,52.295152]});  
var ptsWithin = turf.pointsWithinPolygon(points, warszawa);  
L.geoJSON(warszawa).addTo(map);  
L.geoJSON(ptsWithin).addTo(map);
```



Rys. 8. Wybór punktów znajdujących się wewnątrz poligonu

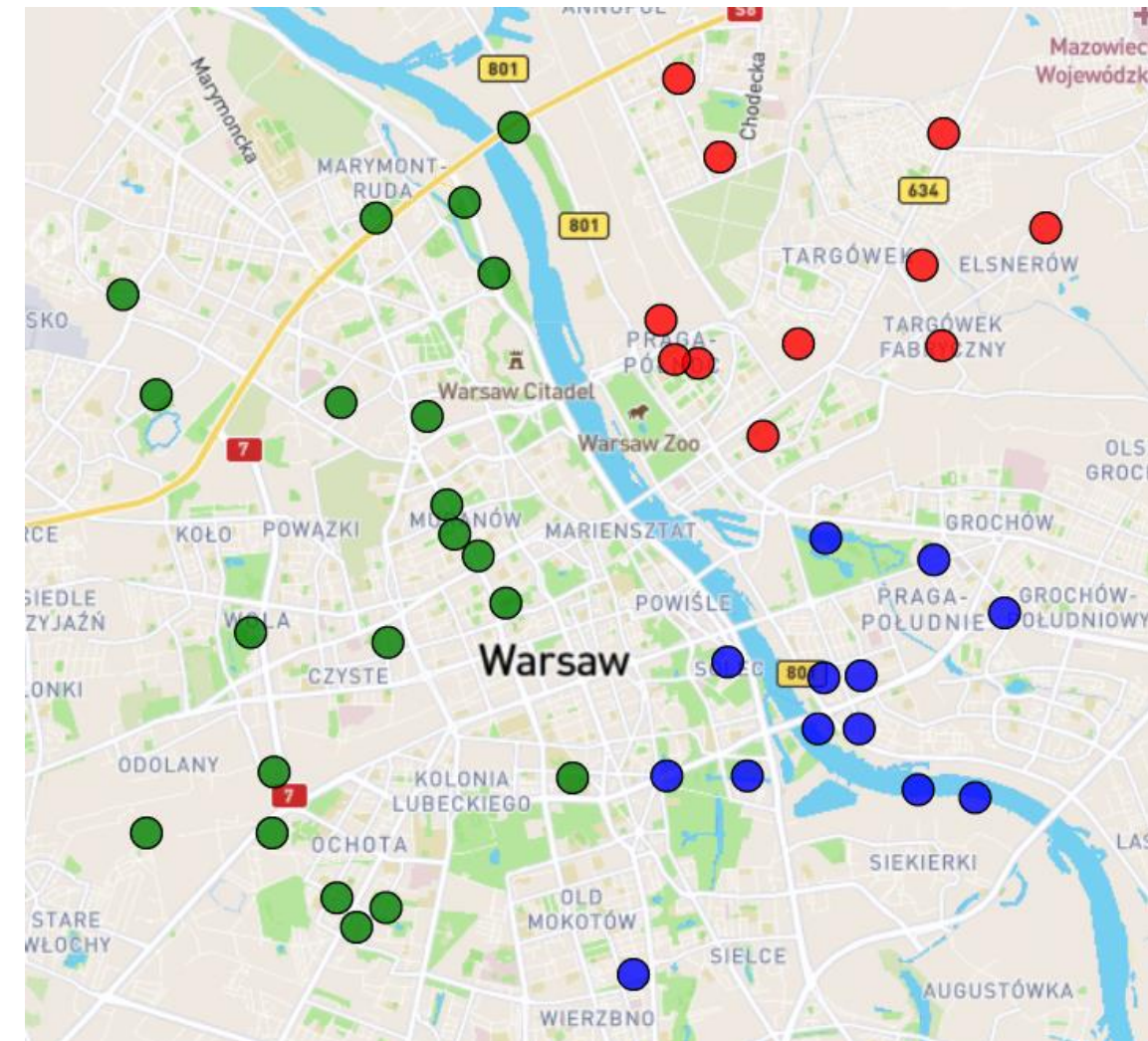
Pokrycie obszaru siatką kwadratów

```
var bbox = [20.92950381,52.195038,21.094045,52.295152];  
var cellSide = 0.5;  
var options = {units: 'kilometers'};  
  
var squareGrid = turf.squareGrid(bbox, cellSide, options);  
L.geoJSON(squareGrid).addTo(map);
```



Klasyfikacja metodą k-średnich

```
var points = turf.randomPoint(45, {bbox: [20.92950381, 52.195038, 21.094045, 52.295152]});  
var options = {numberOfClusters: 3};  
var clustered = turf.clustersKmeans(points, options);  
var clusterColors = {0:'red', 1:'blue', 2:'green'};  
L.geoJSON(clustered, {  
  pointToLayer: function (feature, latlng) {  
    return L.circleMarker(latlng, {  
      radius: 8,  
      fillColor: clusterColors[feature.properties.cluster],  
      color: 'black',  
      weight: 1,  
      opacity: 1,  
      fillOpacity: 0.8  
    });  
  }  
}).addTo(map);
```



Rys. 10. Klastry wykryte metodą k-średnich

Nie tylko Leaflet i Turf

- [Animacje z OpenLayers](#)
- [Openlayers - Geospatial JavaScript – podcast](#)
- [10 open source projects every JavaScript geo dev should know about – blog](#)
- [Calcite Maps](#)
- [Terraformer](#)
- [Mapbox](#)
- [JavaScript for Geospatial applications: An Overview](#)

Miejsce na notatki

A large, empty rectangular area with a dashed blue border, intended for taking notes.

Przydatne linki

- [1] [Losowe punkty](#)
- [2] [Interpolacja](#)
- [3] [Bufor](#)
- [4] [Przecięcie](#)
- [5] [Centroid](#)
- [6] [Koło wielkie](#)
- [7] [Usuwanie powtarzających się współrzędnych](#)
- [8] [Upraszczenie geometrii](#)
- [9] [Punkty wewnątrz poligonu](#)
- [10] [Pokrycie obszaru siatką kwadratów](#)
- [11] [Klasyfikacja metodą k-średnich](#)
- [12] [geoJSON z metodą pointToLayer](#)



SMARTFACTOR



+48 798 622 487



ul. Poselska 29
03-931 Warszawa



mail@smartfactor.pl



www.smartfactor.pl