

# ANALIZY PRZESTRZENNE

## WSKAZANIE OPTYMALNEJ LOKALIZACJI FARMY FOTOWOLTAICZNEJ – ANALIZY WIELOKRYTERIALNE

JULIA GOMULSKA  
328936

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Wykonanie ćwiczenia i realizacja skryptu</b>	<b>2</b>
2.1	Ustawienie środowiska i przygotowanie potrzebnych plików . . . . .	2
2.2	Kryterium 1. . . . .	3
2.3	Kryterium 2. . . . .	5
2.4	Kryterium 3. . . . .	6
2.5	Kryterium 4. . . . .	7
2.6	Kryterium 5. . . . .	9
2.7	Kryterium 6. . . . .	9
2.8	Kryterium 7. . . . .	10
2.9	Ocena przydatności terenu . . . . .	11
2.10	Wybranie przydatnych działek/grup działek . . . . .	16
2.11	Koszt przyłącza do sieci . . . . .	20
<b>3</b>	<b>Podsumowanie i wnioski</b>	<b>26</b>
<b>4</b>	<b>Przykład nowego, nieuwzględnionego kryterium</b>	<b>27</b>
<b>5</b>	<b>Testu wykonania modelu na danych z innego obszaru</b>	<b>27</b>
5.1	Opis obszaru . . . . .	27
5.2	Otrzymane wyniki . . . . .	28
5.2.1	Kryterium 1. . . . .	28
5.2.2	Kryterium 2. . . . .	28
5.2.3	Kryterium 3. . . . .	29
5.2.4	Kryterium 4. . . . .	29
5.2.5	Kryterium 5. . . . .	30
5.2.6	Kryterium 6. . . . .	30
5.2.7	Kryterium 7. . . . .	31
5.2.8	Ocena przydatności terenu . . . . .	31
5.2.9	Wybranie przydatnych działek/grup działek . . . . .	32
5.2.10	Koszt przyłącza do sieci . . . . .	33
5.3	Zauważone błędy i problemy . . . . .	35

# 1 Wstęp

Celem projektu było wyznaczenie optymalnej lokalizacji farmy fotowoltaicznej (z wykorzystaniem analizy wielokryterialnej) w gminie Świeradów Zdrój w powiecie lubańskim w województwie dolnośląskim oraz stworzenie skryptu do automatyzacji tego procesu.

Przy wyborze lokalizacji farmy fotowoltaicznej należy wziąć pod uwagę ukształtowanie terenu oraz dostęp do promieni słonecznych. W Polsce zaleca się skierowanie paneli w stronę południową oraz usytuowanie ich na płaskich terenach. Należy zbadać też położenie względem lasów, budynków, słupów by wyeliminować tereny, gdzie mogą wystąpić zacienienia ograniczające dostęp do energii słonecznej. Powinno zbadać się również powierzchnię danych działek, ponieważ działka powinna być tak duża, by ustawione w rzędach panele się nie zasłaniały. Inwestycja powinna być zgodna z miejscowym planem zagospodarowania przestrzennego lub (gdy teren nie jest nim objęty) trzeba sprawdzić, czy w danej gminie instalacje fotowoltaiczne są zaliczane jako inwestycje celu publicznego. Działka pod lokalizację farmy fotowoltaicznej powinna być położona blisko linii energetycznej (maksymalnie 200m odległości), by nie zwiększać wzrostu kosztów inwestycji. Farmy o powierzchni 0.5-1 ha uznawane są za obszary mogące wpływać na środowisko i w niektórych przypadkach wymagają uzyskania decyzji środowiskowej. W związku z tym należy wziąć pod uwagę odległość farmy od użytków ekologicznych, parków narodowych i krajobrazowych, rezerwatów przyrody, obszarach Natura 2000 <sup>1</sup>

## 2 Wykonanie ćwiczenia i realizacja skryptu

Do wykonania ćwiczenia niezbędne były warstwy wektorowe z BDOT10k dla powiatu lubańskiego i lwóweckiego, połączone sekcje NMT, warstwy wektorowe działek dla dwóch powiatów pochodzące z EGIBu oraz warstwa wektorowa zawierająca bufor wokół gminy.

### 2.1 Ustawienie środowiska i przygotowanie potrzebnych plików

Kod wczytuje odpowiednie biblioteki. Następnie ustawia środowisko pracy: folder odczytu i zapisu danych, układ współrzędnych - EPSG 2180, zasięg i maskę na podstawie warstwy wektorowej zawierającej bufor wokół gminy, rozdzielczość, nadpisywanie plików. Przeszukuje pliki w folderze workspace, usuwając kropki z nazw plików, aby uniknąć problemów z ich odczytem. Kolejno identyfikuje warstwy wektorowe na podstawie kodów TERYT oraz unikalnych fragmentów nazw, przypisując każdą znalezioną warstwę do odpowiednich zmiennych reprezentujących dane z powiatów lubańskiego i lwóweckiego. Wczytuje Numeryczny Model Terenu (NMT).

**Kod realizujący ustawienie środowiska i przygotowanie plików :**

Import bibliotek

```
1 import arcpy
2 import os
```

Ustawienie środowiska

```
1 arcpy.env.workspace = "C:\studia\sem5\AnalizyPrzestrzenne\Proj1\warstwy"
2 arcpy.env.outputCoordinateSystem = arcpy.SpatialReference("ETRF2000 -
  PL_CS92")
3 arcpy.env.extent = "C:\studia\sem5\AnalizyPrzestrzenne\Proj1\warstwy\
  A03_Granice_gmin_1992_Buffer.shp"
4 arcpy.env.mask = "C:\studia\sem5\AnalizyPrzestrzenne\Proj1\warstwy\
  A03_Granice_gmin_1992_Buffer.shp"
5 arcpy.env.cellSize = 5
6 arcpy.env.overwriteOutput = True
```

Usunięcie kropek z nazw plików

```
1 directory1 = arcpy.env.workspace
2 teryt1 = "0210"
3 teryt2 = "0212"
4
```

<sup>1</sup> <https://hymon.pl/wymogi-jakie-musi-splniac-dzialka-pod-fotowoltaike/>

```

5
6 for filename in os.listdir(directory1):
7     name, extension = os.path.splitext(filename)      #rozdzielenie nazwy
8     pliku na nazwę i rozszerzenie
9     if '.' in name:      #sprawdzenie czy nazwa pliku zawiera kropkę
10        new_name = name.replace('.', '')      #usunięcie kropki z nazwy
11        pliku
12        new_filename = new_name + extension #stworzenie nowej nazwy
13        pliku
14        old_file = os.path.join(directory1, filename) #ścieżka do
15        starego pliku
16        new_file = os.path.join(directory1, new_filename) #ścieżka do
17        nowego pliku
18        if not os.path.exists(new_file):      #sprawdzenie czy plik o
19        nowej nazwie już istnieje
20        os.rename(old_file, new_file)      #zmiana nazwy pliku

```

Znalezienie warstw potrzebnych w projekcie i przypisanie ich do odpowiednich zmiennych

```

1
2 for filename in os.listdir(directory1): #przeszukanie folderu z
3     warstwami
4     if teryt1+"__OT_PTWP" in filename and filename.endswith(".shp"):
5         #sprawdzenie czy nazwa pliku zawiera TERYT i OT_PTWP oraz ko
6        ńczy się na .shp
7         ptwp_lubanski = filename      #przypisanie nazwy pliku do
8         zmiennej
9     elif teryt2+"__OT_PTWP" in filename and filename.endswith(".shp"):
10        #analogicznie jak wyżej
11        ptwp_lwowecki = filename
12    elif teryt1+"__OT_SWRS" in filename and filename.endswith(".shp"):
13        swrs_lubanski = filename
14    elif teryt2+"__OT_SWRS" in filename and filename.endswith(".shp"):
15        swrs_lwowecki = filename
16    elif teryt1+"__OT_BUBD" in filename and filename.endswith(".shp"):
17        bubd_lubanski = filename
18    elif teryt2+"__OT_BUBD" in filename and filename.endswith(".shp"):
19        bubd_lwowecki = filename
20    elif teryt1+"__OT_PTLZ" in filename and filename.endswith(".shp"):
21        ptlz_lubanski = filename
22    elif teryt2+"__OT_PTLZ" in filename and filename.endswith(".shp"):
23        ptlz_lwowecki = filename
24    elif teryt1+"__OT_SKDR" in filename and filename.endswith(".shp"):
25        skdr_lubanski = filename
26    elif teryt2+"__OT_SKDR" in filename and filename.endswith(".shp"):
27        skdr_lwowecki = filename
28    elif teryt1+"__OT_SKJZ" in filename and filename.endswith(".shp"):
29        skjz_lubanski = filename
30    elif teryt2+"__OT_SKJZ" in filename and filename.endswith(".shp"):
31        skjz_lwowecki = filename
32    elif teryt1+"__OT_SKRW" in filename and filename.endswith(".shp"):
33        skrw_lubanski = filename
34    elif teryt2+"__OT_SKRW" in filename and filename.endswith(".shp"):
35        skrw_lwowecki = filename
36    elif teryt1+"__OT_SULN" in filename and filename.endswith(".shp"):
37        suln_lubanski = filename
38    elif teryt2+"__OT_SULN" in filename and filename.endswith(".shp"):
39        suln_lwowecki = filename
40
41 nmt = r"C:\studia\sem5\AnalizyPrzestrzenne\Proj1\NMT\NMTCal.tif"

```

## 2.2 Kryterium 1.

Kryterium polegało na wyborze terenów położonych jak najbliżej rzek i zbiorników wodnych, ale z zachowaniem stumetrowej strefy ochronnej. Przyjęto, że optymalną odległością będzie od 102

do 200 metrów – tereny leżące w takiej odległości otrzymają najwyższą wartość funkcji przydatności (równą 1). Za przydatne zostają uznane także obszary w odległości 100-102 metry (w tym przedziale funkcja przydatności liniowo rośnie osiągając wartości między 0 a 1) oraz w odległości 200-1000 metrów (w tym przedziale funkcja przydatności liniowo maleje osiągając wartości między 0 a 1). Pozostałe tereny są uznawane za nieprzydatne i uzyskują wartość funkcji przydatności równą 0. Aby wyznaczyć tereny spełniające kryterium 1 należało pozyskać dane z dwóch warstw wektorowych: SWRS oraz PTWP dla dwóch powiatów. Następnie połączono warstwy rzek oraz wód powierzchniowych z dwóch powiatów za pomocą narzędzia *Merge*. Aby połączyć warstwę rzek i wód powierzchniowych, trzeba było doprowadzić obie warstwy do takiej samej geometrii. W tym celu wykonano centymetrowy bufor (narzędzie *Buffer*) wokół rzek zmieniając tym samym geometrię z liniowej na poligonową. Następnie połączono warstwę wód powierzchniowych i zbuforowaną warstwę rzek w warstwę reprezentującą wszystkie wody. W kolejnym kroku stworzono mapę odległości od wód (korzystając z narzędzia *EucDistance*), funkcje przydatności (rosnącą i malejącą) z wartościami wspomnianymi wyżej (korzystając z narzędzia *FuzzyMembership*) oraz połączono je tworząc mapę przydatności (wykorzystując narzędzie *FuzzyOverlay*). Zapisano powstałą mapę przydatności do nowego pliku rastrowego.

### Kod realizujący kryterium 1:

Złączenie warstw z dwóch powiatów w jedną dla rzek i wód powierzchniowych

```
1 arcpy.management.Merge(
2     inputs=[swrs_lubanski, swrs_lwowecki],
3     output="swrs_merge"
4 )
5 arcpy.management.Merge(
6     inputs=[ptwp_lwowecki, ptwp_lubanski],
7     output="ptwp_merge"
8 )
```

Zmiana geometrii rzek z linii na poligony poprzez stworzenie centymetrowego bufora, by warstwa rzek i wód powierzchniowych miała tę samą geometrię

```
1 arcpy.analysis.Buffer(
2     in_features="swrs_merge",
3     out_feature_class="swrs_poly",
4     buffer_distance_or_field="1 Centimeters",
5 )
```

Złączenie warstw rzek i wód powierzchniowych w jedną warstwę wód

```
1 arcpy.management.Merge(
2     inputs=["swrs_poly", "ptwp_merge"],
3     output="wody"
4 )
```

Stworzenie mapy odległości od wód powierzchniowych

```
1 eucDist_wody = arcpy.sa.EucDistance(
2     in_source_data="wody",
3     cell_size=5)
```

Stworzenie funkcji przydatności

```
1 wody_frosnaca = arcpy.sa.FuzzyMembership(
2     in_raster=eucDist_wody,
3     fuzzy_function = "LINEAR 100 102"
4 )
5
6 wody_fmalejaca = arcpy.sa.FuzzyMembership(
7     in_raster=eucDist_wody,
8     fuzzy_function = "LINEAR 1000 200 "
9 )
```

Stworzenie mapy przydatności dla kryterium 1 poprzez połączenie funkcji przydatności i jej zapis do nowego pliku

```

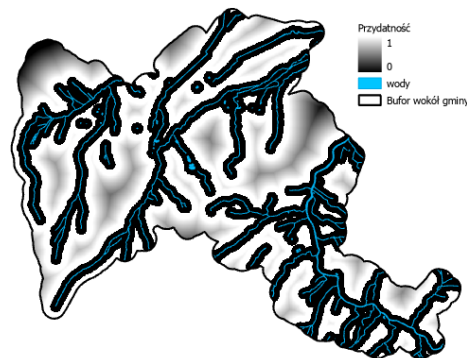
1 mapa_przydatnosci_wody = arcpy.sa.FuzzyOverlay(
2     in_rasters=[wody_frosnaca,wody_fmalejaca],
3     overlay_type="AND")
4 mapa_przydatnosci_wody.save("kryterium1.tif")

```

Otrzymano następujące wyniki:



Mapa przydatności dla kryterium 1.



Wody na tle mapy przydatności dla kryterium 1.

Rysunek 1: Wyniki dla kryterium 1.

Widać, że mapa przydatności została wygenerowana poprawnie. Obszary leżące nad wodą oraz daleko od wody mają przydatność 0, obszary leżące w odległości 102-200 mają największą przydatność. Obszary leżące w odległości 100-102 m oraz 200-1000 m mają mniejszą przydatność, ale nadal są uznawane za przydatne. Uzyskuje się stosunkowo dużo obszarów przydatnych.

## 2.3 Kryterium 2.

Kryterium polegało na wyborze obszarów, które leżą w jak największej odległości od budynków mieszkalnych, przy czym minimalna odległość wynosi 150m. W celu realizacji tego kryterium pobrano warstwę budynków dla dwóch powiatów oraz połączono je w jedną (używając do tego narzędzia *Merge*). Następnie wybrano z niej tylko budynki mieszkalne. W tym celu skorzystano z narzędzia *SelectLayerByAttribute* oraz atrybutu *FOBUD* mówiącego o funkcji ogólnej budynku, który dla budynków mieszkalnych ma wartość *budynki mieszkalne*. Wyselekcjonowane budynki zapisano do nowej warstwy wektorowej (narzędzie *CopyFeatures*). Stworzono mapę odległości od budynków mieszkalnych (narzędzie *EucDistance*) oraz funkcję przydatności (narzędzie *FuzzyMembership*). Funkcja ta przyporządkowuje terenom położonym w odległości poniżej 150m zerową przydatność, dla obszarów położonych w odległości 150-1000m przydatność liniowo rośnie, osiągając wartość 1 dla obszarów położonych w odległości min. 1000m. Wynik funkcji reprezentujący przydatność terenu został zapisany do nowego pliku rastrowego.

### Kod realizujący kryterium 2:

Złączenie warstw budynków z dwóch powiatów w jedną

```

1 arcpy.management.Merge(
2     inputs=[bubd_lwowecki,bubd_lubanski],
3     output="bubd_merge"
4 )

```

Wybranie budynków mieszkalnych

```

1 bubd_selection= arcpy.management.SelectLayerByAttribute(
2     in_layer_or_view="bubd_merge",
3     where_clause="FOBUD LIKE 'budynki mieszkalne'"
4 )
5 arcpy.management.CopyFeatures(bubd_selection, 'bubd_mieszkalne')

```

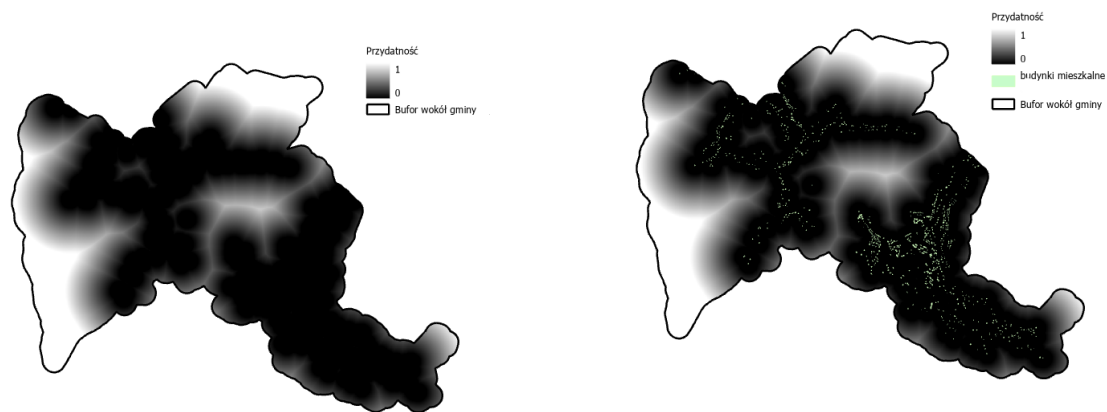
Stworzenie mapy odległości od budynków mieszkalnych

```
1 eucDist_budynki = arcpy.sa.EucDistance(  
2     in_source_data="budd_mieszkalne",  
3     cell_size=5)
```

Stworzenie funkcji przydatności i zapis do nowego pliku

```
1 budynki_froznaca = arcpy.sa.FuzzyMembership(  
2     in_raster=eucDist_budynki,  
3     fuzzy_function="LINEAR 150 1000"  
4 )  
5 budynki_froznaca.save("kryterium2.tif")
```

Otrzymano następujące wyniki:



Mapa przydatności dla kryterium 2.

Budynki mieszkalne na tle mapy przydatności dla kryterium 2.

Rysunek 2: Wyniki dla kryterium 2.

Mapa przydatności generuje się zgodnie z oczekiwaniami. Tereny w pobliżu budynków mieszkalnych są uznawane za nieprzydatne. Im dalej od budynków mieszkalnych tym większa przydatność terenu. Kryterium to wyklucza sporą część terenu położoną w południowo-wschodniej oraz centralnej części gminy.

## 2.4 Kryterium 3.

Kryterium polegało na wyborze terenów, które leżą min. 15m. od lasów, a optymalnie w odległości powyżej 100 m. W celu realizacji tego kryterium połączono warstwę pokrycia terenu zawierającą tereny leśne i zadrzewione dla dwóch powiatów oraz połączono je w jedną (używając do tego narzędzia *Merge*). Korzystając z narzędzia *SelectLayerByAttribute* oraz atrybut *RODZAJ*, który dla lasów przyjmuje wartość *las* wyselekcjonowano same lasy i zapisano je do nowej warstwy wektorowej (narzędzie *CopyFeatures*). Następnie stworzono mapę odległości od lasów (narzędzie *EucDistance*) oraz funkcję przydatności (narzędzie *FuzzyMembership*). Stworzona funkcja przypisuje terenom położonym w odległości poniżej 15m zerową przydatność, dla obszarów położonych w odległości 15-100m przydatność liniowo rośnie, osiągając wartość 1 dla obszarów położonych w odległości min. 100m. Wynik funkcji reprezentujący przydatność terenu został zapisana do nowego pliku rastrowego.

Kod realizujący kryterium 3:

Złączenie odpowiednich warstw z dwóch powiatów w jedną

```
1 arcpy.management.Merge(  
2     inputs=[ptlz_lwowecki,ptlz_lubanski],  
3     output="ptlz_merge"  
4 )
```

Wyselekcjonowanie lasów i zapis do nowej warstwy wektorowej

```
1 ptlz_selection = arcpy.management.SelectLayerByAttribute(  
2     in_layer_or_view="ptlz_merge",  
3     where_clause="RODZAJ = 'las'"  
4 )  
5 arcpy.management.CopyFeatures(ptlz_selection, 'ptlz_lasy')
```

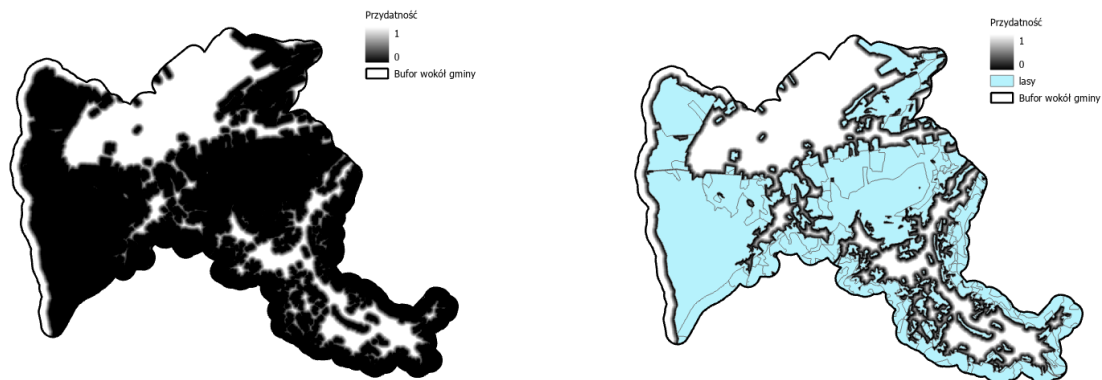
Stworzenie mapy odległości od lasów

```
1 eucDist_lasy = arcpy.sa.EucDistance(  
2     in_source_data="ptlz_lasy",  
3     cell_size=5)
```

Stworzenie funkcji przydatności i zapis do nowego pliku

```
1 lasy_froznaca = arcpy.sa.FuzzyMembership(  
2     in_raster=eucDist_lasy,  
3     fuzzy_function="LINEAR 15 100"  
4 )  
5  
6 lasy_froznaca.save("kryterium3.tif")
```

Otrzymano następujące wyniki:



Mapa przydatności dla kryterium 3.

Lasy na tle mapy przydatności dla kryterium 3.

Rysunek 3: Wyniki dla kryterium 3.

Widać, że mapa przydatności została wygenerowana poprawnie. Poblże lasów sprawia, że teren jest uznawany za nieprzydatny. Można zauważyć, że tereny uznane jako przydatne w kryterium 3. w dużej części pokrywają się z obszarami nieprzydatnymi z kryterium 2. Tereny położone w północnej i południowo-wschodniej części gminy odznaczają się największą przydatnością.

## 2.5 Kryterium 4.

Kryterium polegało na wskazaniu terenów, które dysponują dobrym dostępem do dróg utwardzonych. W celu realizacji tego kryterium połączono warstwę dróg dla dwóch powiatów oraz połączono je w jedną (używając do tego narzędzia *Merge*). Z nowo powstałej warstwy wybrano drogi utwardzone i zapisano je do nowej warstwy wektorowej. Selekcja odpowiednich dróg odbyła się z wykorzystaniem narzędzia *SelectLayerByAttribute* i atrybutu *MATE\_NAWIE*, który określa materiał nawierzchni drogi i dla dróg utwardzonych przyjmuje następujące wartości: *beton*, *bruk*, *kostka kamienna*, *kostka prefabrykowana*, *masa bitumiczna*, *plyty betonowe*. Następnie obliczono gęstość dróg utwardzonych na  $m^2$  narzędziem *LineDensity*. Obliczono maksymalną gęstość dróg utwardzonych na obszarze analizowanej gminy, by wykorzystać tę wartość przy przeskalowywaniu wartości. Korzystając z narzędzia *RescaleByFunction* przeskalowano wartości do zakresu 0-1 i zapisano raster do nowego pliku.

Kod realizujący kryterium 4:



Złączenie odpowiednich warstw z dwóch powiatów w jedną

```
1 arcpy.management.Merge(  
2     inputs=[skdr_lwowecki, skdr_lubanski],  
3     output="skdr_merge"  
4 )
```

Wybranie dróg utwardzonych i zapis ich do nowej warstwy wektorowej

```
1 skdr_selection= arcpy.management.SelectLayerByAttribute(  
2     in_layer_or_view="skdr_merge",  
3     where_clause = "MATE_NAWIE IN ('beton','bruk', 'kostka kamienna', '  
4         kostka prefabrykowana', 'masa bitumiczna', 'plyty betonowe')"  
5 )  
6 arcpy.management.CopyFeatures(skdr_selection, 'skdr_utwardzone')
```

Obliczenie gęstości dróg

```
1 line_den_skdr = arcpy.sa.LineDensity(  
2     in_polyline_features="skdr_utwardzone",  
3     population_field="NONE",  
4     cell_size=5,  
5     search_radius=1000,  
6     area_unit_scale_factor="SQUARE_METERS"  
7 )
```

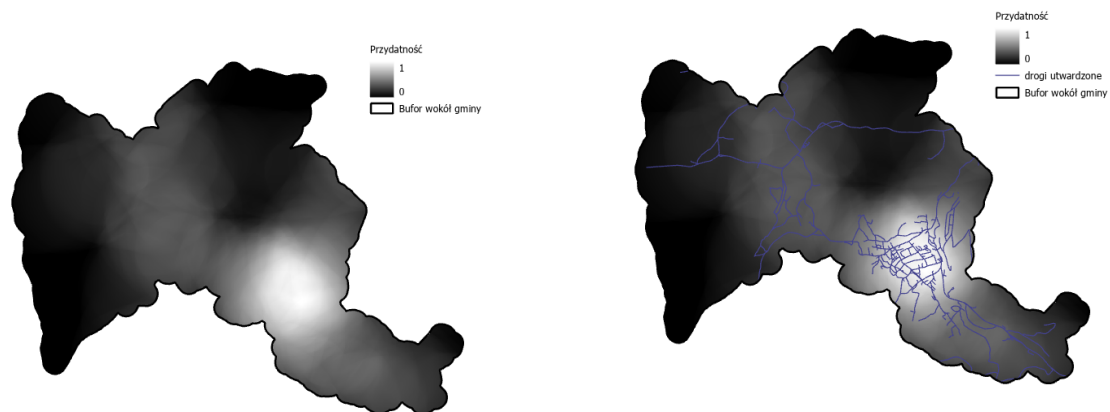
Wybranie maksymalnej wartości by użyć jej do przeskalowania do wartości 0-1

```
1 lide_den_skdr_max = line_den_skdr.maximum
```

Przeskalowanie wartości do wartości 0-1 i zapis rastra do nowego pliku.

```
1 lide_den_skdr_rescale = arcpy.sa.RescaleByFunction(  
2     in_raster=line_den_skdr,  
3     transformation_function="LINEAR 0 lide_den_skdr_max 0 #  
4         lide_den_skdr_max #",  
5     from_scale=0,  
6     to_scale=1  
7 )  
8 lide_den_skdr_rescale.save("kryterium4.tif")
```

Otrzymano następujące wyniki:



Mapa przydatności dla kryterium 4.

Drogi utwardzone na tle mapy przydatności dla kryterium 4.

Rysunek 4: Wyniki dla kryterium 4.

Widać, że mapa przydatności została wygenerowana poprawnie. W południowo-wschodniej części gminy, gdzie występuje dużo dróg utwardzonych obszar został uznany za najbardziej przydatny. Tereny z małym zagęszczeniem dróg utwardzonych są uznawane za mniej przydatne. Tam gdzie drogi utwardzone nie występują, obszar jest uznawany za nieprzydatny. Uzyskuje się niewiele obszarów o największej przydatności.

## 2.6 Kryterium 5.

Kryterium polega na wskazaniu terenów o nachyleniu nieprzekraczającym 10%. Za optymalny spadek terenu uznano spadek równy 0-3%. Aby stworzyć mapę spadków wykorzystano narzędzie *Slope*, gdzie wejściową warstwą rastrową była warstwa powstała w wyniku uprzedniego połączenia sekcji NMT obejmujących obszar analizowanej gminy. Wartości spadków zostały obliczone w procentach. Następnie korzystając z narzędzia *FuzzyMembership* stworzono funkcję przydatności, która obszarom o nachyleniu 0-3% przypisała największą przydatność (równą 1), a obszarom o nachyleniu powyżej 10% przydatność równą 0. W przedziale 3-10% wartości funkcji przydatności liniową maleją. Wynik został zapisany do nowego rastra.

### Kod realizujący kryterium 5:

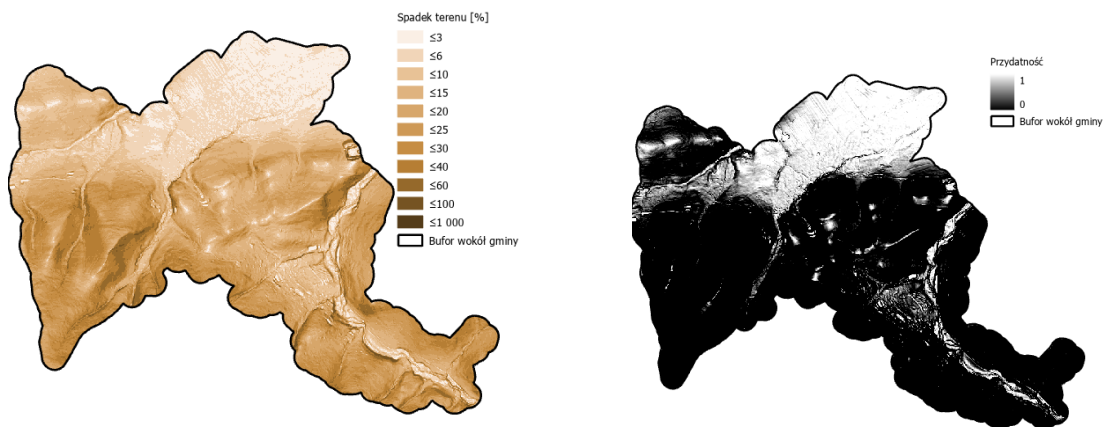
Stworzenie mapy spadków

```
1 slope = arcpy.ddd.Slope(  
2     in_raster=nmt,  
3     out_raster='slope',  
4     output_measurement="PERCENT_RISE",  
5     z_factor=1  
6 )
```

Stworzenie funkcji przydatności i zapis wyniku do nowego pliku

```
1 slope_fmalejaca = arcpy.sa.FuzzyMembership(  
2     in_raster=slope,  
3     fuzzy_function="LINEAR 10 3"  
4 )  
5 slope_fmalejaca.save("kryterium5.tif")
```

Otrzymano następujące wyniki:



Mapa spadków stworzona przy realizacji kryterium 5.

Mapa przydatności dla kryterium 5.

Rysunek 5: Wyniki dla kryterium 5.

Widać, że mapa przydatności zostaje utworzona w sposób poprawny. Obszary najbardziej płaskie są uznawane za najbardziej przydatne, gdy spadek jest większy niż 10% to obszar uznaje się za nieprzydatny. Północna część gminy oraz nieduże obszary w innych częściach gminy zostają uznane jako najbardziej przydatne. Kryterium to eliminuje sporą część terenu gminy.

## 2.7 Kryterium 6.

Kryterium polegało na wskazaniu terenów o najlepszym dostępie do światła słonecznego. W tym celu skorzystano z narzędzia *Aspect* tworząc mapę wystawy słonecznej. Za najlepsze uznano stoki południowe, a za nie najlepsze, ale wciąż przydatne stoki południowo-wschodnie i południowo-zachodnie. Stworzono dwie funkcje przydatności (narzędzie *FuzzyMembership*): rosnącą od 113° do

158° oraz malejącą od 203° do 248°. Następnie połączono je tworząc mapę przydatności (narzędzie *FuzzyOverlay*) i zapisano do nowego pliku rastrowego.

### Kod realizujący kryterium 6:

Stworzenie mapy wystawy słonecznej

```
1 aspect=arcpy.ddd.Aspect(
2     in_raster=nmt,
3     out_raster="aspect"
4 )
```

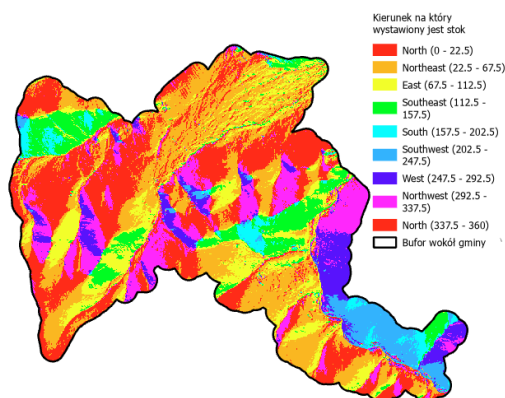
Stworzenie funkcji przydatności

```
1 aspect_froznaca = arcpy.sa.FuzzyMembership(
2     in_raster=aspect,
3     fuzzy_function="LINEAR 113 158"
4 )
5
6 aspect_fmalejaca = arcpy.sa.FuzzyMembership(
7     in_raster=aspect,
8     fuzzy_function="LINEAR 248 203"
9 )
```

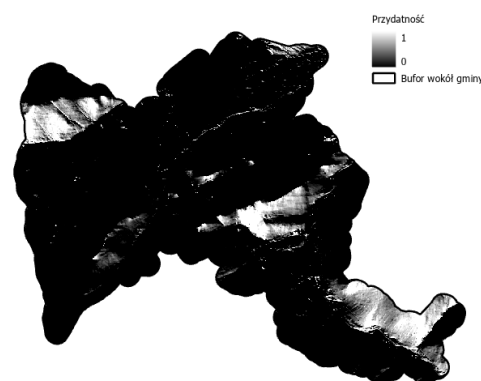
Połączenie funkcji przydatności i stworzenie mapy przydatności, zapis jej do pliku

```
1 mapa_przydatnosci_aspect = arcpy.sa.FuzzyOverlay(
2     in_rasters=[aspect_fmalejaca, aspect_froznaca],
3     overlay_type="AND")
4 mapa_przydatnosci_aspect.save("kryterium6.tif")
```

Otrzymano następujące wyniki:



Mapa wystawy słonecznej stworzona przy realizacji kryterium 6.



Mapa przydatności dla kryterium 6.

Rysunek 6: Wyniki dla kryterium 6.

Otrzymano tereny przydatne w północno-zachodniej, centralnej oraz południowo-wschodniej części gminy. Duża część terenu została uznana za obszar nieprzydatny.

## 2.8 Kryterium 7.

Kryterium polegało na wskazaniu obszarów o jak najkrótszym czasie dojazdu do istotnych drogowych węzłów komunikacyjnych. W celu realizacji tego kryterium najpierw posłużono się narzędziem *Iso - Area as Interpolation (from Layer)* dostarczonym wtyczką QNEAT3 w oprogramowaniu QGIS. Posłużyło ono do wygenerowania rastra zawierającego odległości od węzłów komunikacyjnych. Jako dane wejściowe do tej operacji wykorzystano drogi utwardzone pochodzące z warstwy SKDR z BDOT10k oraz węzły komunikacyjne z warstwy SKRW również z BDOT10k. w skrypcie

wczytano wygenerowany w QGIS raster i obliczono maksymalną wartość występującą w rastrze (narzędzie *GetRasterProperties*) zmieniając od razu pojawiający się w wartości przecinek na kropkę w celu wyeliminowania błędu z jej odczytem. Następnie stworzono funkcję przydatności (narzędzie *FuzzyMembership*). Jeśli odległość danej komórki rastra od węzła komunikacyjnego jest mniejsza niż 45% maksymalnej odległości to przydatność wynosi 1, dla odległości od 45% do 70% maksymalnej odległości przydatność maleje, osiągając w końcu wartość 0 w przypadku odległości większej niż 70% odległości maksymalnej. Wynik funkcji reprezentujący przydatność terenu został zapisana do nowego pliku rastrowego.

### Kod realizujący kryterium 7:

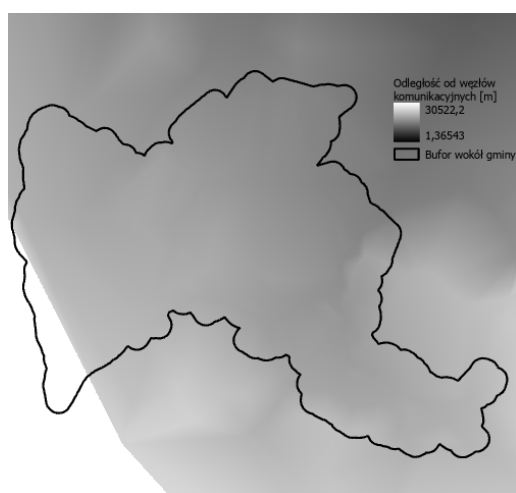
Wczytanie warstwy wygenerowanej z użyciem narzędzia w QGIS oraz obliczenie maksymalnej wartości występującej w rastrze

```
1 interpolacja = r"C:\studia\sem5\AnalizyPrzestrzenne\Proj1\warstwy\
   interpolacja.tif"
2 interpolacja_max = float(arcpy.management.GetRasterProperties(interpolacja,
   "MAXIMUM") [0].replace(',', '.'))
```

Stworzenie funkcji przydatności i zapis wyniku do nowego pliku

```
1 wezly_fmalejaca = arcpy.sa.FuzzyMembership(
2     in_raster=interpolacja_data,
3     fuzzy_function = f"LINEAR {0.7*interpolacja_max} {0.45*interpolacja_max}
4 )
5
6 wezly_fmalejaca.save("kryterium7.tif")
```

Otrzymano następujące wyniki:



Odległość od węzłów komunikacyjnych (w metrach).



Mapa przydatności dla kryterium 7.

Rysunek 7: Wyniki dla kryterium 7.

Otrzymano mało terenów o największej oraz najmniejszej możliwej przydatności. Największą przydatnością charakteryzują się tereny przy wschodniej granicy gminy, a najmniejszą przy zachodniej oraz południowo-wschodniej

## 2.9 Ocena przydatności terenu

Stworzono tabele wag dla kryteriów, w których zastosowano logikę rozmytą (*WSTable*) dla podejścia z wagami równymi oraz różnymi. W przypadku równych wag waga każdego kryteriów wynosi 1/7. Wagi dla podejścia z różnymi wagami prezentują się następująco:

- Kryterium 1 - odległość od wód i zbiorników wodnych: 0.05

- Kryterium 2 - odległość od budynków mieszkalnych: 0.1
- Kryterium 3 - pokrycie terenu: 0.05
- Kryterium 4 - dostęp do dróg utwardzonych: 0.15
- Kryterium 5 - nachylenie stoków: 0.25
- Kryterium 6 - dostęp do światła słonecznego: 0.25
- Kryterium 7 - dostęp do węzłów komunikacyjnych: 0.15

Uznano, że zależy nam na optymalizacji wydajności farmy oraz optymalizacji kosztów budowy. Lokowanie inwestycji w miejscu o niskim nasłonecznieniu skutkowałoby znacznym spadkiem generowanej energii, co skutkowałoby niższymi przychodami, a zapewnienie dobrego nasłonecznienia paneli, wymagało wybrania działek o płaskiej powierzchni - dlatego kryterium 5 i 6 otrzymały najwyższe wagi. Bliskość węzłów komunikacyjnych (kryterium 7.) oraz dróg utwardzonych (kryterium 4.) sprawia, że maleją koszty transportu paneli, urządzeń, oraz ich serwisu. Utrudniony dostęp do farmy fotowoltaicznej mógłby spowodować opóźnienia w przypadku konieczności naprawy lub wymiany paneli.

Połączono ze sobą kryteria rozmyte (w dwóch wariantach), a później połączono je z uprzednio wygenerowanymi kryteriami ostrymi. Aby stworzyć kryteria ostre należało obliczyć maksymalne wartości dla odległości od wody, lasów, budynków mieszkalnych oraz maksymalną wartość nachylenia stoków. Wykorzystano je potem do przekształcania wartości rastrów. Stworzono rastry, które mają wartość 0 poniżej progu określonego w kryteriach (dla wód jest to 100 m, dla budynków 150 m, dla lasów 15 m, dla stoków 10%), a wartości 1 w przedziale od określonego progu do maksymalnej wartości występującej w rastrze. Użyto do tego narzędzia *Reclassify*. Uwzględnienie kryteriów ostrych sprawia, że eliminuje się obszary, które nie spełniają wymagań kryteriów ostrych. Wyniki łączenia kryteriów ostrych i rozmytych zostały zapisane do nowych rastrów.

Następnie dokonano reklasyfikacji uzyskanych w poprzednim kroku rastrów, tak by otrzymać wartości w zakresie 0-1. Posłużono się narzędziem *Reclassify*. Obszarom które mają przydatność powyżej określonego progu przypisano wartość 1, obszarom, których przydatność wynosi mniej niż określony próg wartość 0. Jako próg przydatności przyjęto 50% maksymalnej przydatności. Początkowe założenia projektu mówiły o 80%, jednak otrzymywane wyniki były niezadowalające - otrzymywano obszary, które nie spełniały kryterium powierzchni/szerokości lub otrzymywano tylko jedną działkę, co uniemożliwiało analizę w oparciu o koszt przyłącza. Z zreklasyfikowanego rastra stworzono warstwę wektorową (narzędzie *RasterToPolygon*) i wybrano z niej poligony reprezentujące obszary przydatne tj. poligony o wartości atrybutu *gridcode* równej 1 (skorzystano z narzędzia *SelectLayerByAttribute*). Wyniki zapisano do nowej warstwy wektorowej.

### Kod realizujący tworzenie kryteriów ostrych:

```

1  wody_max = eucDist_wody.maximum
2  wody_ostre = arcpy.sa.Reclassify(
3      in_raster=eucDist_wody,
4      reclass_field="VALUE",
5      remap=f"0 100 0;100 {wody_max} 1",
6      missing_values="NODATA"
7  )
8  wody_ostre.save("kryterium_ostre1.tif")
9
10 budynki_max = eucDist_budynki.maximum
11 budynki_ostre = arcpy.sa.Reclassify(
12     in_raster=eucDist_budynki,
13     reclass_field="VALUE",
14     remap=f"0 150 0;150 {budynki_max} 1",
15     missing_values="NODATA"
16 )
17 budynki_ostre.save("kryterium_ostre2.tif")
18
19 lasy_max = eucDist_lasy.maximum

```

```

20 lasy_ostre = arcpy.sa.Reclassify(
21     in_raster=eucDist_lasy,
22     reclass_field="VALUE",
23     remap=f"0 15 0;15 {lasy_max} 1",
24     missing_values="NODATA"
25 )
26 lasy_ostre.save("kryterium_ostre3.tif")
27
28 slope_max = arcpy.GetRasterProperties_management(slope, "MAXIMUM").
    getOutput(0)
29 stoki_ostre = arcpy.sa.Reclassify(
30     in_raster=slope,
31     reclass_field="VALUE",
32     remap=f"0 10 1;10 {slope_max} 0",
33 )
34 stoki_ostre.save("kryterium_ostre4.tif")

```

**Kod realizujący ocenę przydatności terenu w podejściu z wagami równymi:**  
 Połączenie kryteriów rozmytych z równymi wagami

```

1 waga = 1/7
2 table_rowne = arcpy.sa.WSTable([["kryterium1.tif", "VALUE", waga],
3     ["kryterium2.tif", "VALUE", waga],
4     ["kryterium3.tif", "VALUE", waga],
5     ["kryterium4.tif", "VALUE", waga],
6     ["kryterium5.tif", "VALUE", waga],
7     ["kryterium6.tif", "VALUE", waga],
8     ["kryterium7.tif", "VALUE", waga]])
9
10 kryteria_rowne= arcpy.sa.WeightedSum(table_rowne)
11 kryteria_rowne.save("kryteria_rowne.tif")

```

Połączenie kryteriów ostrych

```

1 kryteria_ostre = arcpy.sa.FuzzyOverlay(
2     in_rasters=["kryterium_ostre1.tif","kryterium_ostre2.tif","
    kryterium_ostre3.tif","kryterium_ostre4.tif"],
3     overlay_type="AND"
4 )
5 kryteria_ostre.save("kryteria_ostre.tif")

```

Połączenie kryteriów rozmytych i ostrych

```

1 kryteria_polaczone = arcpy.sa.FuzzyOverlay(
2     in_rasters=[kryteria_ostre,kryteria_rowne],
3     overlay_type="AND"
4 )
5 kryteria_polaczone.save("kryteria_polaczone_rowne.tif")

```

Reklasyfikacja mapy przydatności

```

1 kryteria_polaczone_max = kryteria_polaczone.maximum
2 granica_przydatnosci = 0.5 * kryteria_polaczone_max
3
4 kryteria_polaczone_reclass = arcpy.sa.Reclassify(
5     in_raster="kryteria_polaczone_rowne.tif",
6     reclass_field="VALUE",
7     remap=f"0 {granica_przydatnosci} 0; {granica_przydatnosci} 1 1",
8     missing_values="NODATA"
9 )
10 kryteria_polaczone_reclass.save("kryteria_polaczone_reclass1.tif")

```

Konwersja rastra na warstwę wektorową

```

1 #zmiana rastra na shapefile
2 arcpy.conversion.RasterToPolygon(
3     in_raster="kryteria_polaczone_reclass1.tif",
4     out_polygon_features="kryteria_polaczone_shp1",
5     simplify="SIMPLIFY",

```

```

6         raster_field="Value",
7         create_multipart_features="SINGLE_OUTER_PART",
8         max_vertices_per_feature=None
9     )

```

Wybranie obszarów przydatnych i zapis do nowej warstwy wektorowej

```

1     obszary_przydatne_rowne = arcpy.management.SelectLayerByAttribute(
2         in_layer_or_view="kryteria_polaczone_shp1",
3         where_clause = "gridcode = 1"
4     )
5     arcpy.management.CopyFeatures(obszary_przydatne_rowne, '
6         obszary_przydatne_rowne')

```

textbfKod realizujący ocenę przydatności terenu w podejściu z wagami różnymi:

```

1 table_rozne = arcpy.sa.WSTable([["kryterium1.tif", "VALUE", 0.05],
2 ["kryterium2.tif", "VALUE", 0.1],
3 ["kryterium3.tif", "VALUE", 0.05],
4 ["kryterium4.tif", "VALUE", 0.15],
5 ["kryterium5.tif", "VALUE", 0.25],
6 ["kryterium6.tif", "VALUE", 0.25],
7 ["kryterium7.tif", "VALUE", 0.15]])
8
9 kryteria_rozne = arcpy.sa.WeightedSum(table_rozne)
10 kryteria_rozne.save("kryteria_rozne.tif")

```

Połączenie kryteriów rozmytych i ostrych

```

1 kryteria_polaczone = arcpy.sa.FuzzyOverlay(
2     in_rasters=[kryteria_ostre, kryteria_rozne],
3     overlay_type="AND"
4 )
5 kryteria_polaczone.save("kryteria_polaczone_rozne.tif")

```

Reklasyfikacja mapy przydatności

```

1 kryteria_polaczone_max = kryteria_polaczone.maximum
2 granica_przydatnosci = 0.5 * kryteria_polaczone_max
3
4 kryteria_polaczone_reclass = arcpy.sa.Reclassify(
5     in_raster="kryteria_polaczone_rozne.tif",
6     reclass_field="VALUE",
7     remap=f"0 {granica_przydatnosci} 0; {granica_przydatnosci} 1 1",
8     missing_values="NODATA"
9 )
10 kryteria_polaczone_reclass.save("kryteria_polaczone_reclass1.tif")

```

Konwersja rastra na warstwę wektorową

```

1 #zmiana rastra na shapefile
2 arcpy.conversion.RasterToPolygon(
3     in_raster="kryteria_polaczone_reclass1.tif",
4     out_polygon_features="kryteria_polaczone_shp1",
5     simplify="SIMPLIFY",
6     raster_field="Value",
7     create_multipart_features="SINGLE_OUTER_PART",
8     max_vertices_per_feature=None
9 )

```

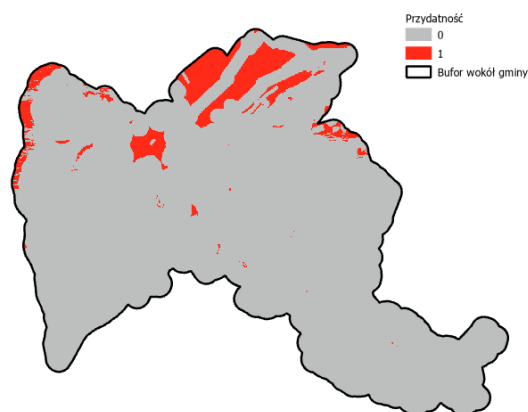
Wybranie obszarów przydatnych i zapis do nowej warstwy wektorowej

```

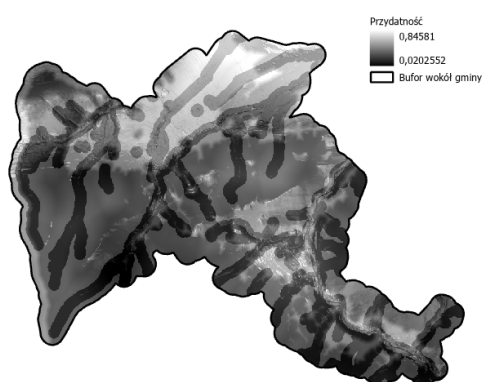
1     obszary_przydatne_rozne = arcpy.management.SelectLayerByAttribute(
2         in_layer_or_view="kryteria_polaczone_shp1",
3         where_clause = "gridcode = 1"
4     )
5     arcpy.management.CopyFeatures(obszary_przydatne_rozne, '
6         obszary_przydatne_rozne')

```

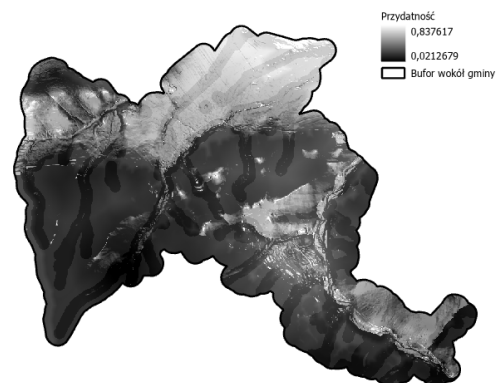
Otrzymano następujące wyniki:



Rysunek 8: Wynik łączenia kryteriów ostrych.

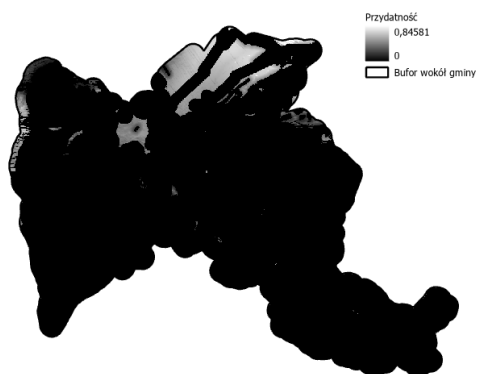


Wynik łączenia kryteriów rozmytych w podejściu z równymi wagami.

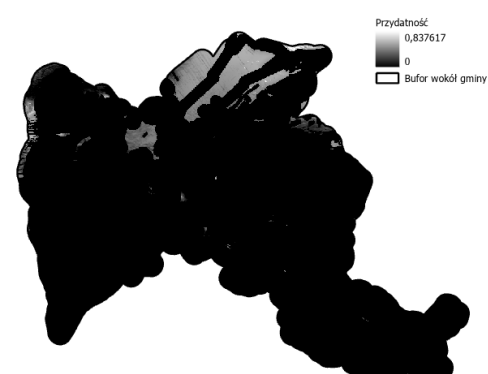


Wynik łączenia kryteriów rozmytych w podejściu z różnymi wagami.

Rysunek 9: Porównanie wyników łączenia kryteriów rozmytych w obu podejściach



Wynik łączenia kryteriów ostrych i rozmytych w podejściu z równymi wagami.

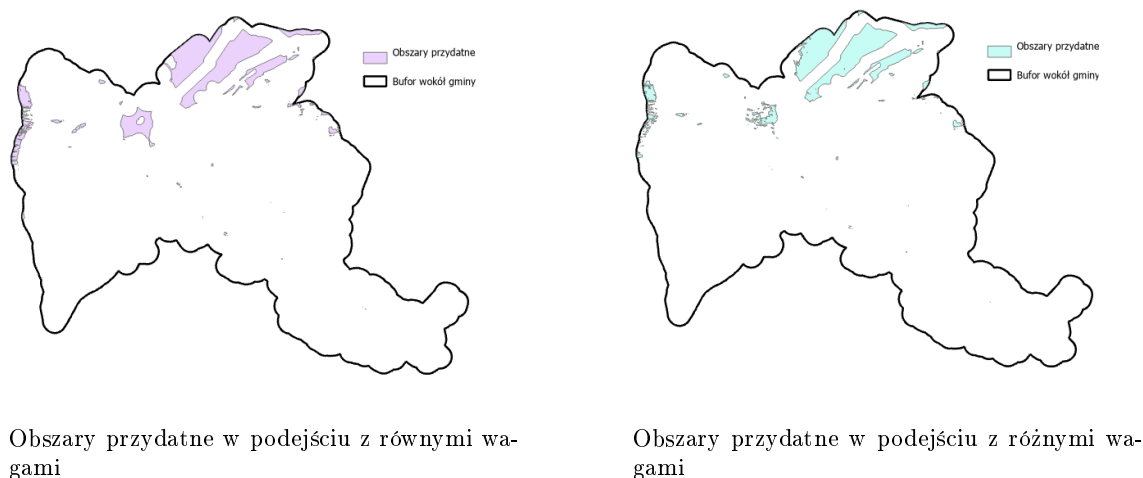


Wynik łączenia kryteriów ostrych i rozmytych w podejściu z różnymi wagami.

Rysunek 10: Porównanie wyników łączenia kryteriów ostrych i rozmytych w obu podejściach



Po połączeniu kryteriów otrzymano niewiele obszarów przydatnych w obu podejściach. Może być to spowodowane wzajemnym wykluczaniem się kryteriów np. obszary przydatne z kryterium 2. są w dużej mierze nieprzydatne w kryterium 3. Maksymalna uzyskana przydatność jest nieco większa w podejściu w wagami równymi.



Rysunek 11: Porównanie obszarów przydatnych uzyskanych w obu podejściach

Porównując obszary przydatne otrzymane w obu podejściach można zauważyć, że są one bardzo podobne. W podejściu z równymi wagami można zauważyć nieco więcej obszarów przydatnych.

## 2.10 Wybranie przydatnych działek/grup działek

Działki z dwóch powiatów uzyskane usługą WFS połączono w jedną warstwę wektorową (*Merge*). Wyselekcjonowano działki leżące na obszarach przydatnych. Użyto do tego narzędzia *SummarizeWithin*. Poprzez wybranie opcji *ADD\_SUM\_SHAPE* uzyskano od razu pole o nazwie *sum\_Area\_H*, które zawiera informację o powierzchni (w ha) obszarów "przecięcia" działek z obszarami przydatnego. Z racji wybrania opcji *ONLY\_INTERSECTING* nastąpiła selekcja tylko tych działek, które przecinają się z obszarami przydatnymi. Podczas korzystania z tego narzędzia otrzymano komunikat mówiący o tym, że ArcGIS wymaga, by wynik *SummarizeWithin* był zapisywany w geobazie lub in\_memory. Z racji tego że jako workspace ustawiono nie geobazę a folder na komputerze konieczny był eksport wyniku narzędzia do folderu ze wszystkimi warstwami (posłużono się narzędziem *FeatureClassToShapefile*). Następnie obliczono ile procent powierzchni działki stanowi obszar przydatny. Narzędziem *CalculateField* stworzono nowe pole o nazwie *pow\_przyd*, w którym wyrażenie  $100 * !sum\_Area\_H! / (!Shape\_Area! / 10000)$  oblicza stosunek powierzchni przecięcia do całkowitej powierzchni działki i przekształca wynik na wartość procentową. Użyte w wyrażeniu *Shape\_Area* to pole powierzchni działki w  $m^2$ . Zostało ono odpowiednio podzielone, by obliczenia były wykonywane na tych samych jednostkach - ha. Działki leżące na obszarach przydatnych z informacją o procentowym udziale obszaru przydatnego w powierzchni działki zapisano do nowej warstwy wektorowej. Następnie wybrano tylko te działki, których minimum 60% powierzchni stanowią obszary przydatne. W tym celu w narzędziu *SelectLayerByAttribute* wykorzystano pole *pow\_przyd*. Wyniki zapisano do nowej warstwy wektorowej. Używając narzędzia *Dissolve* połączono (wyselekcjonowane w poprzednim kroku) sąsiadujące działki w jeden obiekt. Dla połączonych działek użyto narzędzia *CalculateGeometryAttributes* i obliczono powierzchnię (w  $m^2$ ) oraz obwód (w m), a następnie wybrano te, których powierzchnia wynosi przynajmniej 2ha (czyli 20 000  $m^2$ ). Działki o odpowiedniej powierzchni zapisano do nowej warstwy wektorowej i stworzono dla nich prostokąt ograniczający. Posłużono się narzędziem *MinimumBoundingGeometry* z opcją *RECTANGLE\_BY\_WIDTH* oraz *MBG\_FIELDS* dzięki czemu otrzymano wymiary prostokątów ograniczających. Do warstwy działek o odpowiedniej powierzchni przekazano atrybut szerokości prostokąta ograniczającego - atrybut o nazwie *MBG\_Width*. Do przekazania atrybutu skorzystano z narzędzia *JoinField*. Mając dla działek informacje o szerokości dokonano selekcji działek narzędziem *SelectLayerByAttribute* - wybrano te, których szerokość jest większa lub równa 50 m. Wyniki zapisano do nowej warstwy wektorowej.

## Kod realizujący wybór przydatnych działek w podejściu z wagami równymi:

Wybór działek, których min. 60% powierzchni stanowią obszary przydatne

```
1      dzialki_przydatne_rowne_wagi_60 = arcpy.management.  
      SelectLayerByAttribute(  
2      in_layer_or_view=dzialki_przydatne_rowne_wagi,  
3      selection_type="NEW_SELECTION",  
4      where_clause="pow_przyd >= 60"  
5      )  
6      arcpy.management.CopyFeatures(dzialki_przydatne_rowne_wagi_60, '  
      dzialki_przydatne_rowne_wagi_60')
```

Połączenie sąsiadujących działki w jeden obiekt

```
1      arcpy.management.Dissolve(  
2      in_features="dzialki_przydatne_rowne_wagi_60",  
3      out_feature_class="dzialki_rowne_wagi_dissolve",  
4      dissolve_field=None,  
5      statistics_fields=None,  
6      multi_part="SINGLE_PART",  
7      unsplit_lines="DISSOLVE_LINES",  
8      concatenation_separator=""  
9      )
```

Obliczenie powierzchni (w m<sup>2</sup>) i obwodu (w m) działek po połączeniu

```
1      arcpy.management.CalculateGeometryAttributes(  
2      in_features="dzialki_rowne_wagi_dissolve",  
3      geometry_property="area AREA;obw PERIMETER_LENGTH",  
4      length_unit="METERS",  
5      area_unit="SQUARE_METERS",  
6      coordinate_system='PROJCS ["ETRF2000-PL_CS92",GEOGCS ["ETRF2000-PL",DATUM  
      ["ETRF2000_Poland",SPHEROID ["GRS_1980",6378137.0,298.257222101]],  
      PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],  
      PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting  
      ",500000.0],PARAMETER["False_Northing",-5300000.0],PARAMETER["  
      Central_Meridian",19.0],PARAMETER["Scale_Factor",0.9993],PARAMETER  
      ["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]',  
7      coordinate_format="SAME_AS_INPUT"  
8      )
```

Wybór działek o powierzchni minimum 2 ha

```
1      dzialki_rowne_wagi_2ha = arcpy.management.SelectLayerByAttribute(  
2      in_layer_or_view="dzialki_rowne_wagi_dissolve",  
3      selection_type="NEW_SELECTION",  
4      where_clause="area >= 20000"  
5      )  
6      arcpy.management.CopyFeatures(dzialki_rowne_wagi_2ha, '  
      dzialki_rowne_wagi_2ha')
```

Stworzenie otoczki wypukłej dla działek

```
1      arcpy.management.MinimumBoundingGeometry(  
2      in_features="dzialki_rowne_wagi_2ha",  
3      out_feature_class="dzialki_rowne_wagi_otoczka",  
4      geometry_type="RECTANGLE_BY_WIDTH",  
5      group_option="NONE",  
6      group_field=None,  
7      mbg_fields_option="MBG_FIELDS"  
8      )
```

Przekazanie do działek atrybutu szerokości

```
1      arcpy.management.JoinField(  
2      in_data="dzialki_rowne_wagi_2ha",  
3      in_field="FID",  
4      join_table="dzialki_rowne_wagi_otoczka",
```

```

5     join_field="FID",
6     fields="MBG_Width",
7     fm_option="NOT_USE_FM",
8     field_mapping=None,
9     index_join_fields="NO_INDEXES"
10 )

```

Wybór grup działek o szerokości minimum 50 metrów

```

1 dzialki_rowne_wagi_szerokosc50 = arcpy.management.SelectLayerByAttribute(
2     in_layer_or_view="dzialki_rowne_wagi_2ha",
3     selection_type="NEW_SELECTION",
4     where_clause="MBG_Width >= 50"
5 )
6 arcpy.management.CopyFeatures(dzialki_rowne_wagi_szerokosc50, '
    dzialki_rowne_wagi_szerokosc50')

```

**Kod realizujący wybór przydatnych działek w podejściu z wagami różnymi:**

Wybór działek, których min. 60% powierzchni stanowią obszary przydatne

```

1     dzialki_przydatne_rozne_wagi_60 = arcpy.management.
2         SelectLayerByAttribute(
3     in_layer_or_view=dzialki_przydatne_rozne_wagi,
4     selection_type="NEW_SELECTION",
5     where_clause="pow_przyd >= 60"
6 )
7     arcpy.management.CopyFeatures(dzialki_przydatne_rozne_wagi_60, '
    dzialki_przydatne_rozne_wagi_60')

```

Połączenie sąsiadujących działki w jeden obiekt

```

1     arcpy.management.Dissolve(
2     in_features="dzialki_przydatne_rozne_wagi_60",
3     out_feature_class="dzialki_rozne_wagi_dissolve",
4     dissolve_field=None,
5     statistics_fields=None,
6     multi_part="SINGLE_PART",
7     unsplit_lines="DISSOLVE_LINES",
8     concatenation_separator=""
9 )

```

Obliczenie powierzchni (w m<sup>2</sup>) i obwodu (w m) działek po połączeniu

```

1     arcpy.management.CalculateGeometryAttributes(
2     in_features="dzialki_rozne_wagi_dissolve",
3     geometry_property="area AREA;obw PERIMETER_LENGTH",
4     length_unit="METERS",
5     area_unit="SQUARE_METERS",
6     coordinate_system='PROJCS ["ETRF2000-PL_CS92",GEOGCS ["ETRF2000-PL",DATUM
7         ["ETRF2000_Poland",SPHEROID ["GRS_1980",6378137.0,298.257222101]],
8         PRIMEM ["Greenwich",0.0],UNIT ["Degree",0.0174532925199433]],
9         PROJECTION ["Transverse_Mercator"],PARAMETER ["False_Easting
10             ",500000.0],PARAMETER ["False_Northing",-5300000.0],PARAMETER ["
11             Central_Meridian",19.0],PARAMETER ["Scale_Factor",0.9993],PARAMETER
12             ["Latitude_Of_Origin",0.0],UNIT ["Meter",1.0]]',
13     coordinate_format="SAME_AS_INPUT"
14 )

```

Wybór działek o powierzchni minimum 2 ha

```

1 dzialki_rozne_wagi_2ha = arcpy.management.SelectLayerByAttribute(
2     in_layer_or_view="dzialki_rozne_wagi_dissolve",
3     selection_type="NEW_SELECTION",
4     where_clause="area >= 20000"
5 )
6 arcpy.management.CopyFeatures(dzialki_rozne_wagi_2ha, '
    dzialki_rozne_wagi_2ha')

```

Stworzenie otoczki wypukłej dla działek

```
1 arcpy.management.MinimumBoundingGeometry(  
2     in_features="dzialki_rozne_wagi_2ha",  
3     out_feature_class="dzialki_rozne_wagi_otoczka",  
4     geometry_type="RECTANGLE_BY_WIDTH",  
5     group_option="NONE",  
6     group_field=None,  
7     mbg_fields_option="MBG_FIELDS"  
8 )
```

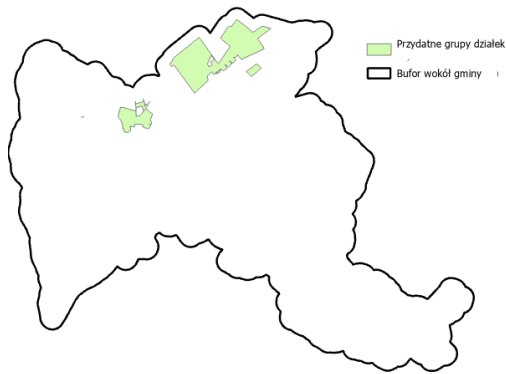
Przekazanie do działek atrybutu szerokości

```
1 arcpy.management.JoinField(  
2     in_data="dzialki_rozne_wagi_2ha",  
3     in_field="FID",  
4     join_table="dzialki_rozne_wagi_otoczka",  
5     join_field="FID",  
6     fields="MBG_Width",  
7     fm_option="NOT_USE_FM",  
8     field_mapping=None,  
9     index_join_fields="NO_INDEXES"  
10 )
```

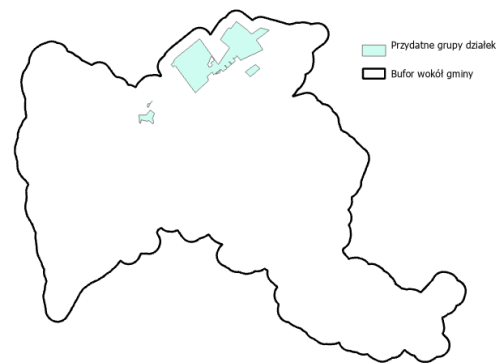
Wybór grup działek o szerokości minimum 50 metrów

```
1 dzialki_rozne_wagi_szerokosc50 = arcpy.management.SelectLayerByAttribute(  
2     in_layer_or_view="dzialki_rozne_wagi_2ha",  
3     selection_type="NEW_SELECTION",  
4     where_clause="MBG_Width >= 50"  
5 )  
6 arcpy.management.CopyFeatures(dzialki_rozne_wagi_szerokosc50, '  
    dzialki_rozne_wagi_szerokosc50')
```

Otrzymano następujące wyniki:



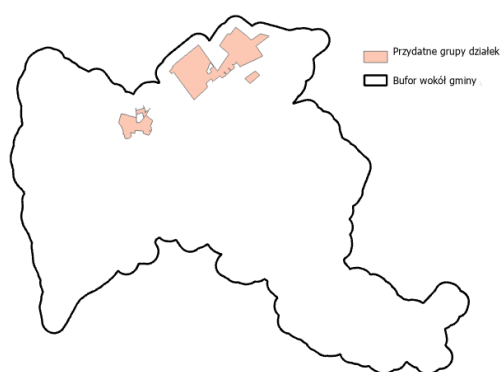
Poligony powstałe w wyniku łączenia działek o określonej przydatności w podejściu z równymi wagami



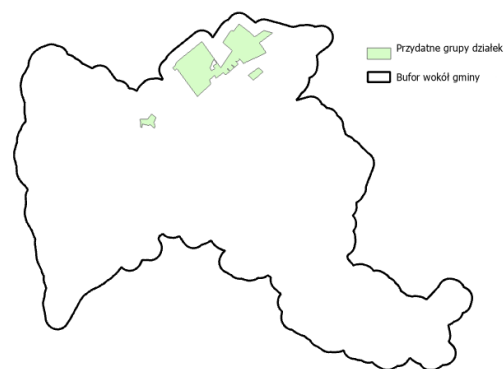
Poligony powstałe w wyniku łączenia działek o określonej przydatności w podejściu z różnymi wagami

Rysunek 12: Porównanie poligonów powstałych w wyniku łączenia działek o określonej przydatności w obu podejściach

Działki uzyskane w podejściu z równymi wagami zajmują większy obszar niż działki uzyskane w podejściu z wagami różnymi. W obu przypadkach znajdują się one w północnej części gminy.



Działki spełniające kryterium powierzchni i szerokości w podejściu z równymi wagami



Działki spełniające kryterium powierzchni i szerokości w podejściu z różnymi wagami

Rysunek 13: Porównanie działek spełniających kryterium powierzchni i szerokości w obu podejściach

Po uwzględnieniu kryterium powierzchni i szerokości w obu podejściach uzyskano 3 grupy działek. W podejściu z równymi wagami grupy jedna z nich jest większa.

## 2.11 Koszt przyłącza do sieci

W tej części posłużono się dostarczoną przez prowadzącą wektorową warstwą pokrycia terenu. Stworzono mapę kosztów względnych w postaci rastrowej, przypisując określonym kategoriom użytkowania terenu koszt względny wg. poniżej tabeli.

Kod klasy obiektów BDOT	Nazwa klasy obiektów BDOT	X_kod	Typ obiektu	Koszt względny
PTWP	woda powierzchniowa	PTWP01	woda morska	0 → NoData
		PTWP02	woda płynąca	200
		PTWP03	woda stojąca	0 → NoData
PTZB	zabudowa	PTZB02	jednorodzinna	100
		PTZB01	wielorodzinna	200
		PTZB05	pozostała zabudowa	50
		PTZB04	handlowo-usługowa	200
		PTZB03	przemysłowo-składowa	200
PTLZ	teren leśny i zadrzewiony	PTLZ01	las	100
		PTLZ02	zagałnik	50
		PTLZ03	zadrzewienie	50
PTRK	roślinność krzewiasta	PTRK01	kępy krzewów	20
		PTRK02	krzewy	15
PTUT	uprawa trwała	PTUT03	sad	100
		PTUT02	plantacja	90
		PTUT04, PTUT05	inne	20
		PTUT01	ogród działkowy	0 → NoData
PTTR	roślinność trawiasta i uprawa rolna	PTTR02	grunt orny	1
		PTTR01	roślinność trawiasta	20
PTKM	teren pod drogami kołowymi, szynowymi i lotniskowymi	PTKM02	torowisko	200
		PTKM01	droga kołowa	200
		PTKM03	teren pod drogą kołową i torowiskiem	200
		PTKM04	teren pod drogą lotniskową	0 → NoData
PTGN	grunt nieużytkowany	PTGN01, PTGN02, PTGN03, PTGN04	-	1
PTPL	plac	PTPL01	-	50
PTSO	składowisko odpadów	PTSO01, PTSO02	-	0 → NoData
PTWZ	wyrobisko i zwałowisko	PTWZ01, PTWZ02	-	0 → NoData
PTNZ	pozostały teren niezabudowany	PTNZ01, PTNZ02	-	150

Tabela 1: Mapa kosztów na podstawie warstw PT (pokrycie terenu) z BDOT

W celu przypisania poszczególnym kategoriom terenu kosztów względnych stworzono funkcję klasyfikującą. Następnie stworzono dla warstwy wektorowej nowe pole *koszt* i przypisano mu odpowiednią wartość na podstawie atrybutu *X\_KOD*. Następnie dokonano konwersji wektora na raster (narzędzie *FeatureToRaster*) oraz przypisano wartości braku danych dla barier absolutnych (narzędzie *SetNull*). Stworzono mapę kosztów skumulowanych oraz mapę kierunków korzystając z narzędzia *CostDistance*. Wykorzystano je do tworzenia ścieżki przyłącza do linii energetycznych. Złączono

warstwy linii energetycznych z dwóch powiatów w jedną. Używając narzędzia *CostPath* stworzono ścieżkę przyłącza w postaci rastrowej i przy pomocy narzędzia *RasterToPolyline* stworzono ścieżkę przyłącza w postaci wektorowej. Ponieważ linie energetyczne przechodzą przez działki nie doszło do utworzenia ścieżki przyłącza.

#### Kod realizujący tworzenie mapy kosztów względnych:

Stworzenie funkcji klasyfikującej i przypisanie poszczególnym typom pokrycia terenu kosztów jednostkowych

```
1 codeblock = ""
2 def liczkoszt(kod):
3     if kod == "PTWP01":
4         return 0
5     elif kod == "PTWP02":
6         return 200
7     elif kod == "PTWP03":
8         return 0
9     elif kod == "PTZB02":
10        return 100
11    elif kod == "PTZB01":
12        return 200
13    elif kod == "PTZB03":
14        return 200
15    elif kod == "PTZB04":
16        return 200
17    elif kod == "PTZB05":
18        return 50
19    elif kod == "PTLZ01":
20        return 100
21    elif kod == "PTLZ02":
22        return 50
23    elif kod == "PTLZ03":
24        return 50
25    elif kod == "PTRK01":
26        return 15
27    elif kod == "PTRK02":
28        return 15
29    elif kod == "PTUT03":
30        return 100
31    elif kod == "PTUT02":
32        return 90
33    elif kod == "PTUT04":
34        return 20
35    elif kod == "PTUT05":
36        return 20
37    elif kod == "PTUT01":
38        return 0
39    elif kod == "PTTR02":
40        return 1
41    elif kod == "PTTR01":
42        return 20
43    elif kod == "PTKM01":
44        return 100
45    elif kod == "PTKM02":
46        return 200
47    elif kod == "PTKM03":
48        return 170
49    elif kod == "PTKM04":
50        return 200
51    elif kod == "PTGN01":
52        return 1
53    elif kod == "PTGN02":
54        return 1
55    elif kod == "PTGN03":
56        return 1
57    elif kod == "PTGN04":
```

```

58         return 1
59     elif kod == "PTPL01":
60         return 50
61     elif kod == "PTS001":
62         return 0
63     elif kod == "PTS002":
64         return 0
65     elif kod == "PTWZ01":
66         return 0
67     elif kod == "PTWZ02":
68         return 0
69     elif kod == "PTNZ01":
70         return 150
71     elif kod == "PTNZ02":
72         return 150
73     else:
74         return 0
75 """

```

Wczytanie warszty pokrycia terenu dostarczonej przez prowadzącą i przypisanie kosztów względnych

```

1 PT = r"C:\studia\sem5\AnalizyPrzestrzenne\Proj1\
    mapa_kosztow_BDOT10k_dane_archiwalne\PT_merge_cliped.shp"
2 arcpy.management.AddField(PT, "koszt", "DOUBLE", None, None, None, "'", "
    NULLABLE", "REQUIRED")
3 PT_koszty = arcpy.management.CalculateField(
4     in_table=PT,
5     field="koszt",
6     expression="liczkoszt(!X_KOD!)",
7     expression_type="PYTHON3",
8     code_block=codeblock,
9 )

```

Konwersja warstwy wektorowej na rastrową

```

1 koszty_raster = arcpy.conversion.FeatureToRaster(
2     in_features=PT_koszty,
3     field="koszt",
4     out_raster="koszty_raster",
5     cell_size=5
6 )

```

Przypisanie wartości braku danych dla barier absolutnych

```

1 mapa_kosztow_wzglednych = arcpy.sa.SetNull(
2     in_conditional_raster=koszty_raster,
3     in_false_raster_or_constant=koszty_raster,
4     where_clause="VALUE = 0"
5 )
6 mapa_kosztow_wzglednych.save("mapa_kosztow_wzglednych.tif")

```

Kod realizujący tworzenie ścieżki przyłącza do linii energetycznych w podejściu z wagami równymi:

Stworzenie mapy kosztów skumulowanych oraz mapy kierunków

```

1 cost_map_rowne_wagi = arcpy.sa.CostDistance(
2     in_source_data="dzialki_rowne_wagi_szerokosc50",
3     in_cost_raster="mapa_kosztow_wzglednych.tif",
4     maximum_distance=None,
5     out_backlink_raster="backlink_rowne_wagi.tif",
6     source_cost_multiplier=None,
7     source_start_cost=None,
8     source_resistance_rate=None,
9     source_capacity=None,
10    source_direction=""
11 )

```



```
12 cost_map_rowne_wagi.save("cost_map_rowne_wagi.tif")
```

Złączenie odpowiednich warstw z dwóch powiatów w jedną

```
1 arcpy.management.Merge(  
2 inputs=[suln_lubanski,suln_lwowecki],  
3 output="suln_merge"  
4 )
```

Stworzenie ścieżki przyłącza w postaci rastra

```
1 cost_path_rowne_wagi = arcpy.sa.CostPath(  
2     in_destination_data="suln_merge",  
3     in_cost_distance_raster="cost_map_rowne_wagi.tif",  
4     in_cost_backlink_raster="backlink_rowne_wagi.tif",  
5     path_type="BEST_SINGLE",  
6     force_flow_direction_convention="INPUT_RANGE"  
7 )  
8 cost_path_rowne_wagi.save("cost_path_rowne_wagi.tif")
```

Stworzenie ścieżki przyłącza w postaci wektorowej

```
1 arcpy.conversion.RasterToPolyline(  
2     in_raster=cost_path_rowne_wagi,  
3     out_polyline_features="cost_path_rowne_wagi_vector",  
4 )
```

**Kod realizujący tworzenie ścieżki przyłącza do linii energetycznych w podejściu z wagami różnymi:**

Stworzenie mapy kosztów skumulowanych oraz mapy kierunków

```
1 cost_map_rozne_wagi = arcpy.sa.CostDistance(  
2     in_source_data="dzialki_rozne_wagi_szerokosc50",  
3     in_cost_raster="mapa_kosztow_wzgleonych.tif",  
4     maximum_distance=None,  
5     out_backlink_raster="backlink_rozne_wagi.tif",  
6     source_cost_multiplier=None,  
7     source_start_cost=None,  
8     source_resistance_rate=None,  
9     source_capacity=None,  
10    source_direction=""  
11 )  
12 cost_map_rozne_wagi.save("cost_map_rozne_wagi.tif")
```

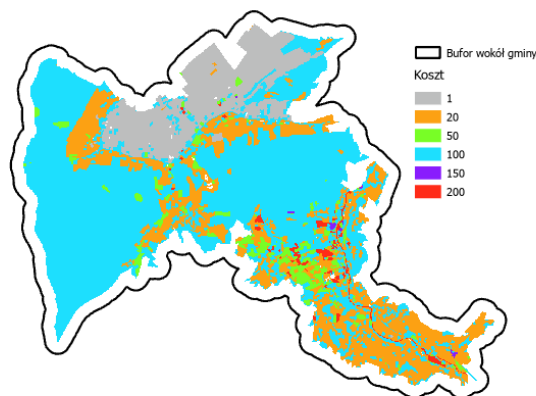
Stworzenie ścieżki przyłącza w postaci rastra

```
1 cost_path_rozne_wagi = arcpy.sa.CostPath(  
2     in_destination_data="suln_merge",  
3     in_cost_distance_raster="cost_map_rozne_wagi.tif",  
4     in_cost_backlink_raster="backlink_rozne_wagi.tif",  
5     path_type="BEST_SINGLE",  
6     force_flow_direction_convention="INPUT_RANGE"  
7 )  
8 cost_path_rozne_wagi.save("cost_path_rozne_wagi.tif")
```

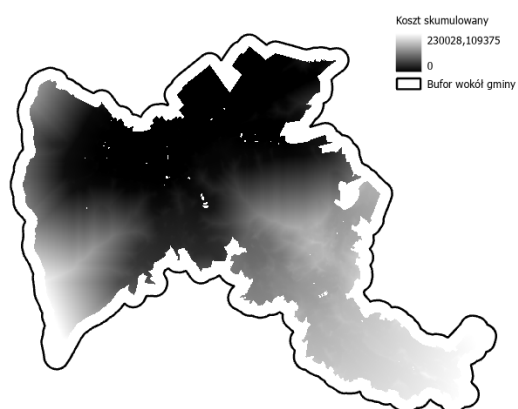
Stworzenie ścieżki przyłącza w postaci wektorowej

```
1 arcpy.conversion.RasterToPolyline(  
2     in_raster=cost_path_rozne_wagi,  
3     out_polyline_features="cost_path_rozne_wagi_vector",  
4 )
```

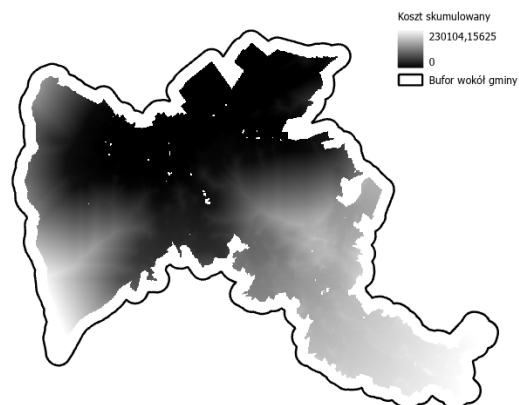
**Otrzymano następujące wyniki:**



Rysunek 14: Mapa kosztów względnych



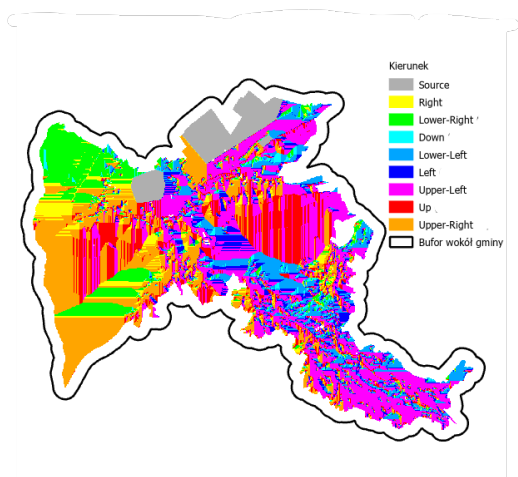
Mapa kosztów skumulowanych w podejściu z wagami równymi



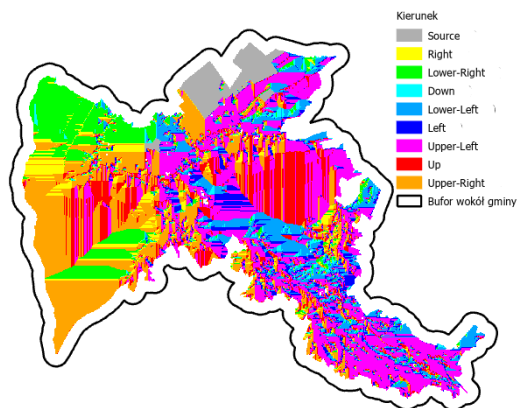
Mapa kosztów skumulowanych w podejściu z wagami różnymi

Rysunek 15: Porównanie map kosztów skumulowanych w obu podejściach

W podejściu z różnymi wagami maksymalny koszt skumulowany jest większy niż w podejściu z wagami równymi.

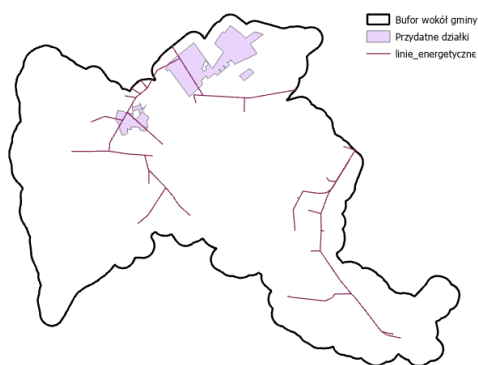


Mapa kierunków w podejściu z wagami równymi

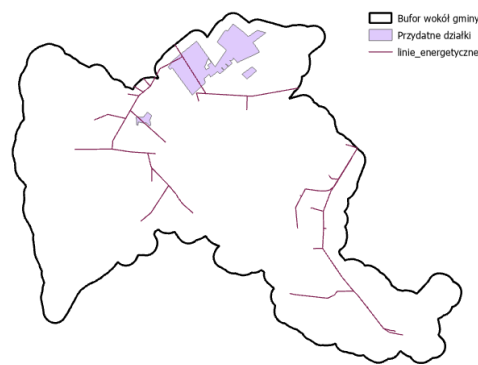


Mapa kierunków w podejściu z wagami różnymi

Rysunek 16: Porównanie map kierunków w obu podejściach



Mapa przedstawiająca przydatne działki i linie energetyczne w podejściu z wagami równymi



Mapa przedstawiająca przydatne działki i linie energetyczne w podejściu z wagami różnymi

Rysunek 17: Porównanie przydatnych działek i linii energetycznych w obu podejściach

Ponieważ w obu wariantach linie energetyczne przechodzą przez niektóre działki, nie doszło do utworzenia widocznej ścieżki przyłącza. Jest ona reprezentowana jako pojedynczy piksel, w którym teren działki i sieć energetyczna się przecinają. Za najlepsze pod względem inwestycji działki można uznać te, których teren przecina się z liniami energetycznymi.



Mapa przedstawiająca działki, przez które przechodzą linie energetyczne w podejściu z wagami równymi



Mapa przedstawiająca działki, przez które przechodzą linie energetyczne w podejściu z wagami różnymi

Rysunek 18: Porównanie przydatnych działek i linii energetycznych w obu podejściach

### 3 Podsumowanie i wnioski

W ramach projektu udało się zrealizować postawione zadanie, ale konieczna była modyfikacja założeń. Przy ustalaniu kryteriów ważne jest odpowiednie dobranie wartości liczbowych. Postawione kryteria były zbyt restrykcyjne, w wyniku czego otrzymywano zbyt mało obszarów przydatnych i trzeba było obniżyć próg przydatności.

Tworzenie skryptów znacząco ułatwia i przyspiesza pracę, szczególnie gdy trzeba powtarzać czynności na wielu warstwach. Tworzenie tego skryptu było przyjemniejsze i szybsze niż manualne używanie narzędzi w programie ArcGIS.

Nie udało się uzyskać terenu, który spełniałby wszystkie postawione kryteria - obszaru, który po połączeniu kryteriów rozmytych i ostrych miałby przydatność 1. Pokazuje to, że trudno znaleźć teren idealny pod inwestycje.

Wyniki dla podejścia z wagami równymi i różnymi były podobne, ale nie identyczne, co pokazuje, że dobór wag ma wpływ na wyniki. W podejściu z wagami równymi uzyskano większy obszar przy-

datny.

Podczas korzystania ze stworzonego modelu należy upewnić się czy wykorzystywane warstwy mają odpowiednie atrybuty. Jak pokazał test dla danych z innej gminy (przedstawiony w ostatniej sekcji sprawozdania) obecnie udostępniane warstwy pokrycia terenu nie mają atrybutu *X\_KOD*, co generuje błędy i sprawia konieczność korzystania z danych archiwalnych.

Możnaby rozszerzyć skrypt o dodatkowe działanie w przypadku, gdy linie energetyczne przechodzą przez działki. Wtedy można by było analizować, które z nich mają największą powierzchnię lub dla której z nich obszar przydatny stanowi większą część jej powierzchni. Aby uczynić korzystanie ze stworzonego modelu przyjemniejszym możnaby stworzyć Toolbox umożliwiający wykonanie stworzonego modelu, gdzie użytkownik wprowadzałby odpowiednie parametry z poziomu ArcGISa.

Model można rozbudować o analizę odległości inwestycji od obszarów chronionych.

W projekcie wykorzystywano dane dla gminy oraz 200 metrowego bufora wokół niej. Analizowana gmina leży na granicy powiatu, więc trzeba było uzyskać dane dla dwóch powiatów. Samo pobranie danych, a potem ich selekcja (by mieć te dla interesującej nas gminy) zajęło dużo czasu, ponieważ najmniejsza dostępna paczka danych BDOT10k to paczka dla całego powiatu. W niektórych przypadkach może okazać się, że potrzeba danych dla jeszcze większej liczby powiatów. Może to stwarzać pewne ograniczenia czasowe oraz pamięciowe.

ArcGIS nie radzi sobie ze ścieżkami dostępu do plików, które zawierają kropki. Jest to znaczne ograniczenie, ponieważ wymaga zmiany nazw plików.

Problematyczne okazało się uruchomienia Python Notebooka dostępnego w ArcGISie. Podjęto próbę reinstalacji programu oraz kontaktu ze wsparciem technicznych firmy ESRI, ale nie przyniosła ona rezultatów. Zdecydowano się na stworzenie skryptu w Pythonie w zewnętrznym IDE.

## 4 Przykład nowego, nieuwzględnionego kryterium

Nowym kryterium może być odległość od form ochrony przyrody: parków narodowych, parków krajobrazowych, rezerwatów przyrody, obszarów Natura 2000. Warstwy wektorowe takich obszarów można łatwo pobrać bezpośrednio poprzez odnośniki zamieszczone na rządowej stronie dostępnej pod linkiem: <https://www.gov.pl/web/gdos/dostep-do-danych-geoprzestrzennych> lub pozyskać je z Geoportalu z bazy BDOT10k (warstwy TCON - obszar Natura 2000, TCRZ - park krajobrazowy, TCPN - park narodowy, TCRZ - rezerwat). Po pobraniu tych danych należało by połączyć ze sobą pobrane warstwy (narzędziem *Merge*) i wyselekcjonować obszary leżące na terenie, którego analizę przeprowadzamy z wykorzystaniem funkcji *SelectLayerByLocation*. Następnie można by stworzyć mapę odległości od obszarów ochrony przyrody oraz mapę przydatności, bazującą na funkcji przydatności.

Proponuję następujące wartości funkcji przydatności:

- 1 - dla obszarów oddalonych od obszarów ochrony przyrody o co najmniej 500m
- 0 - dla obszarów oddalonych od obszarów ochrony przyrody o mniej niż 200m
- między 0 a 1 - dla obszarów oddalonych od obszarów ochrony przyrody o 200-500m (funkcja rosnąca)

Później trzeba by połączyć powstałą mapę przydatności z innymi kryteriami.

## 5 Testu wykonania modelu na danych z innego obszaru

### 5.1 Opis obszaru

Obszar testowy to gmina Pleśna w powiecie tarnowskim w województwie małopolskim. W tym przypadku wystraczyły dane z jednego powiatu. Część z nich została pobrana z serwisu Geoportal, część pochodziła z materiałów dostarczonych przez prowadzącą.

## 5.2 Otrzymane wyniki

### 5.2.1 Kryterium 1.



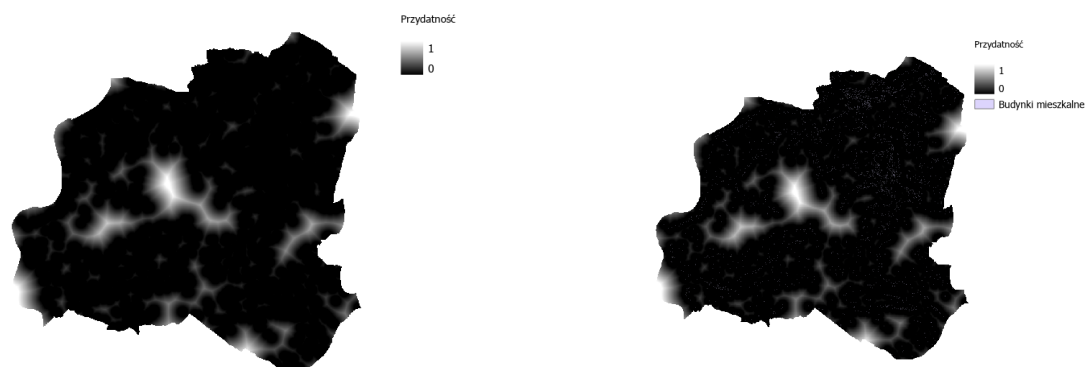
Mapa przydatności dla kryterium 1.

Wody na tle mapy przydatności dla kryterium 1.

Rysunek 19: Wyniki dla kryterium 1.

Duża część obszaru gminy Pleśna jest pokryta wodami, dlatego duża jej część zostaje uznana za nieprzydatną, ponieważ kryterium zakłada, że optymalną odległością od wód jest 102-200 m.

### 5.2.2 Kryterium 2.



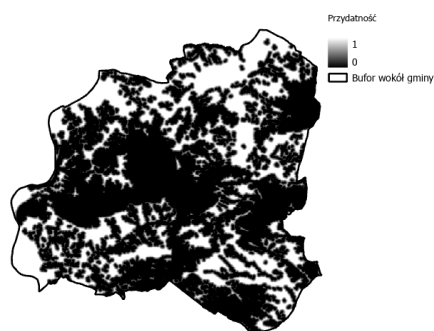
Mapa przydatności dla kryterium 2.

Budynki mieszkalne na tle mapy przydatności dla kryterium 2.

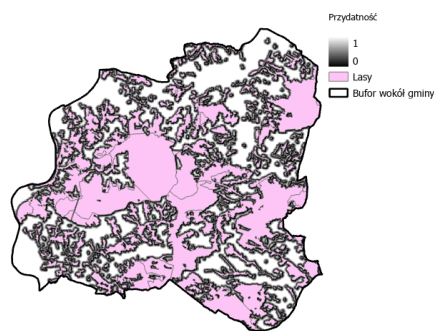
Rysunek 20: Wyniki dla kryterium 2.

W tym kryterium uzyskano mało terenów przydatnych. Wynika to z tego, że budynki mieszkalne są „rozsiiane” po całej gminie i trudno znaleźć obszary położone od nich o przynajmniej 150m.

### 5.2.3 Kryterium 3.



Mapa przydatności dla kryterium 3.

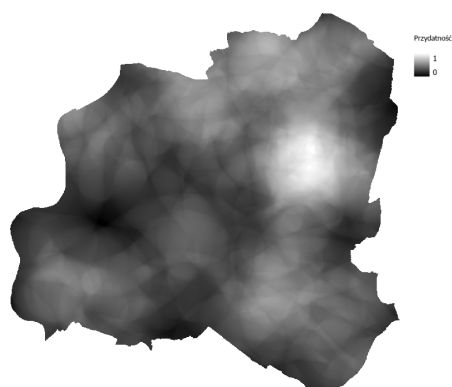


Lasy na tle mapy przydatności dla kryterium 3.

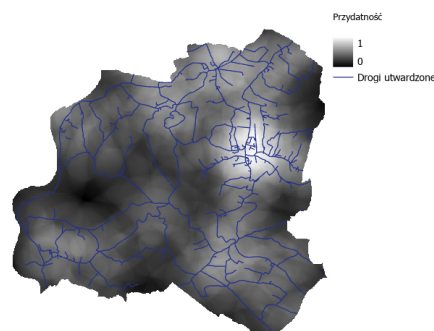
Rysunek 21: Wyniki dla kryterium 3.

Na terenie gminy Pleśna występuje dużo terenów leśnych, dlatego duża część gminy została uznana za nieprzydatną. Podobnie jak w przypadku gminy Świeradów-Zdrój obszary uznane za przydatne w kryterium 3 oraz kryterium 2 się wykluczają.

### 5.2.4 Kryterium 4.



Mapa przydatności dla kryterium 4.

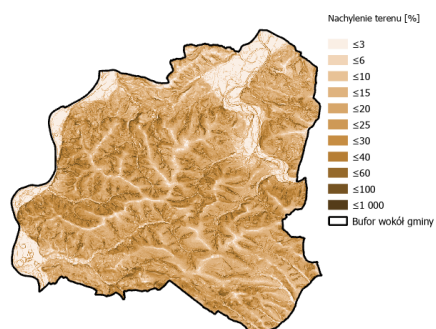


Drogi utwardzone na tle mapy przydatności dla kryterium 4.

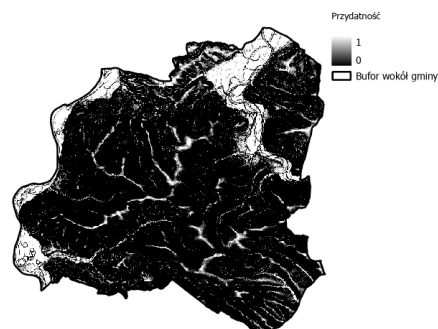
Rysunek 22: Wyniki dla kryterium 4.

Największą przydatnością cechuje się niewielki obszar położony we wschodniej części gminy. Drogi utwardzone występują mniej więcej na całym terytorium gminy, więc obszarów o zerowej przydatności jest niewiele.

### 5.2.5 Kryterium 5.



Mapa spadków stworzona przy realizacji kryterium 5.

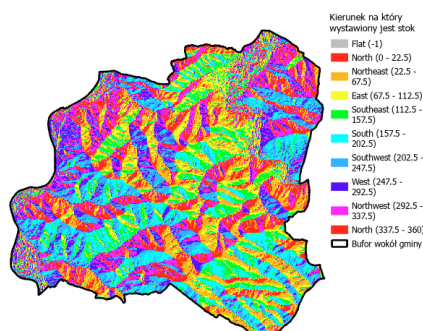


Mapa przydatności dla kryterium 5.

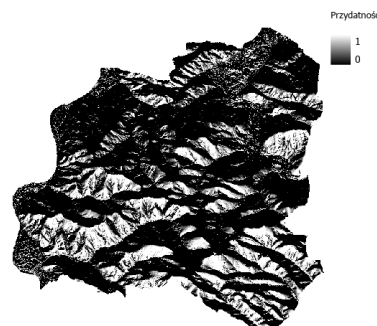
Rysunek 23: Wyniki dla kryterium 5.

Na terenie gminy występuje dużo terenów o nachyleniu powyżej 10%, dlatego duża część gminy zostaje uznana za nieprzydatną w tym kryterium. Obszary przydatne uzyskano przy zachodniej granicy gminy oraz w jej północno-wschodniej części.

### 5.2.6 Kryterium 6.



Mapa wystawy słonecznej stworzona przy realizacji kryterium 6.

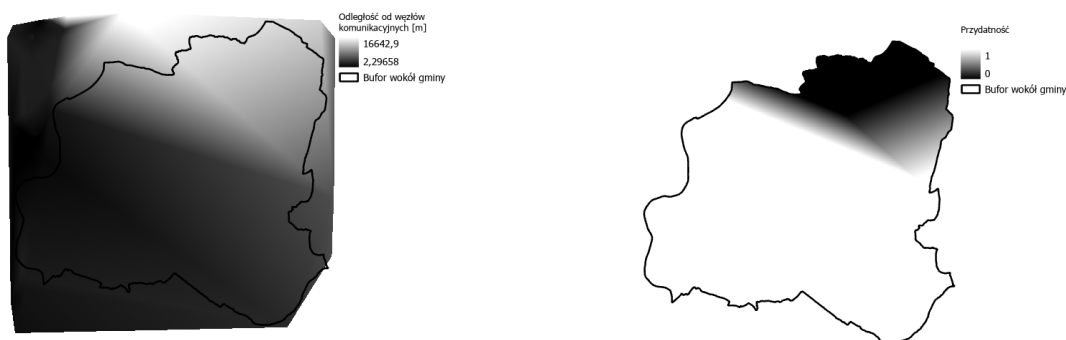


Mapa przydatności dla kryterium 6.

Rysunek 24: Wyniki dla kryterium 6.

Pod względem wystawy słonecznej spora część gminy ma korzystne warunki

### 5.2.7 Kryterium 7.



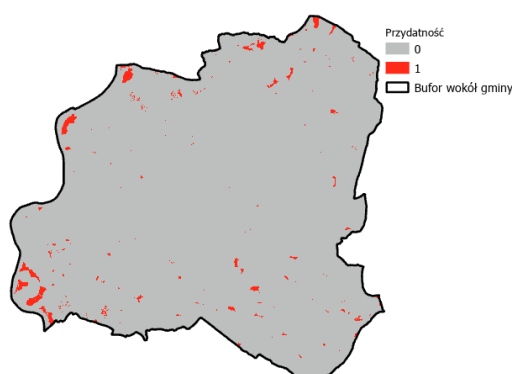
Odległość od węzłów komunikacyjnych (w metrach).

Mapa przydatności dla kryterium 7.

Rysunek 25: Wyniki dla kryterium 7.

W tym kryterium duża część gminy charakteryzuje się największą możliwą przydatnością.

### 5.2.8 Ocena przydatności terenu



Rysunek 26: Wynik łączenia kryteriów ostrych.

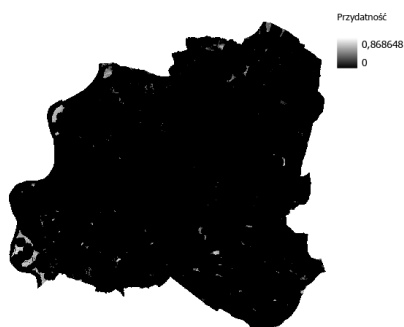


Wynik łączenia kryteriów rozmytych w podejściu z równymi wagami.

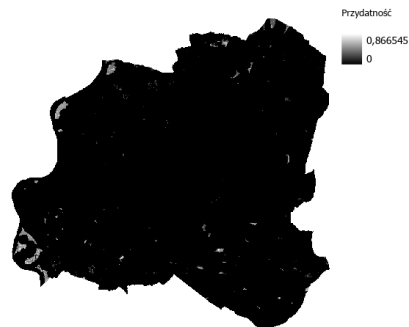
Wynik łączenia kryteriów rozmytych w podejściu z różnymi wagami.

Rysunek 27: Porównanie wyników łączenia kryteriów rozmytych w obu podejściach



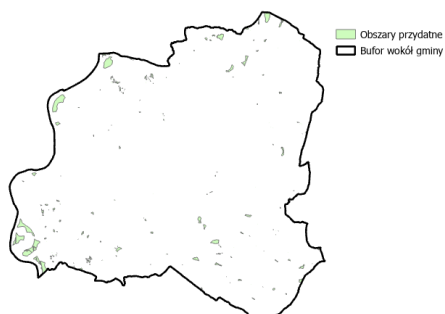


Wynik łączenia kryteriów ostrych i rozmytych w podejściu z równymi wagami.

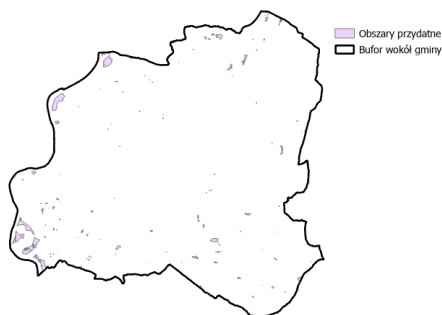


Wynik łączenia kryteriów ostrych i rozmytych w podejściu z różnymi wagami.

Rysunek 28: Porównanie wyników łączenia kryteriów ostrych i rozmytych w obu podejściach



Obszary przydatne w podejściu z równymi wagami



Obszary przydatne w podejściu z różnymi wagami

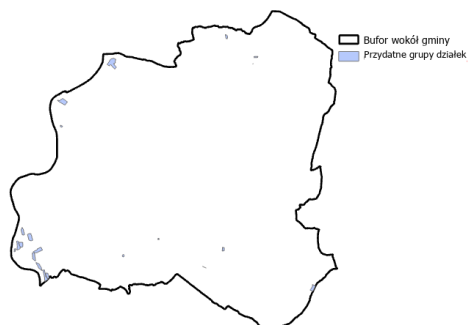
Rysunek 29: Porównanie obszarów przydatnych w obu podejściach

Duża część gminy jest nieprzydatna pod względem inwestycji. Niewielkie obszary przydatne znajdują się głównie przy granicach gminy.

### 5.2.9 Wybranie przydatnych działek/grup działek



Poligony powstałe w wyniku łączenia przydatnych działek w podejściu z równymi wagami



Poligony powstałe w wyniku łączenia przydatnych działek w podejściu z różnymi wagami

Rysunek 30: Porównanie poligonów powstałych w wyniku łączenia przydatnych działek w obu podejściach



Działki spełniające kryterium powierzchni i szerokości w podejściu z równymi wagami

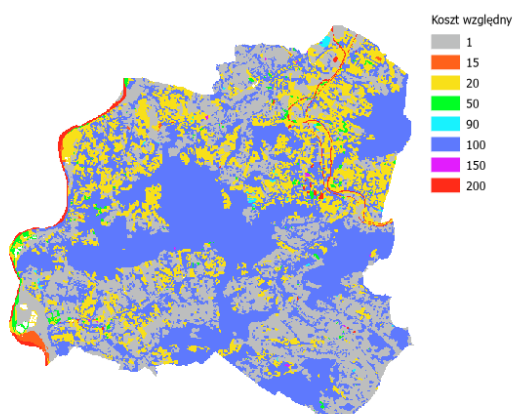


Działki spełniające kryterium powierzchni i szerokości w podejściu z różnymi wagami

Rysunek 31: Porównanie działek spełniających kryterium powierzchni i szerokości w obu podejściach

W obu podejściach uzyskano podobne wyniki. Przydatne grupy działek są zlokalizowane przy zachodniej granicy gminy. W podejściu z równymi wagami są one większe

#### 5.2.10 Koszt przyłącza do sieci



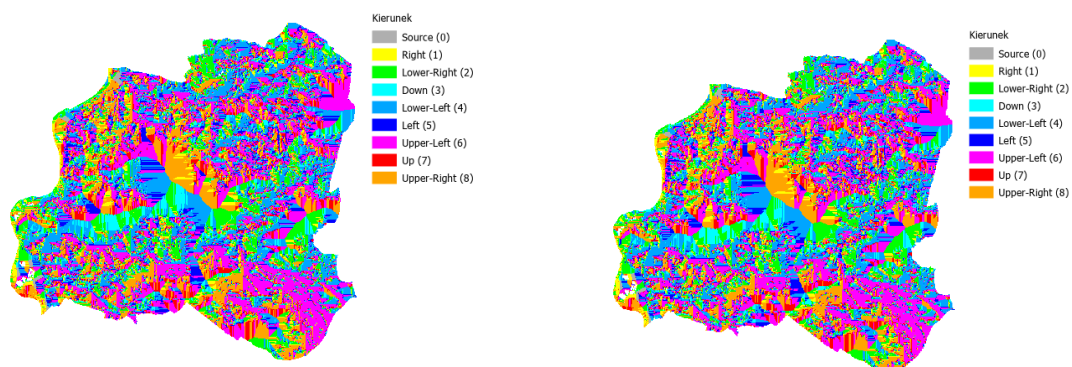
Rysunek 32: Mapa kosztów względnych



Mapa kosztów skumulowanych w podejściu z wagami równymi

Mapa kosztów skumulowanych w podejściu z wagami różnymi

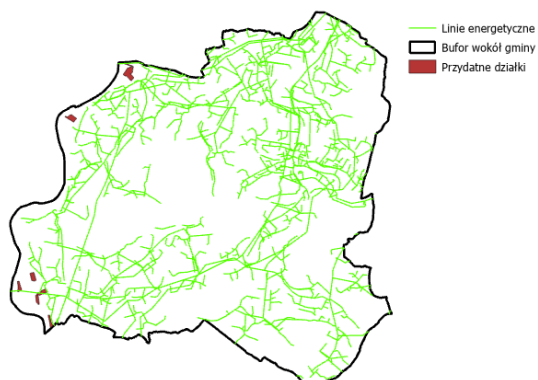
Rysunek 33: Porównanie map kosztów skumulowanych w obu podejściach



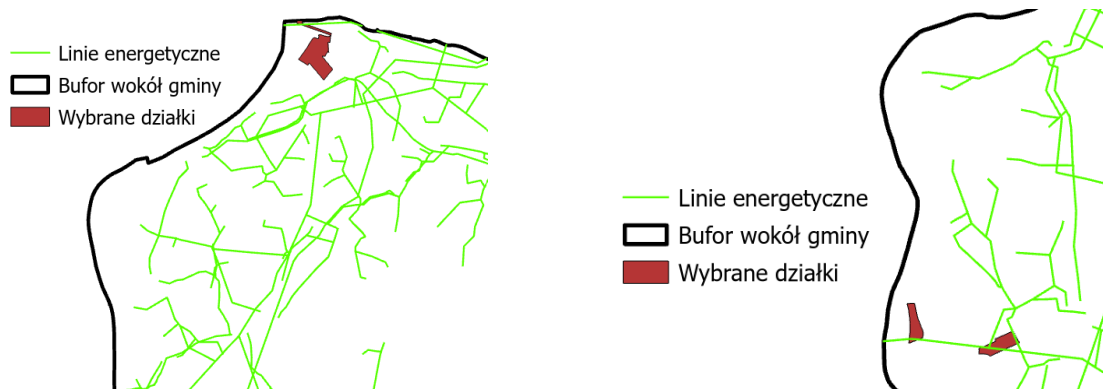
Mapa kierunków w podejściu z wagami równymi

Mapa kierunków w podejściu z wagami różnymi

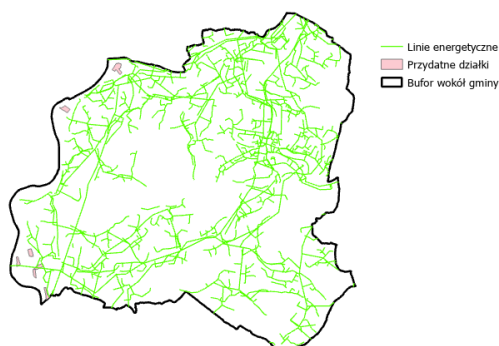
Rysunek 34: Porównanie map kierunków w obu podejściach



Rysunek 35: Mapa przedstawiająca przydatne działki i linie energetyczne w podejściu z wagami równymi



Rysunek 36: Przydatne działki, przez które przechodzą linie energetyczne w podejściu z wagami równymi



Rysunek 37: Mapa przedstawiająca przydatne działki i linie energetyczne w podejściu z wagami różnymi



Rysunek 38: Przydatne działki, przez które przechodzą linie energetyczne w podejściu z wagami różnym

Ponieważ w obu wariantach linie energetyczne przechodzą przez niektóre działki, nie doszło do utworzenia widocznej ścieżki przyłącza. Jest ona reprezentowana jako pojedynczy piksel, w którym teren działki i sieć energetyczna się przecinają. Za najlepsze pod względem inwestycji działki można uznać te, których teren przecina się z liniami energetycznymi.

### 5.3 Zauważone błędy i problemy

W danych udostępnianych na Geoportalu warstwy pokrycia terenu nie mają atrybutu X\_KOD, dlatego trzeba było korzystać z danych archiwalnych dostarczonych przez prowadzącą. Do prze-

przewodzenia analizy na terenie gminy Pleśna wystarczyły dane z jednego powiatu, co za tym idzie stworzona część skryptu wyszukująca warstwy oraz łącząca warstwy z dwóch powiatów generowała błędy. Aby pozbyć się tego błędu należało przypisać odpowiednim zmiennym ścieżki do odpowiednich warstw. Aby zwiększyć uniwersalność modelu można by dodać mechanizm obsługujący różne liczby potrzebnych warstw.