

# voronoi\_diagrams\_01

July 2, 2025

```
[ ]: # CONFIG CELL
from notebook_utils import set_root_directory

set_root_directory()

[ ]: import geopandas as gpd
import pandas as pd

from app import constants
from app.missing_values_percentage_filter import MissingValuesPercentageFilter
from app.voronoi_transformer import VoronoiTransformer

[ ]: SENSOR_METADATA = "input_files/sensor_metadata.parquet"
MEASUREMENTS_24H = "input_files/measurements_24h.parquet"
POLAND_POLYGON = "polska.zip"

OUTPUT_FILE = "input_files/sensor_metadata_24h_with_voronoi.parquet"

[ ]: poland = gpd.read_file(POLAND_POLYGON)
sensor_metadata = pd.read_parquet(SENSOR_METADATA)
measurements = pd.read_parquet(MEASUREMENTS_24H)

[ ]: vt = VoronoiTransformer(
    mask_polygon=poland,
    buffer_size=constants.BUFFER_SIZE,
    buffer_points_amount=constants.BUFFER_POINTS_AMOUNT,
)

[ ]: for variable in constants.TARGET_VARIABLES:
    if variable not in measurements.columns:
        print(f"Variable {variable} not found in measurements. Skipping Voronoi_
↳ transformation.")
        continue

    mvpf = MissingValuesPercentageFilter(threshold=constants.
↳ MISSING_VALUES_THRESHOLD)
    filtered_measurements = mvpf.fit_transform(X=measurements,
↳ variable=variable)
```

```

    for year in range(2000, 2024):
        unique_sensor_ids = filtered_measurements.query(f"`{constants.YEAR}` == {year}")
        constants.UNIQUE_ID
    ].unique()

    sensor_metadata_subset = sensor_metadata.query(
        f"`{constants.SENSOR_ID}` in {unique_sensor_ids.tolist()}"
    )

    if sensor_metadata_subset.empty:
        print(
            f"No sensor metadata found for year {year}, variable {variable}."
        )
        continue

    gdf_sensor_metadata = gpd.GeoDataFrame(
        sensor_metadata_subset,
        geometry=gpd.points_from_xy(
            sensor_metadata_subset[constants.LONGITUDE],
            sensor_metadata_subset[constants.LATITUDE],
        ),
        crs=constants.GLOBAL_EPSG,
    )
    gdf_sensor_metadata = gdf_sensor_metadata.to_crs(constants.POLAND_EPSG)
    gdf_sensor_metadata = gdf_sensor_metadata.reset_index(drop=True)
    try:
        gdf_sensor_metadata[constants.VORONOI_GEOMETRY] = vt.fit_transform(
            X=gdf_sensor_metadata
        )
    except Exception as e:
        print(f"Error during Voronoi transformation for year {year}, variable {variable}: {e}")
        continue

    sensor_metadata = sensor_metadata.merge(
        gdf_sensor_metadata[[constants.SENSOR_ID, constants.VORONOI_GEOMETRY]],
        on=constants.SENSOR_ID,
        how="left",
    )

    sensor_metadata = sensor_metadata.rename(
        columns={constants.VORONOI_GEOMETRY: f"{variable}_{year}_{constants.VORONOI_GEOMETRY}"}

```

```
)
```

```
[ ]: sensor_metadata = gpd.GeoDataFrame(  
    sensor_metadata,  
    geometry=gpd.points_from_xy(  
        sensor_metadata[constants.LONGITUDE],  
        sensor_metadata[constants.LATITUDE],  
    ),  
    crs=constants.GLOBAL_EPSG,  
)  
sensor_metadata = sensor_metadata.to_crs(constants.POLAND_EPSG)  
sensor_metadata.to_parquet(f"{OUTPUT_FILE}")
```