Q1: Spatial Reasoning & Data Filtering [5 Marks]

```
import pandas as pd
import numpy as np
import os
import geopandas as gpd


delhi_ncr = gpd.read_file("/content/data/delhi_ncr_region.geojson").to_crs(epsg=32644)
delhi_airshed = gpd.read_file("/content/data/delhi_airshed.geojson").to_crs(epsg=32644)
```

Plot the Delhi-NCR shapefile using matplotlib and overlay a 60×60 km uniform grid (1 marks)

```
from shapely.geometry import box

xmin, ymin, xmax, ymax = delhi_ncr.total_bounds
grid_size = 60000
cols = list(range(int(xmin), int(xmax), grid_size))
rows = list(range(int(ymin), int(ymax), grid_size))
polygons = []

for x in cols:
    for y in rows:
        polygons.append(box(x, y, x + grid_size, y + grid_size))

grid = gpd.GeoDataFrame({'geometry': polygons}, crs=delhi_ncr.crs)


from google.colab import output
output.enable_custom_widget_manager()
```

Support for third party widgets will remain active for the duration of the session. To disable support:

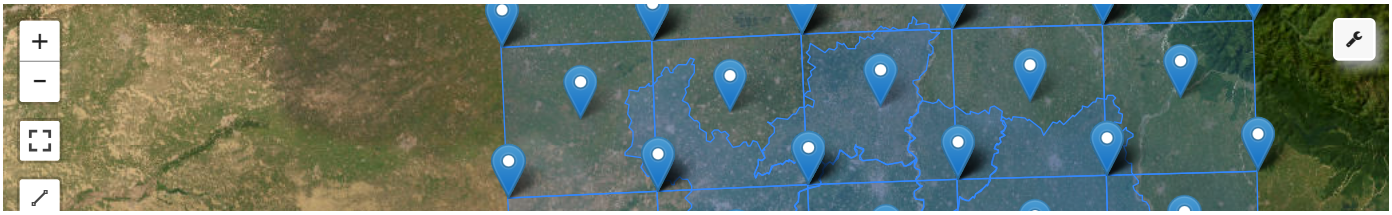
```
from google.colab import output
output.disable_custom_widget_manager()
```

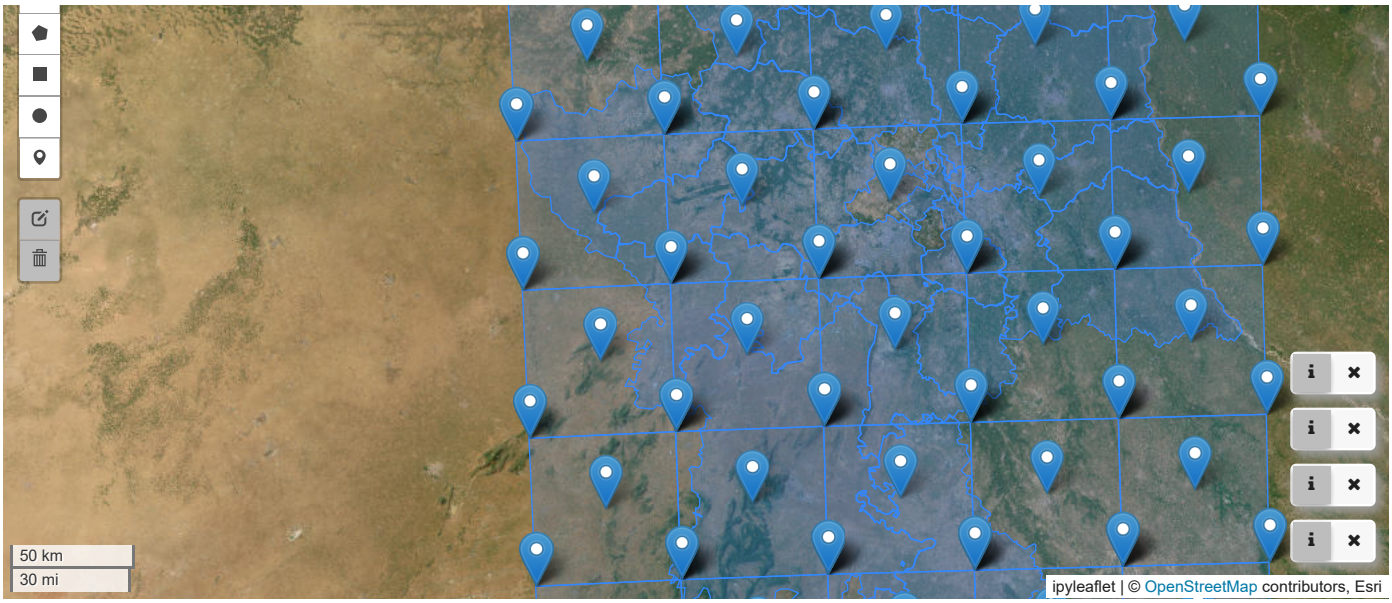
Overlay this grid on a satellite basemap using geemap or leafmap (1 mark)

```
!pip install leafmap
import leafmap

grid_4326 = grid.to_crs(epsg=4326)
delhi_ncr_4326 = delhi_ncr.to_crs(epsg=4326)

m = leafmap.Map(center=[28.6, 77.2], zoom=8)
m.add_basemap("SATELLITE")

m.add_gdf(delhi_ncr_4326, layer_name="Delhi-NCR Boundary")
m.add_gdf(grid_4326, layer_name="60x60 Grid")

m.save("delhi_ncr_grid_leafmap.html")

m
```

```
Collecting leafmap
  Downloading leafmap-0.48.6-py2.py3-none-any.whl.metadata (16 kB)
Collecting anywidget (from leafmap)
  Downloading anywidget-0.9.18-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: bqplot in /usr/local/lib/python3.11/dist-packages (from leafmap) (0.12.45)
Requirement already satisfied: duckdb in /usr/local/lib/python3.11/dist-packages (from leafmap) (1.2.2)
Requirement already satisfied: folium in /usr/local/lib/python3.11/dist-packages (from leafmap) (0.19.7)
Requirement already satisfied: gdown in /usr/local/lib/python3.11/dist-packages (from leafmap) (5.2.0)
Collecting geojson (from leafmap)
  Downloading geojson-3.2.0-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: geopandas in /usr/local/lib/python3.11/dist-packages (from leafmap) (1.0.1)
Requirement already satisfied: ipyevents in /usr/local/lib/python3.11/dist-packages (from leafmap) (2.0.2)
Requirement already satisfied: ipyfilechooser in /usr/local/lib/python3.11/dist-packages (from leafmap) (0.6.0)
Requirement already satisfied: ipyleaflet in /usr/local/lib/python3.11/dist-packages (from leafmap) (0.20.0)
Collecting ipyvuetify (from leafmap)
  Downloading ipyvuetify-1.11.3-py2.py3-none-any.whl.metadata (7.5 kB)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.11/dist-packages (from leafmap) (7.7.1)
Collecting maplibre (from leafmap)
  Downloading maplibre-0.3.4-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from leafmap) (3.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from leafmap) (2.0.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from leafmap) (2.2.2)
Requirement already satisfied: plotly in /usr/local/lib/python3.11/dist-packages (from leafmap) (5.24.1)
Collecting pystac-client (from leafmap)
  Downloading pystac_client-0.8.6-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: python-box in /usr/local/lib/python3.11/dist-packages (from leafmap) (7.3.2)
Requirement already satisfied: scooby in /usr/local/lib/python3.11/dist-packages (from leafmap) (0.10.1)
Collecting whiteboxgui (from leafmap)
  Downloading whiteboxgui-2.3.0-py2.py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: xyzservices in /usr/local/lib/python3.11/dist-packages (from leafmap) (2025.4.0)
Collecting psygnal>=0.8.1 (from anywidget->leafmap)
  Downloading psygnal-0.14.0-cp311-cp311-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (6.0 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.11/dist-packages (from anywidget->leafmap) (4.14.0)
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (6.17.1)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (0.2.0)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (5.7.1)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (3.6.10)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (7.34.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from ipywidgets->leafmap) (3.0.15)
Requirement already satisfied: traittypes>=0.0.6 in /usr/local/lib/python3.11/dist-packages (from bqplot->leafmap) (0.2.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->leafmap) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->leafmap) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->leafmap) (2025.2)
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from folium->leafmap) (0.8.1)
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.11/dist-packages (from folium->leafmap) (3.1.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from folium->leafmap) (2.32.3)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from gdown->leafmap) (4.13.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from gdown->leafmap) (3.18.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from gdown->leafmap) (4.67.1)
Requirement already satisfied: pyogrio>=0.7.2 in /usr/local/lib/python3.11/dist-packages (from geopandas->leafmap) (0.11.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from geopandas->leafmap) (24.2)
Requirement already satisfied: pyproj>=3.3.0 in /usr/local/lib/python3.11/dist-packages (from geopandas->leafmap) (3.7.1)
Requirement already satisfied: shapely>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from geopandas->leafmap) (2.1.1)
Requirement already satisfied: jupyter-leaflet<0.21,>=0.20 in /usr/local/lib/python3.11/dist-packages (from ipyleaflet->leafmap) (0.20.0
Collecting ipyvue<2,>=1.7 (from ipyvuetify->leafmap)
  Downloading ipyvue-1.11.2-py2.py3-none-any.whl.metadata (1.1 kB)
Collecting eval-type-backport (from maplibre->leafmap)
  Downloading eval_type_backport-0.2.2-py3-none-any.whl.metadata (2.2 kB)
Requirement already satisfied: pydantic>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from maplibre->leafmap) (2.11.7)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (4.58.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (1.4.8)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->leafmap) (3.2.3)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly->leafmap) (8.5.0)
Collecting pystac>=1.10.0 (from pystac[validation]>=1.10.0->pystac-client->leafmap)
  Downloading pystac-1.13.0-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: ipytree in /usr/local/lib/python3.11/dist-packages (from whiteboxgui->leafmap) (0.2.2)
Collecting whitebox (from whiteboxgui->leafmap)
  Downloading whitebox-2.3.6-py2.py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: debugpy>=1.0 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->leafmap) (1.8
Requirement already satisfied: jupyter-client>=6.1.12 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->lea
Requirement already satisfied: matplotlib-inline>=0.1 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->lea
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->leafmap) (1.6
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->leafmap) (5.9.5)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->leafmap) (24.0.1
Requirement already satisfied: tornado>=6.1 in /usr/local/lib/python3.11/dist-packages (from ipykernel>=4.5.1->ipywidgets->leafmap) (6.4
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (7
Collecting jedi>=0.16 (from ipython>=4.0.0->ipywidgets->leafmap)
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from ipython>=4.
```

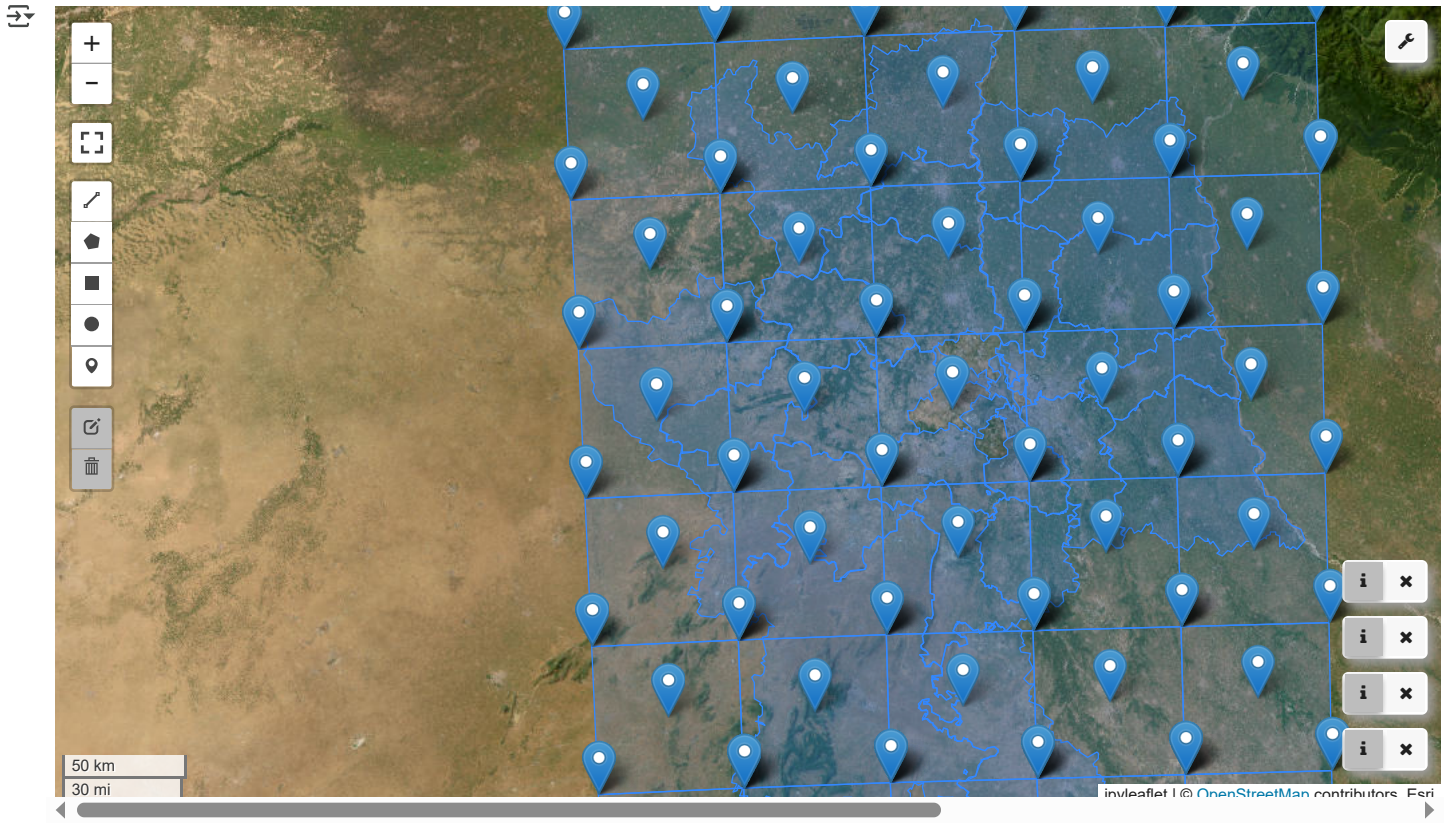
```
Requirement already satisfied: pygments in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (2.19.2)
Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (0.2.0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages (from ipython>=4.0.0->ipywidgets->leafmap) (4.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2>=2.9->folium->leafmap) (3.0.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2.5.3->maplibre->leafma
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2.5.3->maplibre->leafmap
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic>=2.5.3->maplibre->leaf
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from pyogrio>=0.7.2->geopandas->leafmap) (2025.6.15)
Requirement already satisfied: jsonschema~=4.18 in /usr/local/lib/python3.11/dist-packages (from pystac[validation]>=1.10.0->pystac-clie
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->leafmap) (1.17.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->folium->leafmap) (3.4
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->folium->leafmap) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->folium->leafmap) (2.4.0)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.11/dist-packages (from widgetsnbextension~=3.6.0->ipywidgets->l
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown->leafmap) (2.7)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown->leafmap)
Requirement already satisfied: Click>=6.0 in /usr/local/lib/python3.11/dist-packages (from whitebox->whiteboxgui->leafmap) (8.2.1)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-packages (from jedi>=0.16->ipython>=4.0.0->ipywidge
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-packages (from jsonschema~=4.18->pystac[validation]>=1.10
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.11/dist-packages (from jsonschema~=4.18->p
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.11/dist-packages (from jsonschema~=4.18->pystac[validation]
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/dist-packages (from jsonschema~=4.18->pystac[validation]>=1.1
Requirement already satisfied: jupyter-core>=4.6.0 in /usr/local/lib/python3.11/dist-packages (from jupyter-client>=6.1.12->ipykernel>=4
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->
Requirement already satisfied: nbformat in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipy
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0-
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.11/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-packages (from pexpect>4.3->ipython>=4.0.0->ipywidgets-
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ip
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.11/dist-packages (from jupyter-core>=4.6.0->jupyter-client>=6
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.11/dist-packages (from nbclassic>=0.4.7->notebook>=4.4.1->
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.11/dist-packages (from bleach[css]!=5.0.0->nbconvert>=5->notebook
Requirement already satisfied: defusedxml in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbexten
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1->widge
Requirement already satisfied: mistune<4,>=2.0.3 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgets
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnb
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.11/dist-packages (from nbconvert>=5->notebook>=4.4.1->widg
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.11/dist-packages (from nbformat->notebook>=4.4.1->widgetsn
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.11/dist-packages (from argon2-cffi->notebook>=4.4.1->widge
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from bleach!=5.0.0->bleach[css]!=5.0.0->nbconver
Requirement already satisfied: tinycss2<1.5,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from bleach[css]!=5.0.0->nbconvert>=5->n
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/python3.11/dist-packages (from notebook-shim>=0.2.3->nbclassic>=
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from argon2-cffi-bindings->argon2-cffi->notebook>
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi
Requirement already satisfied: anyio>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=0.2.
Requirement already satisfied: websocket-client in /usr/local/lib/python3.11/dist-packages (from jupyter-server<3,>=1.8->notebook-shim>=
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio>=3.1.0->jupyter-server<3,>=1.8->noteb
Downloading leafmap-0.48.6-py2.py3-none-any.whl (568 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 568.1/568.1 kB 19.5 MB/s eta 0:00:00
Downloading anywidget-0.9.18-py3-none-any.whl (220 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 220.7/220.7 kB 17.3 MB/s eta 0:00:00
Downloading geojson-3.2.0-py3-none-any.whl (15 kB)
Downloading ipyvuetify-1.11.3-py2.py3-none-any.whl (6.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.3/6.3 MB 81.1 MB/s eta 0:00:00
Downloading maplibre-0.3.4-py3-none-any.whl (1.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 56.7 MB/s eta 0:00:00
Downloading pystac_client-0.8.6-py3-none-any.whl (41 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.4/41.4 kB 3.7 MB/s eta 0:00:00
Downloading whiteboxgui-2.3.0-py2.py3-none-any.whl (108 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 108.6/108.6 kB 11.7 MB/s eta 0:00:00
Downloading ipyvue-1.11.2-py2.py3-none-any.whl (2.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 101.8 MB/s eta 0:00:00
Downloading psygnal-0.14.0-cp311-cp311-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (837 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 837.4/837.4 kB 62.2 MB/s eta 0:00:00
Downloading pystac-1.13.0-py3-none-any.whl (206 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 206.8/206.8 kB 21.1 MB/s eta 0:00:00
Downloading eval_type_backport-0.2.2-py3-none-any.whl (5.8 kB)
Downloading whitebox-2.3.6-py2.py3-none-any.whl (74 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.0/74.0 kB 6.1 MB/s eta 0:00:00
Downloading jedi-0.19.2-py2.py3-none-any.whl (1.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 81.7 MB/s eta 0:00:00
Installing collected packages: whitebox, psygnal, jedi, geojson, eval-type-backport, pystac, maplibre, pystac-client, ipyvue, anywidget,
Successfully installed anywidget-0.9.18 eval-type-backport-0.2.2 geojson-3.2.0 ipyvue-1.11.2 ipyvuetify-1.11.3 jedi-0.19.2 leafmap-0.48.
```




50 km
30 mi
ipyleaflet | © OpenStreetMap contributors, Esri

Mark the four corners and the center of each grid cell (1 mark)

```
from shapely.geometry import Point

corners = []
centers = []

for cell in grid.geometry:
    minx, miny, maxx, maxy = cell.bounds

    corners.extend([
        Point(minx, miny),
        Point(maxx, miny),
        Point(minx, maxy),
        Point(maxx, maxy)
    ])

    center_x = (minx + maxx) / 2
    center_y = (miny + maxy) / 2
    centers.append(Point(center_x, center_y))

corners_gdf = gpd.GeoDataFrame(geometry=corners, crs=grid.crs).to_crs(epsg=4326)
centers_gdf = gpd.GeoDataFrame(geometry=centers, crs=grid.crs).to_crs(epsg=4326)

m.add_gdf(corners_gdf, "Grid Corners", style={"color": "blue"})
m.add_gdf(centers_gdf, "Grid Centers", style={"color": "red"})

m
```



Filter images based on whether their center coordinates fall within the grid (1 mark)

```
image_folder = "/content/images"
image_files = [f for f in os.listdir(image_folder) if f.endswith(".png")]

records = []

for filename in image_files:
    name_part = os.path.splitext(filename)[0]

    try:
        lat_str, lon_str = name_part.split("_")
```

```
            lat = float(lat_str)
            lon = float(lon_str)
            records.append({'filename': filename, 'lat': lat, 'lon': lon})
        except Exception as e:
            print(f"Skipping {filename}: {e}")

    coords_df = pd.DataFrame(records)
    coords_df.to_csv("data/image_coords.csv", index=False)



    coords_df = pd.read_csv("data/image_coords.csv")
    coords_gdf = gpd.GeoDataFrame(
        coords_df, geometry=gpd.points_from_xy(coords_df.lon, coords_df.lat), crs="EPSG:4326"
    )
    coords_gdf = coords_gdf.to_crs(epsg=32644)
```

Count and report the number of images before and after filtering (1 mark)

```
filtered = coords_gdf[coords_gdf.geometry.within(delhi_ncr.unary_union)]
print("Total Images:", len(coords_df))
print("Filtered Images:", len(filtered))
```

```
Total Images: 1206
Filtered Images: 1206
```

Q2: Label Construction & Dataset Preparation [10 Marks]

For each image, extract a 128×128 patch from the land_cover.tif centered at the image's coordinate (2 marks)

```
!pip install rasterio
import rasterio

land_cover = rasterio.open("/content/data/worldcover_bbox_delhi_ncr_2021.tif")
coords_gdf = coords_gdf.to_crs(land_cover.crs)

image_filenames = []
land_cover_patches = []

for idx, row in coords_gdf.iterrows():
    x, y = row.geometry.x, row.geometry.y

    try:
        row_idx, col_idx = land_cover.index(x, y)

        half_size = 64
        window = rasterio.windows.Window(
            col_off=col_idx - half_size,
            row_off=row_idx - half_size,
            width=128,
            height=128
        )

        patch = land_cover.read(1, window=window)

        image_filenames.append(row['filename'])
        land_cover_patches.append(patch)

    except Exception as e:
        print(f"Error processing image {row['filename']}: {e}")
        continue
```

```
Collecting rasterio
  Downloading rasterio-1.4.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.1 kB)
Collecting affine (from rasterio)
  Downloading affine-2.4.0-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.11/dist-packages (from rasterio) (25.3.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from rasterio) (2025.6.15)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.11/dist-packages (from rasterio) (8.2.1)
Collecting cligj>=0.5 (from rasterio)
  Downloading cligj-0.7.2-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: numpy>=1.24 in /usr/local/lib/python3.11/dist-packages (from rasterio) (2.0.2)
```

```
Collecting click-plugins (from rasterio)
  Downloading click_plugins-1.1.1.2-py2.py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.11/dist-packages (from rasterio) (3.2.3)
Downloading rasterio-1.4.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (22.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22.2/22.2 MB 107.6 MB/s eta 0:00:00
Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Downloading affine-2.4.0-py3-none-any.whl (15 kB)
Downloading click_plugins-1.1.1.2-py2.py3-none-any.whl (11 kB)
Installing collected packages: cligj, click-plugins, affine, rasterio
Successfully installed affine-2.4.0 click-plugins-1.1.1.2 cligj-0.7.2 rasterio-1.4.3
```

Assign a label using the mode (most frequent) land cover class in the patch (2 marks)

```
from scipy.stats import mode

labels = []

for patch in land_cover_patches:
    flat = patch.flatten()
    dominant_class = mode(flat, keepdims=True)[0][0]
    labels.append(dominant_class)
```

Map ESA class codes to 11 standardized labels (1 mark)

```
ESA_TO_LABEL = {
    10: 'Tree Cover', 20: 'Shrubland', 30: 'Grassland', 40: 'Cropland', 50: 'Built-up',
    60: 'Bare/Sparse Veg', 70: 'Snow/Ice', 80: 'Water', 90: 'Wetland', 95: 'Mangrove', 100: 'Moss/Lichen'
}

esa_codes = sorted(ESA_TO_LABEL.keys())
label_to_index = {code: i for i, code in enumerate(esa_codes)}
index_to_label = {i: ESA_TO_LABEL[code] for code, i in label_to_index.items()}


label_indices = [label_to_index.get(lbl, -1) for lbl in labels]
```

Handle edge cases and discuss treatment of no-data pixels or mixed class dominance (2 marks)

```
valid_images = []
valid_labels = []

for img, lbl in zip(image_filenames, label_indices):
    if lbl != -1:
        valid_images.append(img)
        valid_labels.append(lbl)

print("✅ Total valid image-label pairs:", len(valid_images))
```

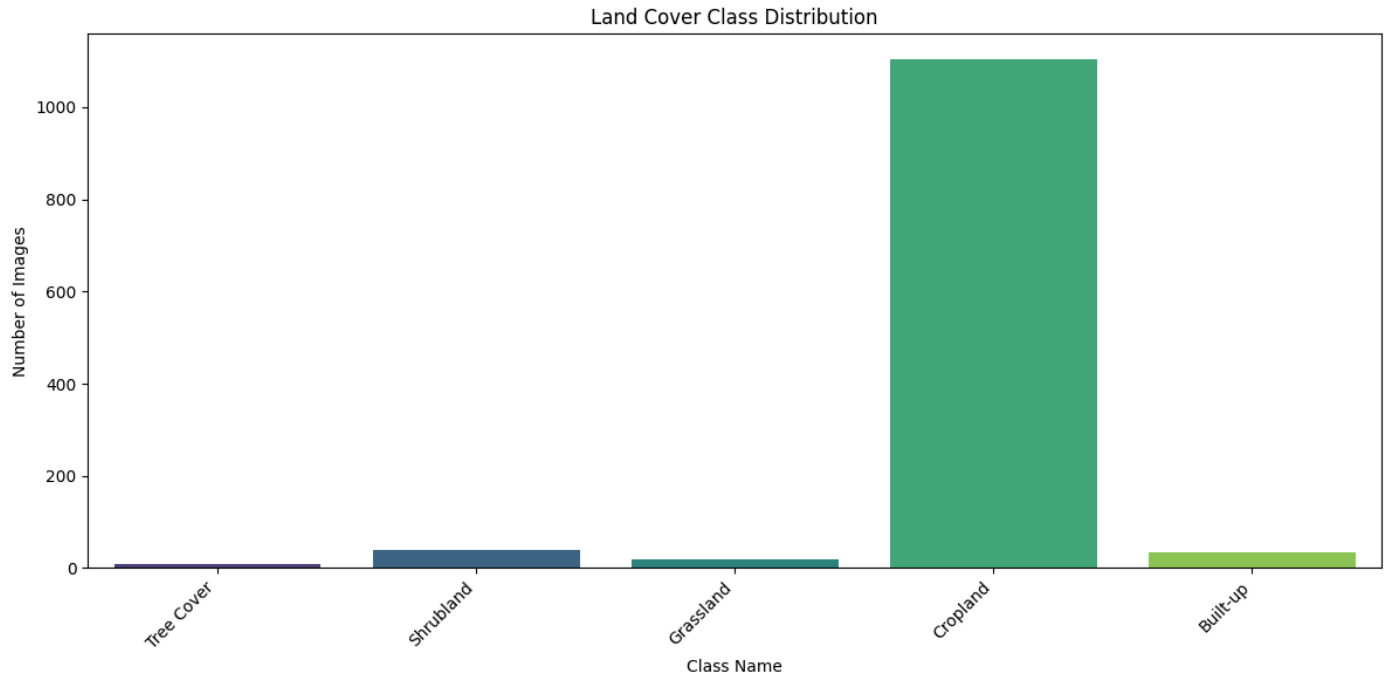
```
✅ Total valid image-label pairs: 1206
```

Perform a 60/40 train-test split randomly (1 mark)

```
from sklearn.model_selection import train_test_split

dataset = pd.DataFrame({'image': valid_images, 'label_idx': valid_labels})

label_counts = dataset['label_idx'].value_counts()
valid_classes = label_counts[label_counts >= 2].index
dataset = dataset[dataset['label_idx'].isin(valid_classes)].reset_index(drop=True)

train_df, test_df = train_test_split(
    dataset,
    test_size=0.4,
    stratify=dataset['label_idx'],
    random_state=42
)
```

Visualize class distribution and discuss balance (or imbalance) (2 marks)

```
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter

label_counts = Counter(valid_labels)

labels_sorted = sorted(label_counts.keys())
counts = [label_counts[lbl] for lbl in labels_sorted]
class_names = [index_to_label[lbl] for lbl in labels_sorted]

plt.figure(figsize=(12, 6))
sns.barplot(x=class_names, y=counts, palette='viridis')
plt.xticks(rotation=45, ha='right')
plt.title("Land Cover Class Distribution")
plt.xlabel("Class Name")
plt.ylabel("Number of Images")
plt.tight_layout()
plt.show()
```



Q3: Model Training & Evaluation [10 Marks]

Train a CNN classifier (e.g., ResNet18) on the training set (3 marks)

```
import torch
from torch import nn
from torch.utils.data import Dataset, DataLoader
from torchvision import models, transforms
from PIL import Image


class LandCoverDataset(Dataset):
    def __init__(self, df, image_dir, transform=None):
        self.df = df.reset_index(drop=True)
        self.image_dir = image_dir
        self.transform = transform

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        img_path = os.path.join(self.image_dir, self.df.loc[idx, "image"])
```

```
        image = Image.open(img_path).convert("RGB")
        label = self.df.loc[idx, "label_idx"]

        if self.transform:
            image = self.transform(image)

        return image, label


transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.ToTensor(),
])



batch_size = 32
train_dataset = LandCoverDataset(train_df, image_dir="images", transform=transform)
test_dataset = LandCoverDataset(test_df, image_dir="images", transform=transform)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True, num_workers=2)
test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False, num_workers=2)



num_classes = len(set(train_df['label_idx']))

model = models.resnet18(pretrained=False)
model.fc = nn.Linear(model.fc.in_features, num_classes)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = model.to(device)



criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)


for epoch in range(5):
    model.train()
    total_loss = 0

    for imgs, labels in train_loader:
        imgs, labels = imgs.to(device), labels.to(device)

        optimizer.zero_grad()
        outputs = model(imgs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        total_loss += loss.item()

    print(f"Epoch {epoch+1}, Loss: {total_loss / len(train_loader):.4f}")
```

```
Epoch 1, Loss: 0.5342
Epoch 2, Loss: 0.3602
Epoch 3, Loss: 0.3454
Epoch 4, Loss: 0.2907
Epoch 5, Loss: 0.2751
```

Evaluate using a custom F1 score implementation (2 marks)

```
from sklearn.metrics import f1_score

model.eval()
y_true, y_pred = [], []

with torch.no_grad():
    for imgs, labels in test_loader:
        imgs = imgs.to(device)
        outputs = model(imgs)
        preds = outputs.argmax(dim=1).cpu().numpy()
        y_pred.extend(preds)
        y_true.extend(labels.numpy())
```

```
print("Custom F1:", f1_score(y_true, y_pred, average='weighted'))
```

```
Custom F1: 0.3038895232349067
```

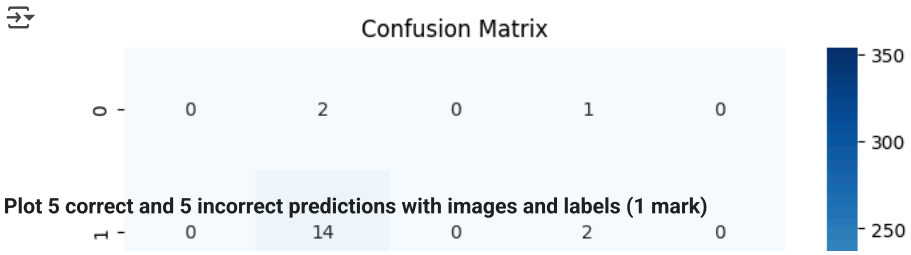
Evaluate using torchmetrics.F1Score and compare results (2 marks)

```
from torchmetrics.classification import MulticlassF1Score

# Step 1: Collect predictions and labels
all_preds = []
all_labels = []

with torch.no_grad():
    for imgs, labels in test_loader:
        imgs, labels = imgs.to(device), labels.to(device)
        outputs = model(imgs)
        preds = outputs.argmax(dim=1)
        all_preds.append(preds)
        all_labels.append(labels)

# Step 2: Concatenate all predictions and labels
all_preds = torch.cat(all_preds)
all_labels = torch.cat(all_labels)

# Step 3: Define the torchmetrics F1Score correctly
num_classes = len(torch.unique(all_labels))
f1_metric = MulticlassF1Score(num_classes=num_classes, average='weighted').to(device)

# Step 4: Compute the F1 score
f1_score_torchmetrics = f1_metric(all_preds, all_labels)
print("Torchmetrics F1:", f1_score_torchmetrics.item())
```

```
Torchmetrics F1: 0.30388951301574707
```

Show and explain a confusion matrix (2 marks)

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```



Plot 5 correct and 5 incorrect predictions with images and labels (1 mark)

```
correct = []
incorrect = []

model.eval()

with torch.no_grad():
    for imgs, labels in test_loader:
        imgs = imgs.to(device)
        labels = labels.to(device)

        outputs = model(imgs)
        _, preds = torch.max(outputs, 1)

        for i in range(len(labels)):
            img = imgs[i].cpu()
            pred_label = preds[i].item()
            true_label = labels[i].item()

            img_np = img.permute(1, 2, 0).numpy()

            if pred_label == true_label and len(correct) < 5:
                correct.append((img_np, true_label, pred_label))
            elif pred_label != true_label and len(incorrect) < 5:
                incorrect.append((img_np, true_label, pred_label))

        if len(correct) >= 5 and len(incorrect) >= 5:
            break



def plot_predictions(predictions, title):
    plt.figure(figsize=(15, 4))
    for i, (img, true_label, pred_label) in enumerate(predictions):
        plt.subplot(1, 5, i+1)
```