import arcpy

# Access the current project, map, and layer

aprx = arcpy.mp.ArcGISProject("CURRENT")

map\_name = "Map" # Replace with your map name

layer\_name = "Accessibility Index" # Replace with your layer name

m = aprx.listMaps(map\_name)[0]

lyr = m.listLayers(layer\_name)[0]

# Verify the symbology

if lyr.symbology.renderer.type == "GraduatedColorsRenderer":

renderer = lyr.symbology.renderer

classification\_field = renderer.classificationField

# List fields in the layer

fields = [field.name for field in arcpy.ListFields(lyr.dataSource)]

# Add the fillColor field if it doesn't exist

if "fillColor" not in fields:

arcpy.AddField\_management(lyr.dataSource, "fillColor", "TEXT", field\_length=10)

print("Added field: fillColor")

# Populate the fillColor field

with arcpy.da.UpdateCursor(lyr.dataSource, ["fillColor", classification\_field]) as cursor:

for row in cursor:

value = row[1] # Value of the classification field

for i, class\_break in enumerate(renderer.classBreaks):

# Infer the lower bound for all class breaks, except the first one

if i == 0:

lower\_bound = float('-inf') # Use negative infinity for the first break

else:

lower\_bound = renderer.classBreaks[i - 1].upperBound # Previous class's upperBound

if lower\_bound <= value < class\_break.upperBound:

# Get RGB color from the dictionary

rgb = class\_break.symbol.color['RGB']

red, green, blue = rgb[0], rgb[1], rgb[2]

# Convert to hex color

hex\_color = "#{:02X}{:02X}{:02X}".format(red, green, blue)

row[0] = hex\_color # Assign hex color to the fillColor field

cursor.updateRow(row)

break

print("fillColor field populated with symbology colors.")