

Supplementary Material

Tab. S1 - Multicollinearity diagnostics for OLS (HC3): variance inflation factors (VIF) for all predictors.

Variable	VIF
NDVI trend	1.15
Slope	1.40
Aspect (sine)	1.01
Aspect (cosine)	1.02
LST	2.73
NDMI	3.17
Texture Contrast	1.51
Log-transformed distance to roads	1.28
Log-transformed distance to villages	1.53
Shrub/Open Forest (Classes 2 & 3)	7.19
Cropland (Class 4.0)	5.56
Bare/Water (Class 5.0)	3.46

Appendix 1 - Source code for preprocessing and burn-severity mapping. Well-commented GEE scripts to build pre/post-fire composites, compute NBR/dNBR, stack predictors, and export samples as used in Methods.

```
/******  
COMPLETE WILDFIRE BURN SEVERITY ANALYSIS SCRIPT (GEE)  
Research on Long-term NDVI Trends and Burn Severity Prediction  
Kashkan Watershed, Zagros Mountains, Iran  
Author: Research Team  
Version: Final  
*****/  
  
// ===== STEP 1: AOI + DEM + Slope & Aspect =====  
var aoi = ee.FeatureCollection('projects/papers-467123/assets/Marz');  
var dem = ee.Image('projects/papers-467123/assets/Dem_UTM');  
  
Map.centerObject(aoi, 9);  
Map.addLayer(aoi, {color: 'black', fillColor: '00000000'}, 'AOI - Kashkan');  
Map.addLayer(dem, {min:0, max:3000}, 'DEM (UTM)');  
  
var slope = ee.Terrain.slope(dem);  
var aspect = ee.Terrain.aspect(dem);  
Map.addLayer(slope, {min:0, max:60}, 'Slope (deg)');  
Map.addLayer(aspect, {min:0, max:360}, 'Aspect (deg)');  
  
print('AOI feature count:', aoi.size());  
print('DEM projection:', dem.projection());  
  
var projUTM = dem.projection();  
var projUTM38 = ee.Projection('EPSG:32638');  
var proj30 = projUTM.atScale(30);  
  
// ===== Landsat 8 SR + NDVI 2014-2023 for AOI =====  
var startYear = 2014;  
var endYear = 2023;  
  
// Cloud/shadow mask + scale factors  
function maskL8sr(image) {  
  var qa = image.select('QA_PIXEL');  
  var cloudBitMask = 1 << 3;  
  var shadowBitMask = 1 << 4;  
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)  
    .and(qa.bitwiseAnd(shadowBitMask).eq(0));  
  var scaled = image.select(['SR_B4', 'SR_B5']).multiply(0.0000275).add(-0.2);  
  return image.updateMask(mask)  
    .addBands(scaled, ['SR_B4', 'SR_B5'], true)  
    .copyProperties(image, ["system:time_start"]);  
}  
  
var landsatCollection = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')  
  .filterBounds(aoi)  
  .filter(ee.Filter.calendarRange(startYear, endYear, 'year'))  
  .map(maskL8sr);  
  
// NDVI  
function addNDVI(image) {  
  var ndvi = image.normalizedDifference(['SR_B5', 'SR_B4']).rename('NDVI');  
  return image.addBands(ndvi).clip(aoi);  
}
```

```
var landsatNDVI = landsatCollection.map(addNDVI);

// Example 2014 NDVI
var ndvi2014 = landsatNDVI
  .filter(ee.Filter.calendarRange(2014,2014,'year'))
  .select('NDVI').mean().clip(aoi);
Map.addLayer(ndvi2014, {min:-1, max:1, palette:['blue','white','green']}, 'NDVI 2014');

// Yearly series (May-Sep)
var years = ee.List.sequence(startYear, endYear);
var annualNDVI = ee.ImageCollection.fromImages(
  years.map(function(y) {
    var start = ee.Date.fromYMD(y, 5, 1);
    var end = ee.Date.fromYMD(y, 9, 30);
    var ndviYear = landsatNDVI.filterDate(start, end).select('NDVI').mean().clip(aoi);
    var yearImg = ee.Image.constant(y).float().rename('year');
    return ndviYear.addBands(yearImg).set('year', y);
  })
);

// NDVI linear trend (slope)
var linearFit = annualNDVI.select(['year','NDVI']).reduce(ee.Reducer.linearFit());
var slopeNDVI = linearFit.select('scale');

var ghostForest = slopeNDVI.lt(0);
Map.centerObject(aoi, 9);
Map.addLayer(slopeNDVI, {min:-0.02, max:0.02, palette:['red','yellow','green']}, 'NDVI Slope');
Map.addLayer(ghostForest.updateMask(ghostForest), {palette:['black']}, 'Ghost Forests');

print('Annual NDVI ImageCollection:', annualNDVI);
print('Slope Image:', slopeNDVI);

// Area and intensity of NDVI decline
var pixelArea = ee.Image.pixelArea();
var ghostArea = pixelArea.updateMask(ghostForest).reduceRegion({
  reducer: ee.Reducer.sum(),
  geometry: aoi,
  scale: 30,
  maxPixels: 1e13
});
print('Ghost Forest total area (m2):', ghostArea);

var ghostClasses = slopeNDVI.expression(
  "b('scale') <= -0.02 ? 3" +
  ": b('scale') <= -0.01 ? 2" +
  ": b('scale') < 0 ? 1" +
  ": 0"
).rename('Ghost_Class');

Map.addLayer(ghostClasses.updateMask(ghostClasses),
  {min:1, max:3, palette:['yellow','orange','red']},
  'Ghost Forest Classes');

var areas = ghostClasses.reduceRegion({
  reducer: ee.Reducer.frequencyHistogram(),
  geometry: aoi,
  scale: 30,
  maxPixels: 1e13
});
print('Pixel counts per Ghost Class:', areas);
```

```
// =====
// PHASE 3: Fire severity analysis (dNBR) with Sentinel-2
// =====

// 3.1 Fire windows
var preFireStart = '2024-06-10';
var preFireEnd = '2024-06-20';
var postFireStart = '2024-06-23';
var postFireEnd = '2024-07-12';

// 3.2 Sentinel-2 preprocess
var sentinel2 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED");
function maskS2clouds(image) {
  var qa = image.select('QA60');
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
  return image.updateMask(mask);
}
function addS2Indices(image) {
  var nbr = image.normalizedDifference(['B8', 'B12']).rename('NBR');
  return image.addBands(nbr);
}

// 3.3 Pre/Post composites
var preFireImg = sentinel2.filterBounds(aoi).filterDate(preFireStart, preFireEnd)
  .map(maskS2clouds).map(addS2Indices).median().clip(aoi);
var postFireImg = sentinel2.filterBounds(aoi).filterDate(postFireStart, postFireEnd)
  .map(maskS2clouds).map(addS2Indices).median().clip(aoi);

// 3.4 dNBR
var dNBR = preFireImg.select('NBR').subtract(postFireImg.select('NBR')).rename('dNBR');
Map.addLayer(dNBR, {min:-0.1, max:0.8, palette:['green','yellow','red']}, 'Fire Severity (dNBR)');

// ===== Road/Village distance - FIXED =====
var basin = ee.FeatureCollection('projects/papers-467123/assets/Marz');
var roadsFC = ee.FeatureCollection('projects/papers-467123/assets/Roads2');
var villagesFC = ee.FeatureCollection('projects/papers-467123/assets/Village');

// Use FeatureCollection.distance, avoid reproject here; set scale at
// export/visualization time
var roadDist = roadsFC.distance({searchRadius: 100000, maxError: 50})
  .clip(aoi)
  .rename('Road_Dist'); // meters

var villDist = villagesFC.distance({searchRadius: 100000, maxError: 50})
  .clip(aoi)
  .rename('Dist_Villages'); // meters

Map.addLayer(roadDist, {min:0, max:5000, palette:['#ffffd9','#41b6c4','#0c2c84']},
  'Road_Dist (m)');
Map.addLayer(villDist, {min:0, max:10000, palette:['red','yellow','blue']},
  'Dist_Villages (m)');

// Aliases used later
var roadDist30 = roadDist;
var villDist30 = villDist;

// =====
// PHASE 4.1: Slope and aspect classification (corrected version with Clip)
```

```
// =====

var slope2 = ee.Terrain.slope(dem);
var slopeClasses = slope2.expression(
  "(b(0) > 30) ? 4" +
  ": (b(0) > 15) ? 3" +
  ": (b(0) > 5) ? 2" +
  ": 1"
).rename('slope_class').clip(aoi);

var aspect2 = ee.Terrain.aspect(dem);
var aspectClasses = aspect2.expression(
  "(b(0) > 315 || b(0) <= 45) ? 1" +
  ": (b(0) > 45 && b(0) <= 135) ? 2" +
  ": (b(0) > 135 && b(0) <= 225) ? 3" +
  ": 4"
).rename('aspect_class').clip(aoi);

Map.addLayer(slopeClasses, {min:1, max:4, palette:
['#ffffcc','#aldab4','#41b6c4','#225ea8']}, 'Slope Classes');
Map.addLayer(aspectClasses, {min:1, max:4, palette:
['#d73027','#fee08b','#1a9850','#4575b4']}, 'Aspect Classes (N,E,S,W)');

// =====
// PHASE 4.2: Regional statistics (final and corrected version)
// =====

var burnedAreaMask = dNBR.mask();

// Mean dNBR by slope class
var dnbrAndSlope = dNBR.addBands(slopeClasses);
var slopeStats = dnbrAndSlope.updateMask(burnedAreaMask).reduceRegion({
  reducer: ee.Reducer.mean().group({groupField: 1, groupName: 'slope_class'}),
  geometry: aoi,
  scale: 30,
  maxPixels: 1e13
});
print('Mean dNBR per Slope Class:', slopeStats.get('groups'));

// Mean dNBR by aspect class
var dnbrAndAspect = dNBR.addBands(aspectClasses);
var aspectStats = dnbrAndAspect.updateMask(burnedAreaMask).reduceRegion({
  reducer: ee.Reducer.mean().group({groupField: 1, groupName: 'aspect_class'}),
  geometry: aoi,
  scale: 30,
  maxPixels: 1e13
});
print('Mean dNBR per Aspect Class:', aspectStats.get('groups'));

// =====
// PHASE 5: dNBR by distance from roads
// =====
var distanceClasses = roadDist30.expression(
  "b('Road_Dist') <= 100 ? 1" +
  ": b('Road_Dist') <= 500 ? 2" +
  ": b('Road_Dist') <= 1000 ? 3" +
  ": 4"
).rename('dist_class').clip(basin);

var dnbrAndDist = dNBR.addBands(distanceClasses);
var distStats = dnbrAndDist.reduceRegion({
```

```

    reducer: ee.Reducer.mean().group({groupField: 1, groupName: 'dist_class'}),
    geometry: basin,
    scale: 30,
    maxPixels: 1e13
  });
print('Mean dNBR per Distance Class:', distStats.get('groups'));

Map.addLayer(distanceClasses, {min:1, max:4, palette:
['#ffffcc','#aldab4','#41b6c4','#225ea8']}, 'Distance Classes');

// ===== Ensure ndviSlope exists & Ghost vs Non-Ghost stats =====
var region = (typeof aoi !== 'undefined') ? aoi :
              ((typeof basin !== 'undefined') ? basin :
               ee.FeatureCollection("projects/papers-467123/assets/Marz"));

var ndviSlope;
if (typeof slopeNDVI !== 'undefined') {
  ndviSlope = slopeNDVI;
  print('Using existing slopeNDVI as ndviSlope.');
```

```

} else {
  print('slopeNDVI not found: computing ndviSlope from landsatNDVI...');
  var yearsList = ee.List.sequence(startYear, endYear);
  var annualNDVI2 = ee.ImageCollection.fromImages(
    yearsList.map(function(y) {
      var start = ee.Date.fromYMD(y, 5, 1);
      var end   = ee.Date.fromYMD(y, 9, 30);
      var ndviYear = landsatNDVI.filterDate(start,
end).select('NDVI').mean().clip(region);
      var yearImg = ee.Image.constant(y).float().rename('year');
      return ndviYear.addBands(yearImg).set('year', y);
    })
  );
  var linearFit_new =
annualNDVI2.select(['year', 'NDVI']).reduce(ee.Reducer.linearFit());
  ndviSlope = linearFit_new.select('scale');
  print('Computed ndviSlope from annual NDVI.');
```

```

}

var ghostMask = ndviSlope.lt(0);
var nonGhostMask = ghostMask.not();

var dNBR_ghost = dNBR.updateMask(ghostMask).clip(region);
var dNBR_nonGhost = dNBR.updateMask(nonGhostMask).clip(region);

var reducerCombined = ee.Reducer.mean()
  .combine(ee.Reducer.stdDev(), '', true)
  .combine(ee.Reducer.count(), '', true);

var ghostStats = dNBR_ghost.reduceRegion({
  reducer: reducerCombined,
  geometry: region,
  scale: 30,
  maxPixels: 1e13
});
var nonGhostStats = dNBR_nonGhost.reduceRegion({
  reducer: reducerCombined,
  geometry: region,
  scale: 30,
  maxPixels: 1e13
});
print('🌲 Ghost Forest dNBR stats:', ghostStats);
```

```
print('🇮🇷 Non-Ghost dNBR stats:', nonGhostStats);

// =====
// Load and reclassify ESA WorldCover map (10 m)
// =====
var worldCover = ee.ImageCollection('ESA/WorldCover/v200').first().clip(aoi);
var fromClasses_ESA = [10, 20, 30, 40, 50, 60, 80];
var toClasses_ESA = [1, 2, 2, 3, 4, 4, 5];

var landcover_reclassified = worldCover.remap(fromClasses_ESA, toClasses_ESA)
  .rename('Land_Cover_Reclass');

var visParams_ESA = {
  min: 1, max: 5,
  palette: ['#006400', '#ffbb22', '#ffff4c', '#dcdcdc', '#0046c8']
};
Map.addLayer(landcover_reclassified.updateMask(landcover_reclassified.gt(0)),
visParams_ESA, 'ESA Land Cover Reclassified (10m)');

// =====
// Calculate Land Surface Temperature (LST)
// =====
var preFireLST_start = '2024-05-01';
var preFireLST_end = '2024-06-20';
var landsat8_L2 = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2');

function processLST(image) {
  var qa = image.select('QA_PIXEL');
  var cloudMask = qa.bitwiseAnd(1 << 3).eq(0);
  var lst_kelvin = image.select('ST_B10').updateMask(cloudMask);
  var lst_celsius = lst_kelvin.multiply(0.00341802).add(149.0).subtract(273.15);
  return lst_celsius.rename('LST');
}

var preFire_LST = landsat8_L2
  .filterBounds(aoi)
  .filterDate(preFireLST_start, preFireLST_end)
  .map(processLST)
  .mean()
  .clip(aoi);
Map.addLayer(preFire_LST, {min: 15, max: 50, palette: ['blue', 'yellow', 'red']}, 'Pre-
Fire LST (Celsius)');

// =====
// Calculate Moisture Index (NDMI) from pre-fire S2
// =====
var preFire_NDMI = preFireImg.normalizedDifference(['B8', 'B11']).rename('NDMI');
Map.addLayer(preFire_NDMI, {min: -0.5, max: 0.5, palette: ['red', 'yellow', 'green']},
'Pre-Fire NDMI');

// =====
// Export maps for publication (visualized RGB)
// =====
var vis_NDVI_Trend = {min: -0.02, max: 0.02, palette: ['red', 'yellow', 'green']};
var vis_Ghost_Classes = {min: 1, max: 3, palette: ['yellow', 'orange', 'red']};
var vis_dNBR = {min: -0.1, max: 0.8, palette: ['green', 'yellow', 'red']};
var vis_DEM = {min: 0, max: 3000, palette: ['#006633', '#E5FFCC', '#662A00',
'#D8D8D8', '#F5F5F5']};
var vis_Slope = {min: 0, max: 60, palette: ['white', 'yellow', 'orange',
'red']};
var vis_Road_Dist = {min: 0, max: 2000, palette: ['#ffffd9', '#41b6c4', '#0c2c84']};

var ndviTrend_RGB = slopeNDVI.visualize(vis_NDVI_Trend);
```

```
var                                ghostClasses_RGB
ghostClasses.updateMask(ghostClasses.gt(0)).visualize(vis_Ghost_Classes);
var dNBR_RGB                      = dNBR.visualize(vis_dNBR);
var dem_RGB                      = dem.visualize(vis_DEM);
var slope_RGB                   = slope.visualize(vis_Slope);
var roadDist_RGB                = roadDist30.visualize(vis_Road_Dist);

Export.image.toDrive({
  image: ndviTrend_RGB,
  description: 'Map_Export_1_NDVI_Trend',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_1_NDVI_Trend',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: ghostClasses_RGB,
  description: 'Map_Export_2_Ghost_Classes',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_2_Ghost_Classes',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: dNBR_RGB,
  description: 'Map_Export_3_dNBR_Severity',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_3_dNBR_Severity',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: dem_RGB,
  description: 'Map_Export_4_DEM',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_4_DEM',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: slope_RGB,
  description: 'Map_Export_5_Slope',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_5_Slope',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: roadDist_RGB,
  description: 'Map_Export_6_Road_Distance',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_6_Road_Distance',
```

```
region: aoi.geometry(),
scale: 30,
crs: 'EPSG:32638',
maxPixels: 1e13
});

// =====
// Export NDVI maps (first and last year) and aspect
// =====
var ndvi_2014 = annualNDVI.filter(ee.Filter.eq('year', 2014)).first().select('NDVI');
var ndvi_2023 = annualNDVI.filter(ee.Filter.eq('year', 2023)).first().select('NDVI');

var vis_NDVI = {min: 0, max: 0.8, palette: ['#F2F12D', '#347C2C', '#013220']};
var vis_Aspect = {min: 0, max: 360, palette: ['red', 'yellow', 'green', 'cyan', 'blue', 'magenta', 'red']};

var ndvi_2014_RGB = ndvi_2014.visualize(vis_NDVI);
var ndvi_2023_RGB = ndvi_2023.visualize(vis_NDVI);
var aspect_RGB = aspect.visualize(vis_Aspect);

Export.image.toDrive({
  image: ndvi_2014_RGB,
  description: 'Map_Export_7_NDVI_2014',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_7_NDVI_2014',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: ndvi_2023_RGB,
  description: 'Map_Export_8_NDVI_2023',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_8_NDVI_2023',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});
Export.image.toDrive({
  image: aspect_RGB,
  description: 'Map_Export_9_Aspect',
  folder: 'GEE_exports_Paper_Maps',
  fileNamePrefix: 'Map_9_Aspect',
  region: aoi.geometry(),
  scale: 30,
  crs: 'EPSG:32638',
  maxPixels: 1e13
});

// =====
// TEXTURE VARIABLES - Contrast, Homogeneity, Variance (3x3)
// =====
var nir8_8bit = preFireImg.select('B8').multiply(255).divide(10000).uint8();
var glcm = nir8_8bit.glcmTexture({size: 1});
var texContrast = glcm.select('B8_contrast').rename('Texture_Contrast');
var texHomogeneity = glcm.select('B8_idm').rename('Texture_Homogeneity');
var texVariance = glcm.select('B8_var').rename('Texture_Variance');

// =====
// BUILD MULTIBAND ANALYSIS IMAGE (Forest patch/edge removed)
```

```
// =====
var village_dist_raster = villDist;

var analysisImage = dNBR.rename('dNBR')
  .addBands(slopeNDVI.rename('NDVI_trend'))
  .addBands(slope.rename('Slope'))
  .addBands(aspect.rename('Aspect'))
  .addBands(roadDist30.rename('Road_Dist'))
  .addBands(landcover_reclassified.rename('Land_Cover_Reclass'))
  .addBands(preFire_LST.rename('LST'))
  .addBands(preFire_NDMI.rename('NDMI'))
  .addBands(village_dist_raster.rename('Dist_Villages'))
  .addBands(texContrast.rename('Texture_Contrast'))
  .addBands(texHomogeneity.rename('Texture_Homogeneity'))
  .addBands(texVariance.rename('Texture_Variance'));

// =====
// VARIABLE SELECTOR LIST FOR CSV EXPORT (Forest_Patch_Size removed)
// =====
var finalSelectors = [
  'dNBR', 'NDVI_trend', 'Slope', 'Aspect', 'Road_Dist',
  'Land_Cover_Reclass', 'LST', 'NDMI',
  'Dist_Villages',
  'Texture_Contrast', 'Texture_Homogeneity', 'Texture_Variance'
];

// =====
// SPATIALLY UNIFORM SAMPLING (Grid + 90m minimum distance)
// =====
var grid = aoi.geometry().coveringGrid(ee.Projection('EPSG:32638'), 5000);

for (var i = 1; i <= 5; i++) {
  var seed = 1234 + i;

  var points = grid.map(function(cell){
    return ee.FeatureCollection.randomPoints({
      region: ee.Feature(cell).geometry().intersection(aoi.geometry(), 1),
      points: ee.Number(20000).divide(grid.size()).round(),
      seed : seed
    });
  }).flatten();

  var filtered = points.map(function(pt){
    var n = points.filterBounds(pt.geometry().buffer(90)).size();
    return pt.set('n', n);
  }).filter(ee.Filter.eq('n', 1));

  var sample = analysisImage.sampleRegions({
    collection : filtered,
    scale      : 30,
    geometries : true
  }).map(function(f){
    var c = f.geometry().coordinates();
    return f.set({'longitude': c.get(0), 'latitude': c.get(1)});
  });

  Export.table.toDrive({
    collection: sample,
    description: 'Regression_DataSet_' + i,
    folder      : 'GEE_exports_Regression',
    fileNamePrefix: 'Regression_Data_Set_' + i,
    fileFormat  : 'CSV',
```

```
        selectors : finalSelectors.concat(['longitude','latitude'])
    });
}
var ordered = analysisImage.select([
    'dNBR','NDVI_trend','Slope','Aspect','Land_Cover_Reclass',
    'Road_Dist','Dist_Villages','LST','NDMI',
    'Texture_Contrast','Texture_Homogeneity','Texture_Variance'
]);
Export.image.toDrive({
    image: ordered.toFloat(),
    description: 'Analysis_Stack_30m_PythonSample',
    folder: 'GEE_exports_Regression',
    fileNamePrefix: 'analysis_stack_30m_python',
    region: aoi.geometry(),
    scale: 30,
    crs: 'EPSG:32638',
    maxPixels: 1e13,
    fileFormat: 'GeoTIFF'
});

// =====
//  SAMPLING STRATEGY DOCUMENTATION
//  =====
var samplingMetadata = ee.FeatureCollection([
    ee.Feature(null, {
        'sampling_method'      : 'Grid-stratified + 90m inhibition',
        'total_samples_file'   : 20000,
        'grid_size_m'          : 5000,
        'min_distance_m'       : 90,
        'texture_window_size'  : '3x3 pixels',
        'texture_features'     : 'Contrast, Homogeneity(IDM), Variance',
        'datasets_generated'   : 5,
        'spatial_crs'          : 'EPSG:32638',
        'coordinate_system'    : 'UTM Zone 38N'
    })
]);

Export.table.toDrive({
    collection: samplingMetadata,
    description: 'Sampling_Strategy_Documentation',
    folder      : 'GEE_exports_Regression',
    fileNamePrefix: 'Sampling_Metadata',
    fileFormat  : 'CSV'
});

print('✅ Regression datasets queued for export (forest variables removed).');
```