

# GeoAI with ArcGIS Pro

Cedric Despierre Corporon | Rami Alouta

*GIS for a Sustainable World Conference*



# GeoAI With ArcGIS Pro

## Agenda

- What is GeoAI
- What can it do?
- Where can you apply GeoAI (Sneak peak on the GP tools)
- A deeper dive into GeoAI within ArcGIS
- Getting Started | Environment & Licensing
- GeoAI in ArcGIS Pro
  - GUI Demo
- Advanced GeoAI Workflows with Python
  - Notebook Demo

# What is GeoAI?



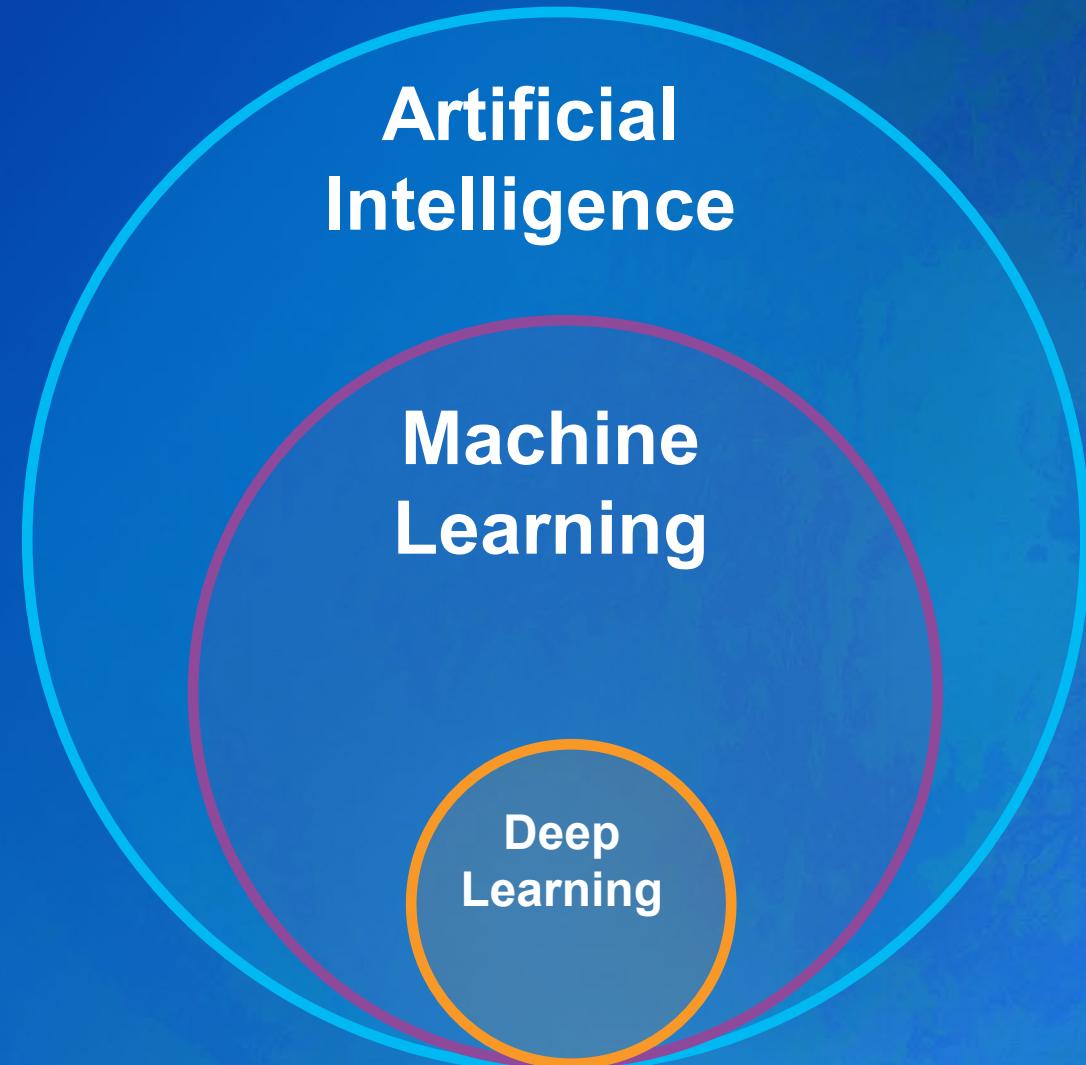
# What Is AI?

Summary:

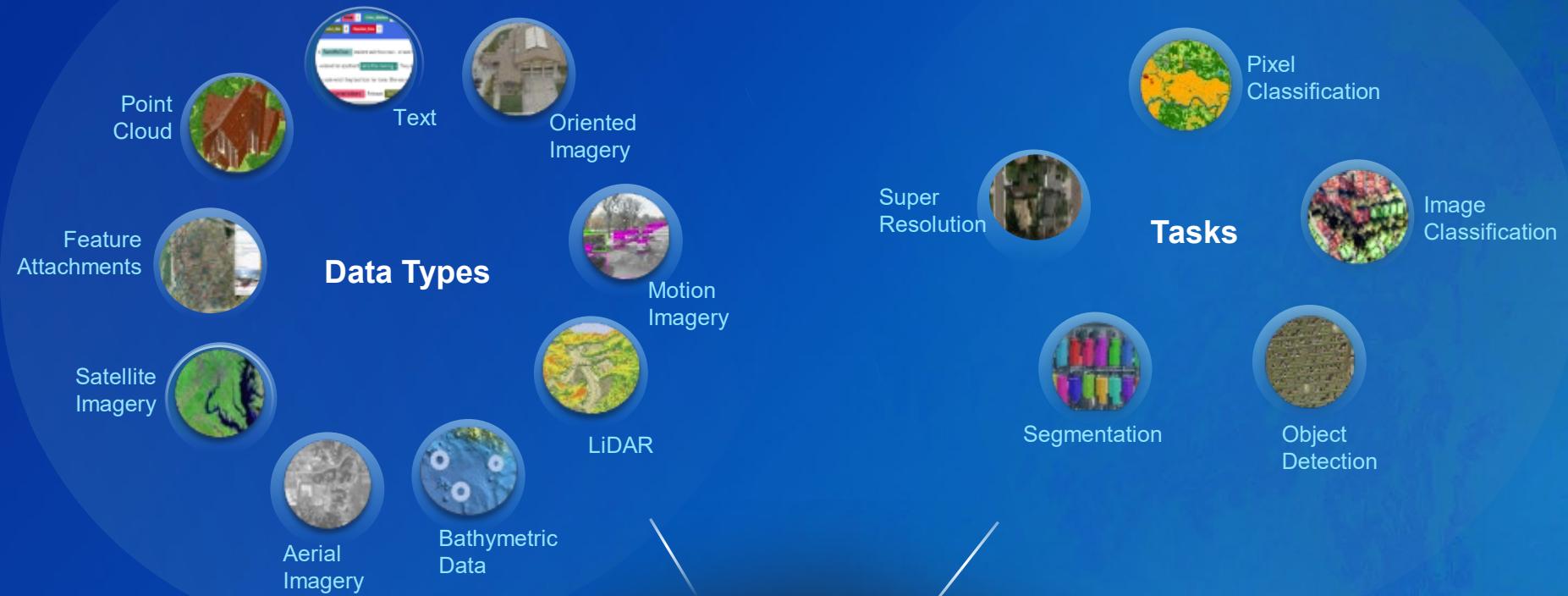
Really, machine learning (ML)

Machine learning is about extracting patterns from data to derive rules, instead of these rules being explicitly programmed.

Deep learning is a type of ML using deep neural networks to find complex patterns especially in unstructured data (such as images, text, voice, and lidar).



# GeoAI in ArcGIS



## Integration

# Things to know

PyTorch

Random Forest

Cognitive Computing

GeoAI

Dimensionality Reduction

Object Tracking  
Artificial Intelligence  
Computer Vision  
Machine Learning  
TensorFlow

Deep Learning

Support Vector Machines

CNTK Object Detection

scikit-learn

Neural Networks

Natural Language Processing

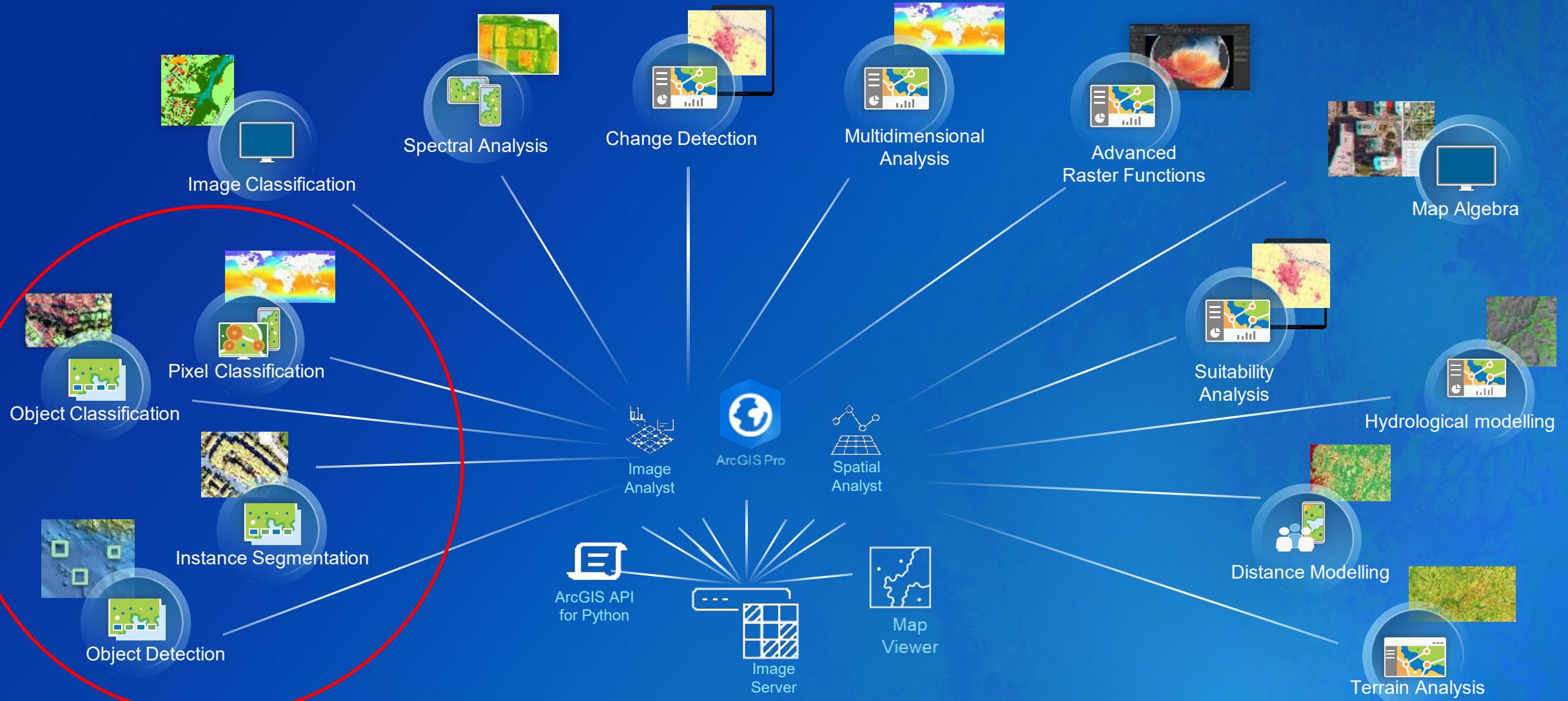
Data Science

fast.ai

Keras

# Image Analysis

## Extracting Information From Imagery



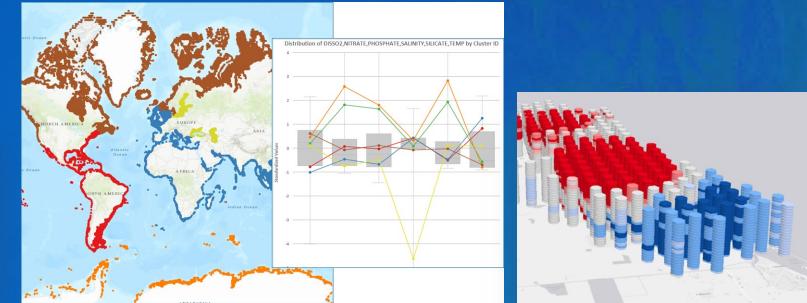
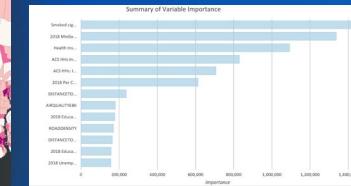
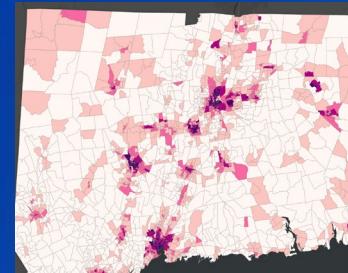
# What can GeoAI Do?



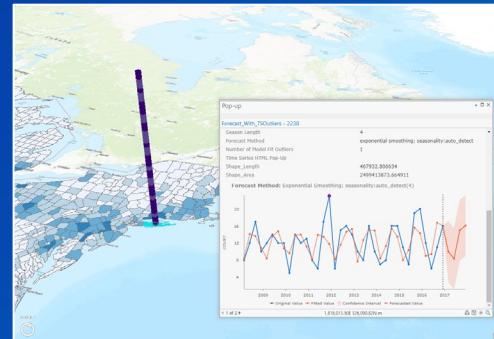
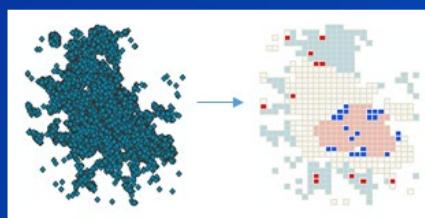
# What Can Machine Learning Do?



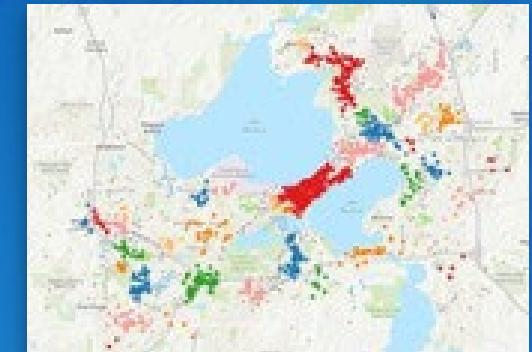
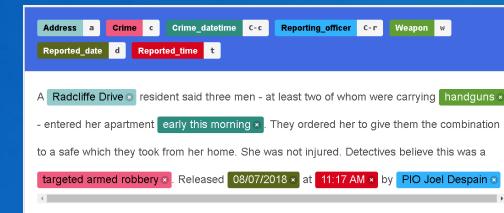
Extract features from  
Imagery & LiDAR



Find patterns & clusters



Detect anomalies



Extract insights from  
unstructured text

## Object Detection, Pixel Classification, Object Classification from Imagery

Building footprints



Roads



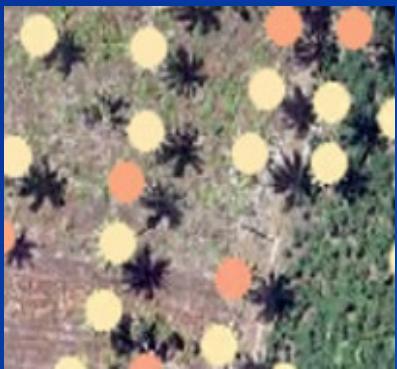
Land cover



Parcel boundaries



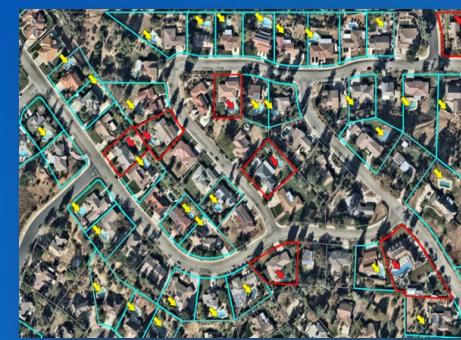
Palm trees



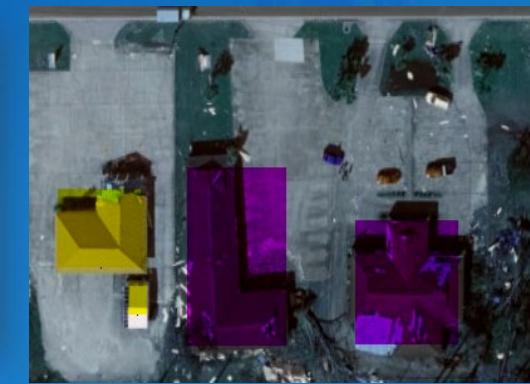
Oil pads



Swimming pools



Damaged structures



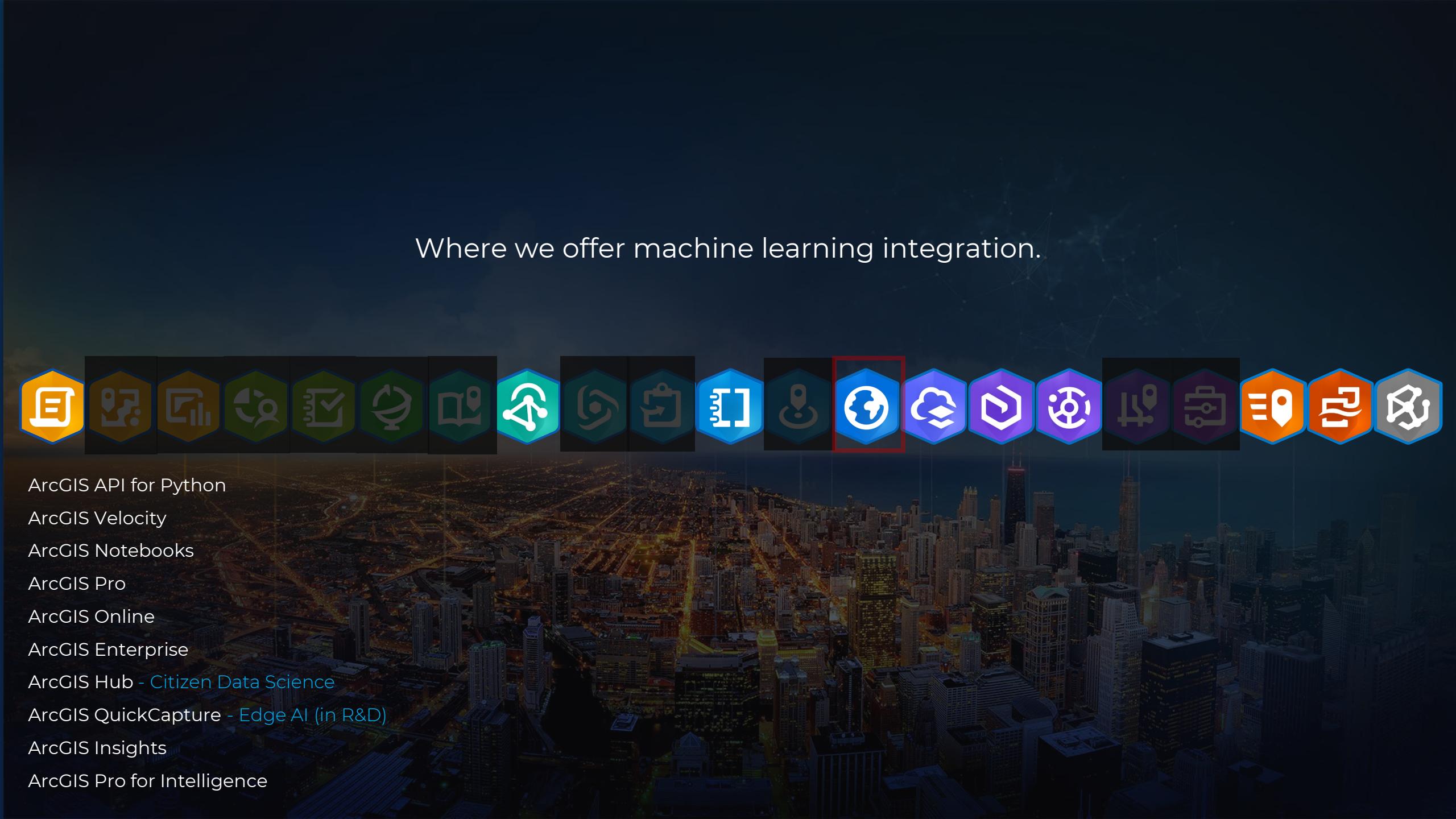
*DL will detect what you can see in the imagery .....*

Where can you apply GeoAI



AI is not one product. It spans the ArcGIS system.





Where we offer machine learning integration.



ArcGIS API for Python

ArcGIS Velocity

ArcGIS Notebooks

ArcGIS Pro

ArcGIS Online

ArcGIS Enterprise

ArcGIS Hub - Citizen Data Science

ArcGIS QuickCapture - Edge AI (in R&D)

ArcGIS Insights

ArcGIS Pro for Intelligence

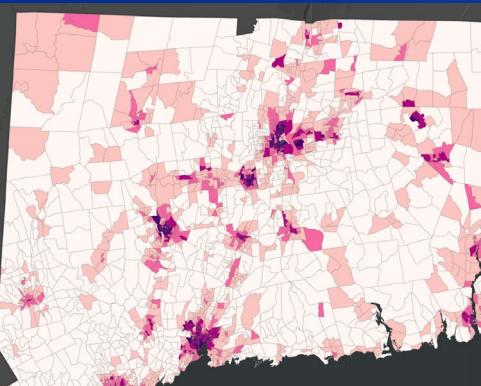
# A deeper dive into GeoAI within ArcGIS



# 1. Predictive Analytics

Prepare Data, Make Predictions, Find Correlations, Understand Top Variables, and More

Predicting asthma rates



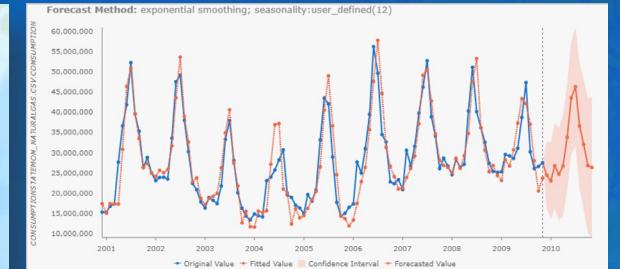
Predicting house pricing



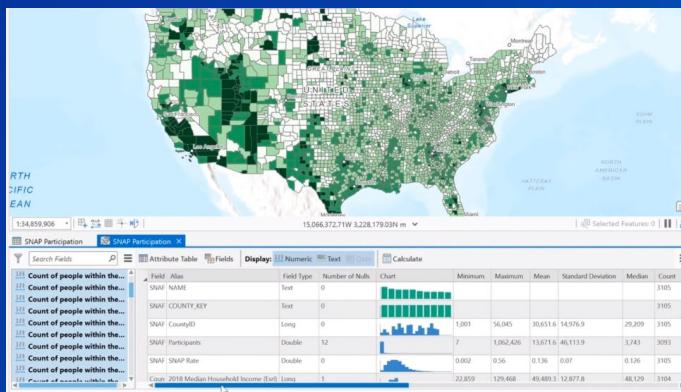
Predicting sea habitat



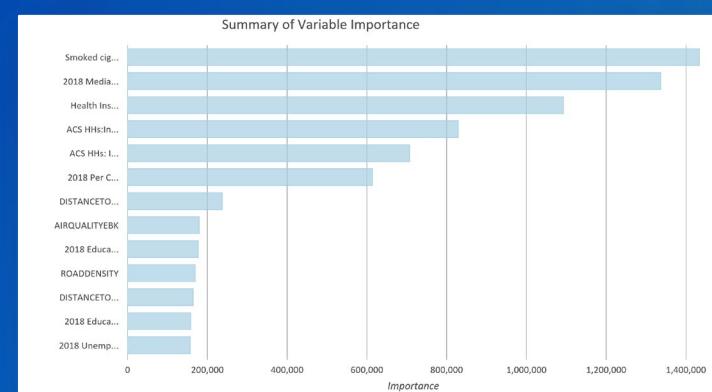
Time series forecasting



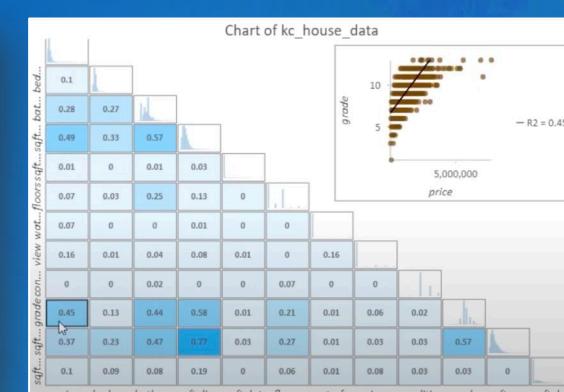
Data engineering



Variable importance

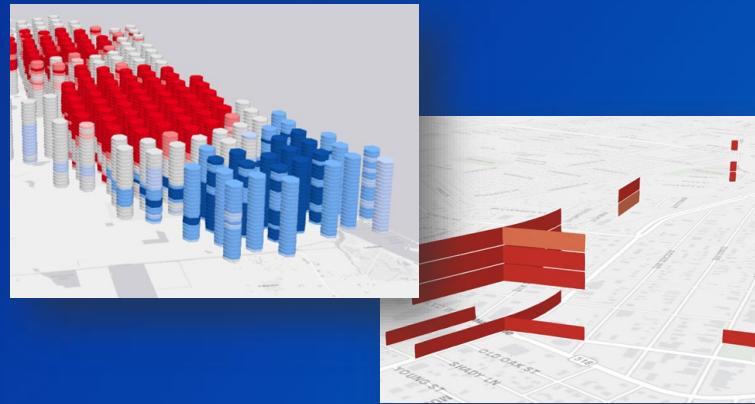
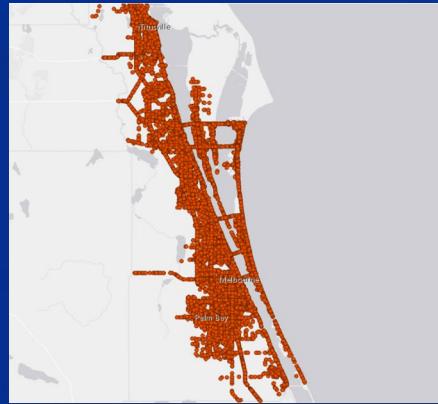


Variable correlation

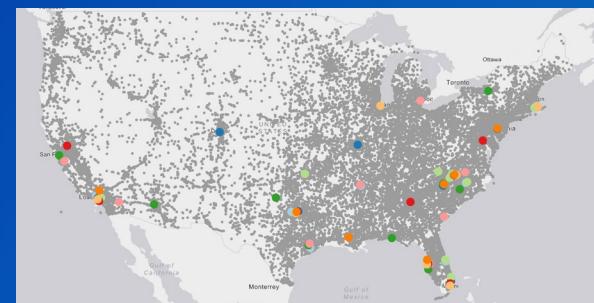
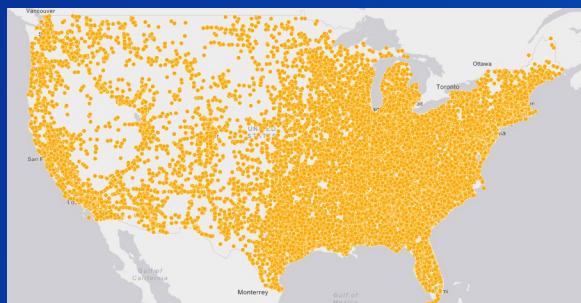
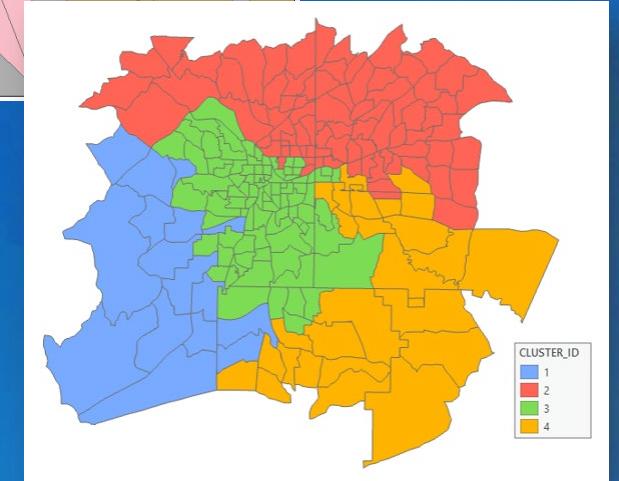
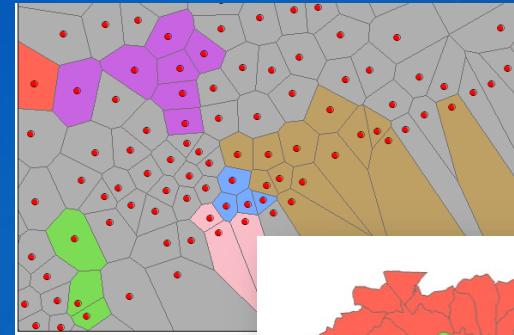


## 2. Pattern Mining and Clustering

Understand Natural Groupings in Data That Are Statistically Significant



Emerging and fading hotspots for Crashes using **SpaceTime Pattern Mining Toolbox**



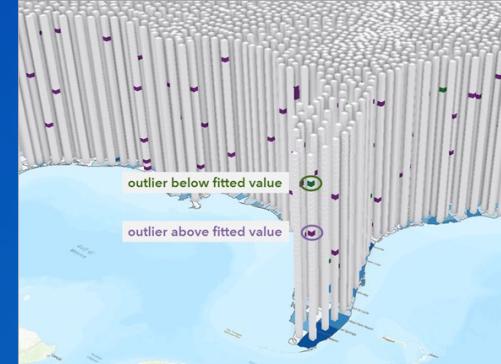
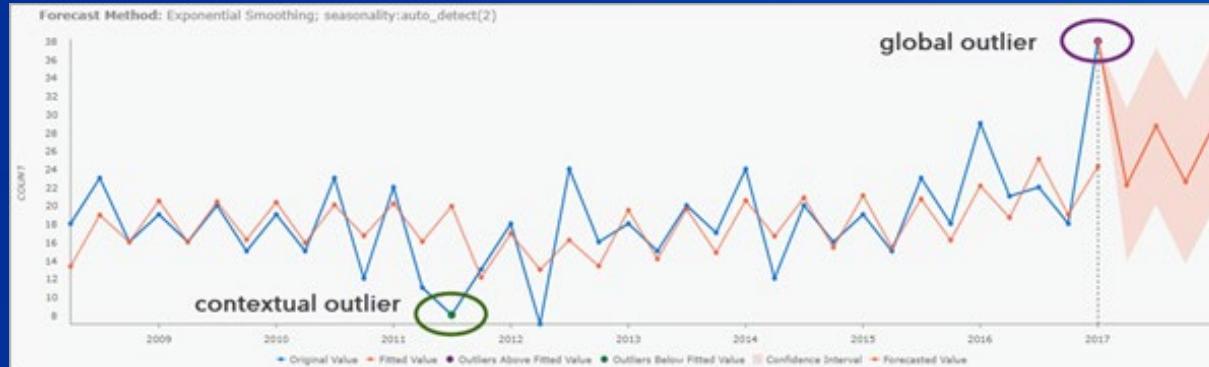
Most important fatal crashes clusters using **Density Based Clustering (DBScan)**

Find spatially contiguous clusters for animal territories using **Spatially Constrained Multivariate Clustering**

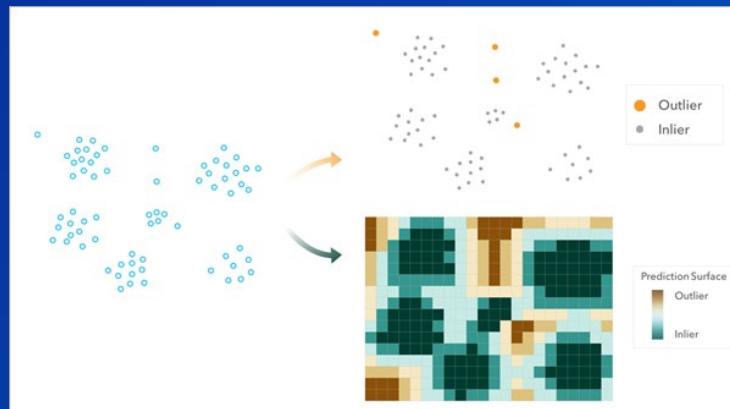
# 3. Anomaly Detection

## Spatial, Temporal, and Spatiotemporal Outliers

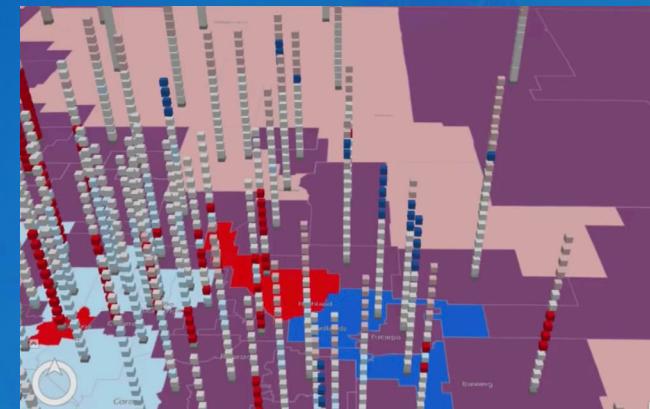
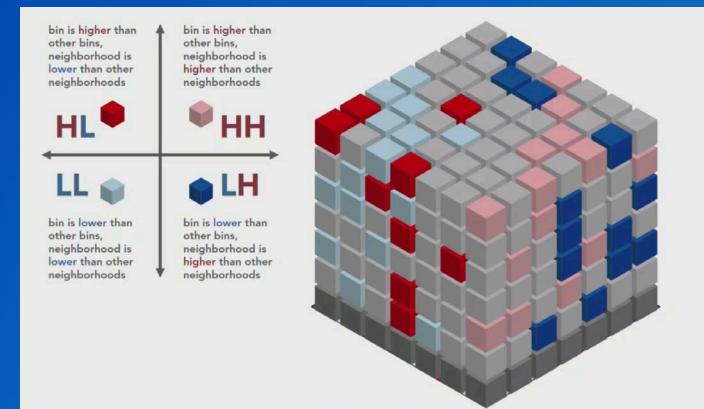
### Time series outliers (temporal)



### Spatial outliers (spatial)

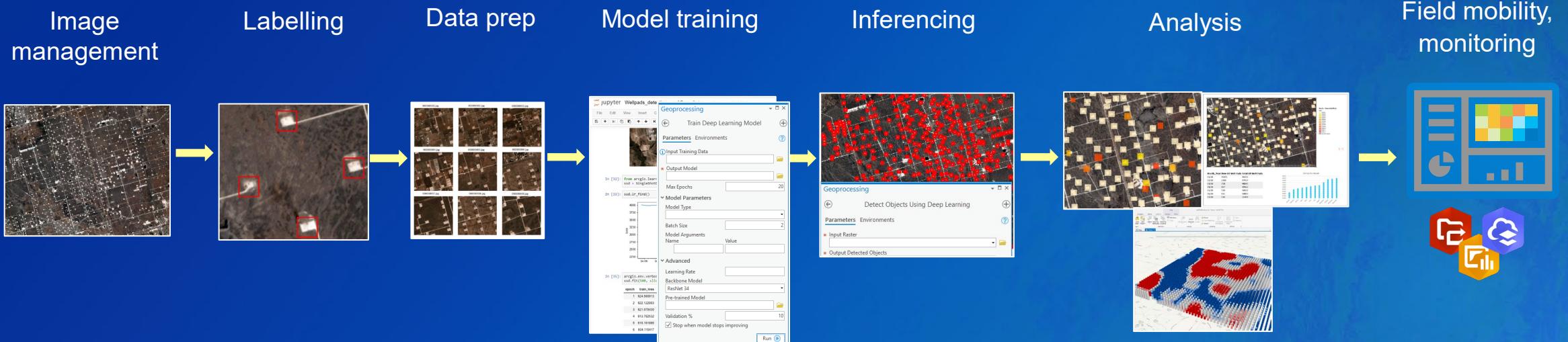


### Local outlier analysis (spatiotemporal)



# Imagery AI: End-to-End Workflow

Extract Insights from Imagery at Scale, with High Speed and Accuracy



## For Wide Range of Data Types

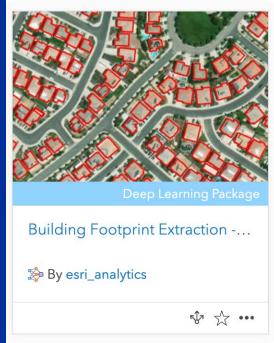
- Aerial
- Satellite
- Radar
- Lidar
- Motion imagery
- Bathymetry
- Point cloud
- Drone

## Implementing Many Tasks

- Object classification
- Object detection
- Pixel classification
- Image translation
- Object tracking
- Scanned maps

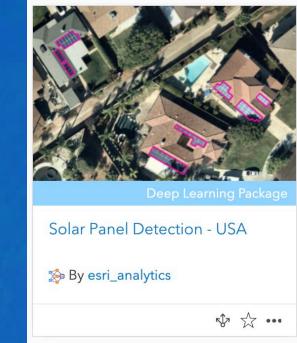
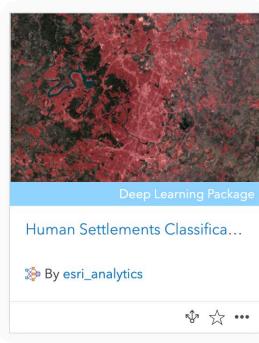
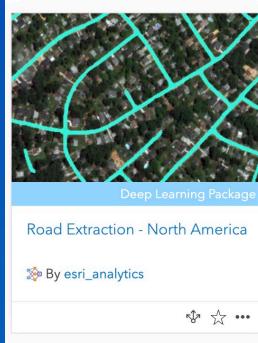
# Pre-trained Models on ArcGIS Living Atlas

Plug-and-Play Models. No Training Needed. Easy Re-training Using Local Data.



Car Detection - USA  
Deep Learning Package By esri\_analytics  
Deep learning model to detect cars in high resolution imagery.

Solar Panel Detection - USA  
Deep Learning Package By esri\_analytics  
Deep learning model to detect solar panels in high resolution imagery.



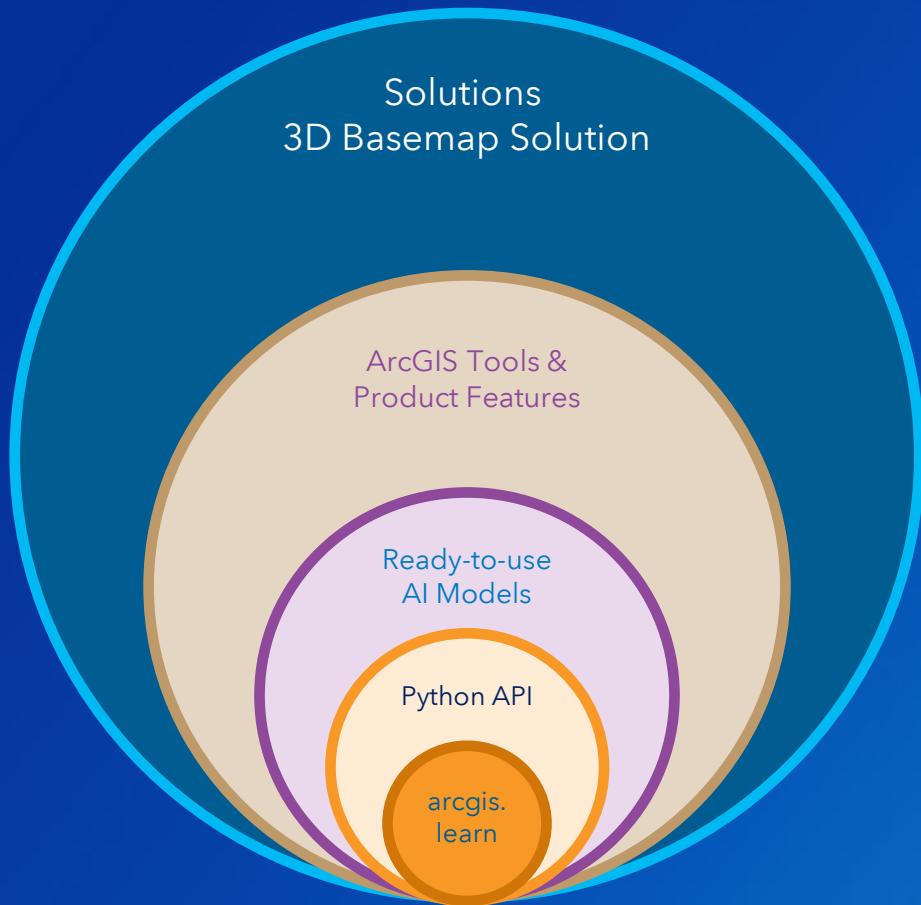
Pool Detection - USA  
Deep Learning Package By esri\_analytics  
Deep learning model to detect swimming pools in high-resolution aerial or satellite imagery.

Object Tracking  
Deep Learning Package By esri\_analytics  
Deep learning model for tracking objects in motion imagery. This model was trained using DAVIS dataset and further fine-tuned on aerial motion imagery.

## Use Models Within

- ArcGIS Pro (+ Image Analyst Extension)
- ArcGIS Enterprise (+ Image Server)
- ArcGIS Online (+ ArcGIS Image)

## 3D Basemap Solution – LOD2 Building Extraction

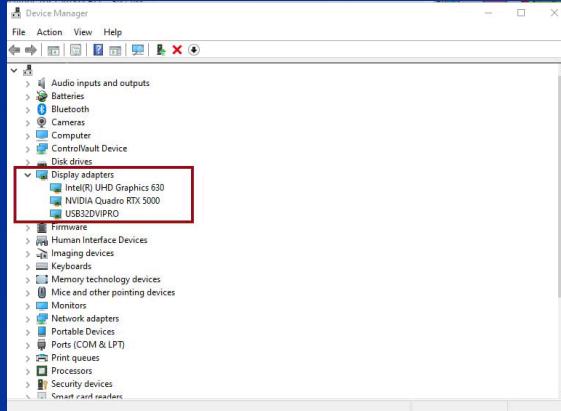


# Getting Started | Environment & Licensing



# Getting Started

## Prerequisites



GPU availability, specifications, and compatibility

A screenshot of a terminal window titled 'cmd.exe' with the path 'C:\Windows\system32\cmd.exe'. The command 'nvidia-smi' is run, showing GPU information. The output includes the NVIDIA-SMI version (451.82), driver version (451.82), and CUDA version (11.0). It lists a Quadro RTX 5000 GPU with 214MiB of memory. The 'Processes' section shows no running processes.

CUDA | nvidia-smi

## Licensing

### ArcGIS Pro Single Use License

Name	Licensed	Expires
Basic	No	N/A
Standard	No	N/A
Advanced	Yes	10/19/2022

### Esri Extensions

Name	Licensed	Expires
Business Analyst	Yes	10/19/2022
Data Interoperability	Yes	10/19/2022
Data Reviewer	Yes	10/19/2022
Defense Mapping	Yes	10/19/2022
Geostatistical Analyst	Yes	10/19/2022
Image Analyst	Yes	10/19/2022
Indoors	Yes	10/16/2022
LocateT	Yes	10/19/2022
Location Referencing	Yes	10/19/2022

ArcGIS Pro availability and version



Deep learning framework for ArcGIS Pro installation

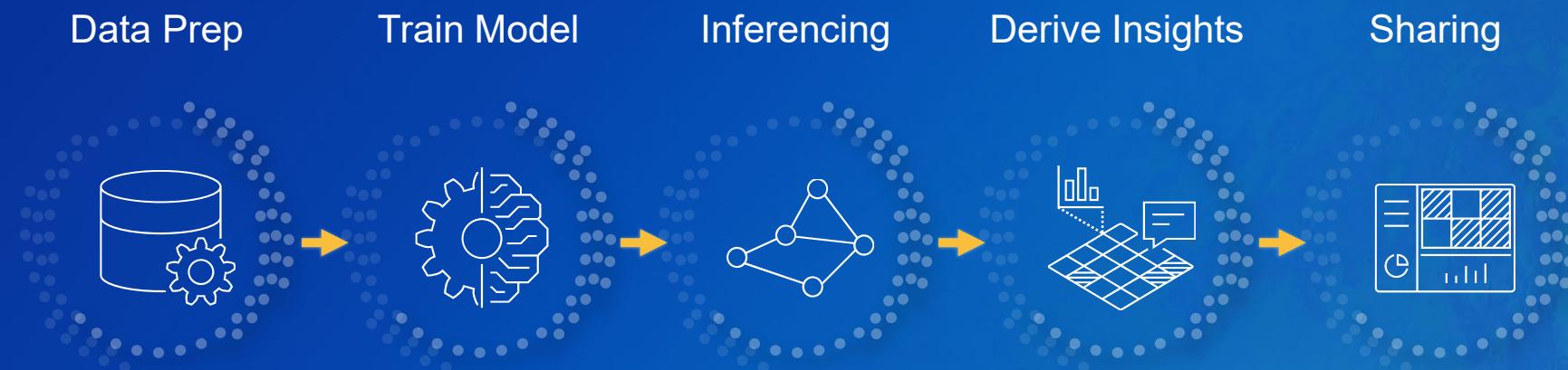
A screenshot of a Python code editor. The code imports torch, fastai, and checks if a GPU is available using torch.cuda.is\_available(). The output shows 'True', indicating successful GPU communication.

Communicating with the GPU

# GeoAI in ArcGIS Pro



# Machine Learning Lifecycle



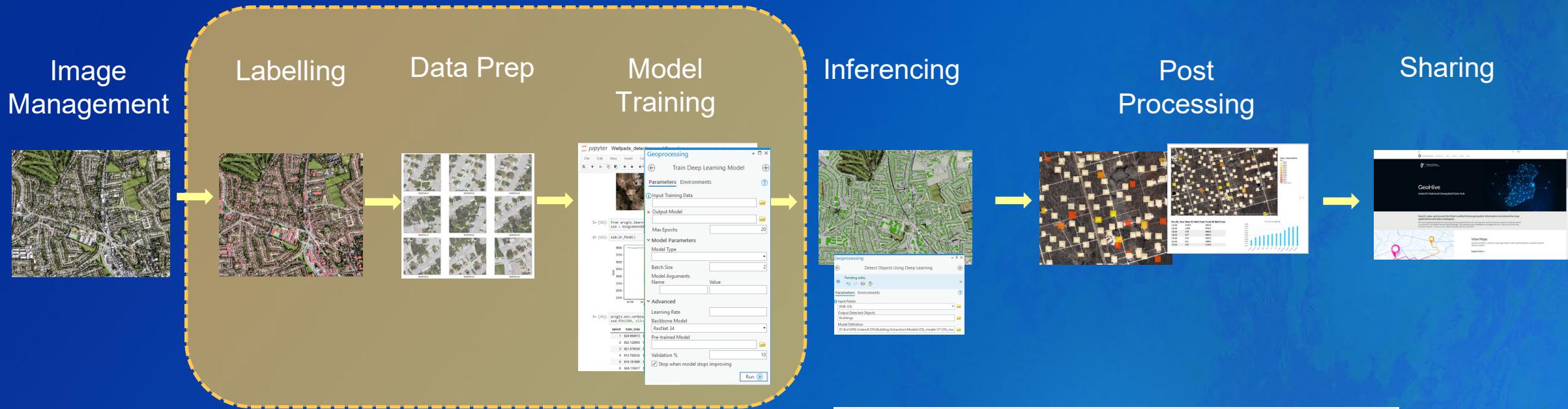
# GeoAI Tools in ArcGIS Pro

The screenshot shows the ArcGIS Pro ribbon with the **Imagery** tab selected. Below the ribbon, there are several toolbars and tool groups. The **Image Classification** group contains tools for **Classification Wizard**, **Classification Tools**, and **Image Analyst Tools**. The **Mensuration** group contains tools for **Point**, **Distance**, and **Area**. The **Tools** group contains **Results**, **Process**, **Indices**, **Pixel Editor**, **Image Information**, **Create Hosted Imagery**, **Video Search**, and **Motion Imagery**. A sidebar on the right lists the following GeoAI tools:

- Classification and Pattern Recognition**
  - Classify Raster
  - Compute Confusion Matrix
  - Compute Segment Attributes
  - Create Accuracy Assessment Points
  - Generate Training Samples From Seed Points
  - Inspect Training Samples
  - Linear Spectral Unmixing
  - Predict Using Regression Model
  - Remove Raster Segment Tiling Artifacts
  - Segment Mean Shift
  - Train ISO Cluster Classifier
  - Train K-Nearest Neighbor Classifier
  - Train Maximum Likelihood Classifier
  - Train Random Trees Classifier
  - Train Random Trees Regression Model
  - Train Support Vector Machine Classifier
  - Update Accuracy Assessment Points
- Image Analyst Tools**
  - Change Detection
  - Classification and Pattern Recognition
  - Deep Learning
  - Extraction
  - Map Algebra
  - Math
  - Motion Imagery
  - Multidimensional Analysis
  - Overlay
  - Statistical
- Deep Learning**
  - Classify Objects Using Deep Learning
  - Classify Pixels Using Deep Learning
  - Compute Accuracy For Object Detection
  - Detect Change Using Deep Learning
  - Detect Objects Using Deep Learning
  - Export Training Data For Deep Learning
  - Non Maximum Suppression
  - Train Deep Learning Model
- Change Detection**
  - Analyze Changes Using CCDC
  - Analyze Changes Using LandTrendr
  - Compute Change Raster
  - Detect Change Using Change Analysis Raster

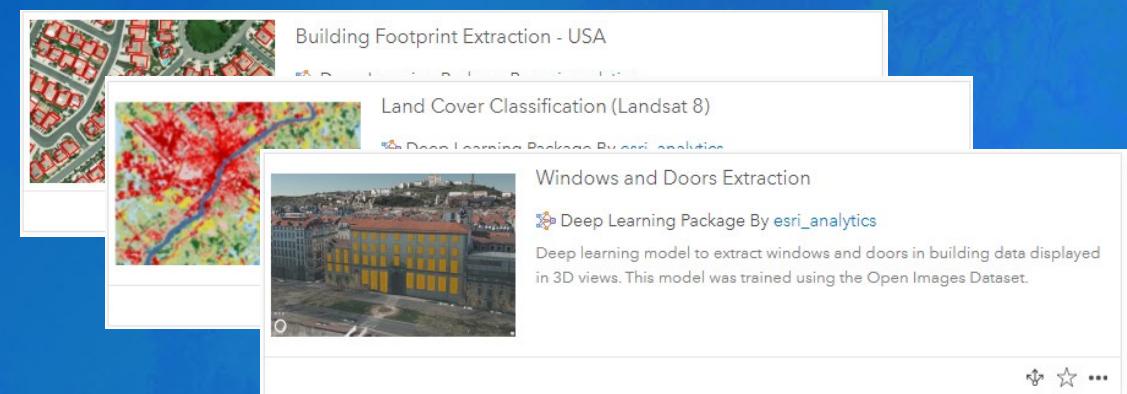
# Deep Learning Workflow

## Pre-trained Models



### Eliminates:

- Imagery requirements for model training
- Labelling requirements
- Training AI models
- Massive compute requirements

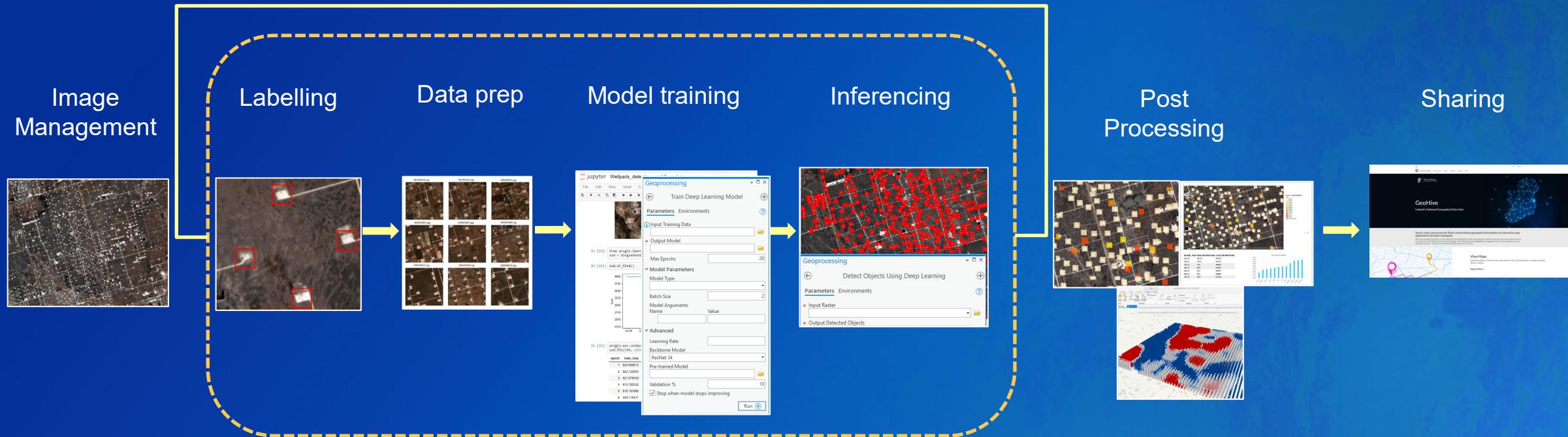


# Applying DL with Out of the Box Model

Building Detection



# Fine-Tuning Mask RCNN Building Detection Model.



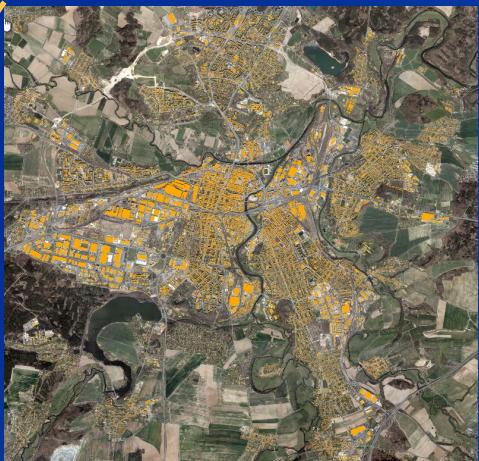
## For Wide Range of Data Types

- Aerial
- Satellite
- Radar
- Lidar
- Motion imagery
- Bathymetry
- Point cloud
- Drone

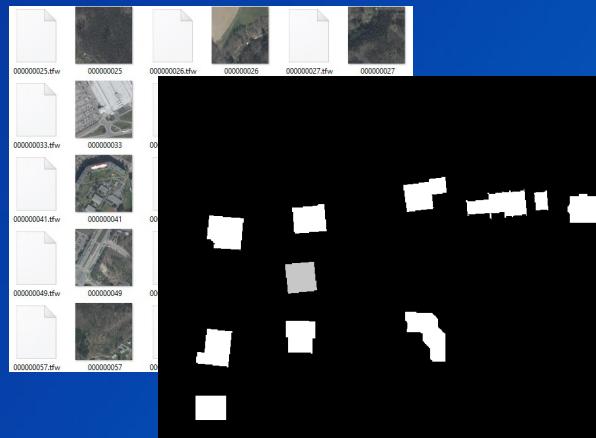
## Implementing Many Tasks

- Object classification
- Object detection
- Pixel classification
- Image translation
- Object tracking
- Scanned maps

# Fine-Tuning Mask RCNN Building Detection Model.



Label with existing buildings



```
In [9]: with arcpy.EnvManager(scratchWorkspace=temp_gdb, processorType="GPU", workspace=temp_gdb):
    TrainDeepLearningModel(training_data,model_output,100, "MASKRCNN", 4, "chip_size 512",
                           None, "RESNET50", pretrained_dlpk, 10, "STOP_TRAINING", "FREEZE_MODEL")

epoch train_loss valid_loss Time
0 1.194625 1.050457 11:58
1 1.099954 1.051591 12:03
2 1.028073 1.040456 12:02
3 1.001396 1.022784 12:03
4 0.987169 1.000456 11:57
5 0.976907 0.985356 11:12
6 0.987169 0.993153 11:08
7 0.968429 0.998451 11:08
8 0.953337 0.986956 11:11
9 0.943848 0.967826 11:08
10 0.953337 0.966999 11:09
11 0.931865 0.966357 11:09
12 0.920803 0.965357 11:10
13 0.899677 0.951855 11:08
14 0.920803 0.963767 11:09
15 0.913574 0.948525 11:14
16 0.915749 0.948235 11:11
17 0.846629 0.948156 11:09
18 0.877141 0.911051 11:10
19 0.880115 0.927745 11:09
20 0.874127 0.938844 11:10
21 0.875783 0.923362 11:10
22 0.880115 0.927745 11:11
23 0.875783 0.923362 11:10
24 0.953438 0.929121 11:09
25 0.844735 0.924087 11:12
26 0.888156 0.924411 11:09
27 0.888156 0.924411 11:10
28 0.838589 0.930189 11:16
29 0.827601 0.929988 11:09

Epoch 29: early stopping
Computing model metrics...
```

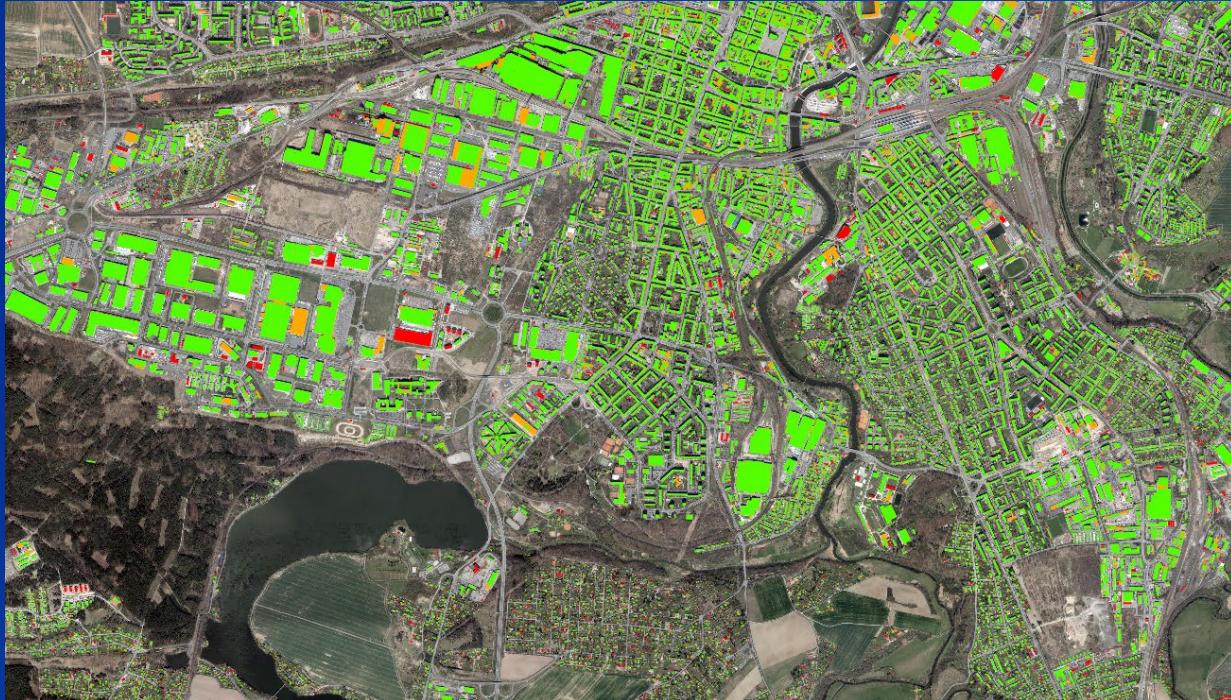
Model training (fine-tuning)



Inference with fine-tuned model (71500 features)

# Fine-Tuning the Deep Learning Package

Building Detection

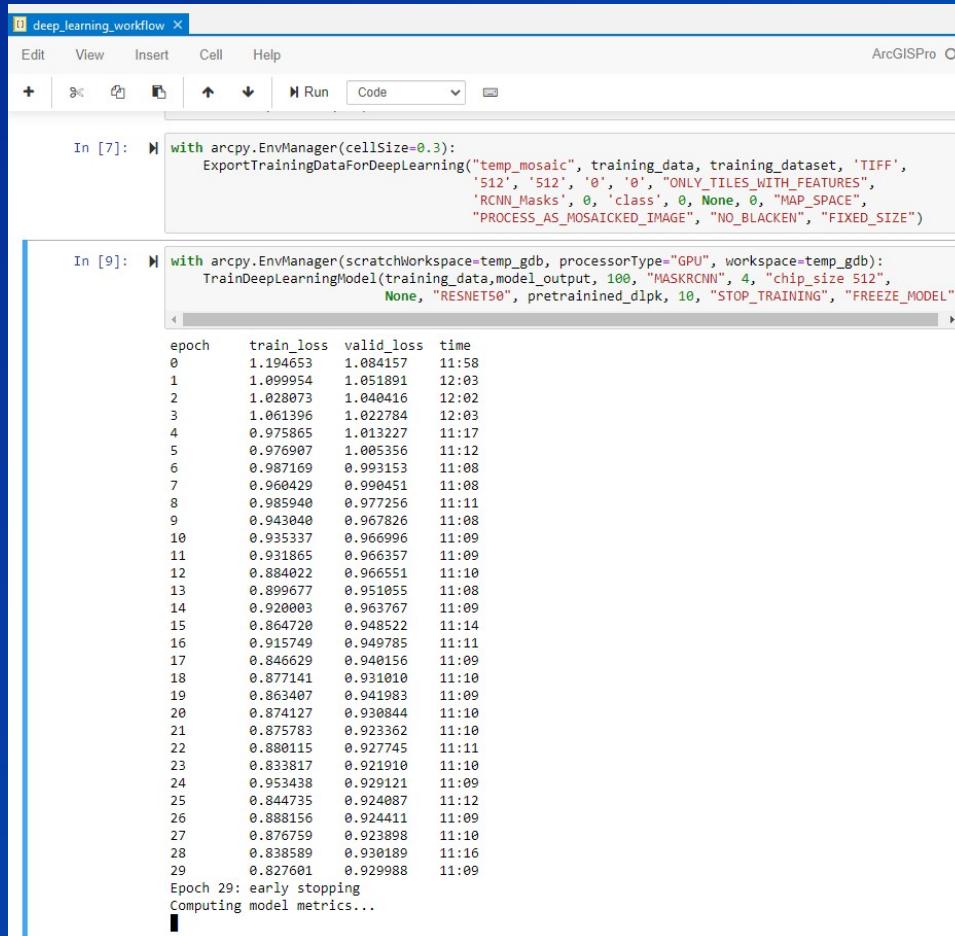


# Advanced GeoAI Workflows with Python



# Automation end to end workflow with ArcGIS Pro Notebooks.

- Utilize ArcPy & Python API to automate process



The screenshot shows an ArcGIS Pro Notebook titled "deep\_learning\_workflow". It contains three code cells:

- In [7]:** A code block using ArcPy's EnvManager to export training data for deep learning. The code uses parameters like cellSize=0.3, temp\_mosaic, training\_data, training\_dataset, and various processing options.
- In [9]:** A code block using ArcPy's EnvManager to train a DeepLearningModel. It specifies a scratch workspace, processorType="GPU", workspace="temp\_gdb", and various training parameters. The cell displays a table of training metrics:

epoch	train_loss	valid_loss	time
0	1.194653	1.084157	11:58
1	1.099954	1.051891	12:03
2	1.028073	1.040416	12:02
3	1.061396	1.022784	12:03
4	0.975865	1.013227	11:17
5	0.976907	1.005356	11:12
6	0.987169	0.993153	11:08
7	0.960429	0.990451	11:08
8	0.985940	0.977256	11:11
9	0.943040	0.967826	11:08
10	0.935337	0.966996	11:09
11	0.931865	0.966357	11:09
12	0.884022	0.966551	11:10
13	0.899677	0.951055	11:08
14	0.920003	0.963767	11:09
15	0.864720	0.948522	11:14
16	0.915749	0.949785	11:11
17	0.846629	0.940156	11:09
18	0.877141	0.931010	11:10
19	0.863407	0.941983	11:09
20	0.874127	0.930844	11:10
21	0.875783	0.923362	11:10
22	0.880115	0.927745	11:11
23	0.833817	0.921910	11:10
24	0.953438	0.929121	11:09
25	0.844735	0.924087	11:12
26	0.888156	0.924411	11:09
27	0.876759	0.923898	11:10
28	0.838589	0.930189	11:16
29	0.827601	0.929988	11:09

Epoch 29: early stopping  
Computing model metrics...

**In [2]:** A code block defining paths for images, pretrained DL PK, model parameters, and training data. It also defines a function to create a GDB directory and execute a CreateFileGDB script.

```
images_directory = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\True_Ortho"
pretrained_dlpk = r"C:\Users\ramialouta\Documents\ArcGIS\Packages\use_building_footprints.dlpk"
model_parameters = ["padding 64;batch_size 16;threshold 0.5;return_bboxes False;tile_size 512",
                     "padding 64;batch_size 4;threshold 0.5;return_bboxes False;tile_size 1024",
                     "padding 128;batch_size 16;threshold 0.5;return_bboxes False;tile_size 512",
                     "padding 128;batch_size 4;threshold 0.5;return_bboxes False;tile_size 1024"]
training_data = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\trainings\buildings_training_data_v2"
footprints_fc = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\Footprints\SURE_Pilsen_building_footprints.gdb\Pilsen_building_footprints_v2"
model_output = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\models\retrained_buildings_v2"
gdb_dir = images_directory.split("\\")[:-1]
gdb = f'{gdb_dir[-1]}.temp.gdb'
final_gdb = f'{gdb_dir[-1]}.gdb'
gdb_dir[0] = f'{gdb_dir[0]}\\"'
gdb_dir = functools.reduce(os.path.join, gdb_dir)
training_dataset = "training_dataset"

# Execute CreateFileGDB
temp_gdb = os.path.join(gdb_dir, gdb)
final_gdb = os.path.join(gdb_dir, "deeplearning_czech.gdb")

if arcpy.Exists(temp_gdb):
    featureclasses = arcpy.ListFeatureClasses()
    fsc = []
    for fc in featureclasses:
        fc=fc.replace("_", "-")
        fsc.append(fc)
else:
    arcpy.CreateFileGDB_management(gdb_dir, gdb)
    arcpy.env.workspace = temp_gdb
    arcpy.env.overwriteOutput = True

output_detected_buildings_fc = os.path.join(temp_gdb, "detected_buildings")
raw_buildings = os.path.join(final_gdb, "raw_detected_buildings")
arcpy.env.overwriteOutput = True
```

**In [4]:** A code block that iterates through model parameters to create a DL PK, set up a workspace, and use DetectObjectsUsingDeepLearning to process images. It also prints processing time.

```
dlpk_name_extesion = model_output.split("\\")[-1]
dlpk= os.path.join(model_output, f'{dlpk_name_extesion}.dlpk')
start_datetime = datetime.now()
detected_outputs = []
for parameter in model_parameters:
    iter_startdt = datetime.now()
    padding = parameter.split(";")[0].split(" ")[-1]
    btachsize = parameter.split(";")[1].split(" ")[-1]
    threshold = str(int(float(parameter.split(";")[2].split(" ")[-1])*100))
    tilesize = parameter.split(";")[1].split(" ")[-1]
    output_fc = f"czech_buildings_30m_{padding}_{btachsize}_Threshold(threshold)_Tile(tilesize)_{dlpk_name_extesion}"
    with arcpy.EnvManager(processorType="GPU", cellsize=0.3, workspace=final_gdb):
        arcpy.ia.DetectObjectsUsingDeepLearning("temp_mosaic", output_fc, dlpk,
                                               parameter, "NO_NMS", "Confidence",
                                               "Class", 0, "PROCESS_ITEMS_SEPARATELY")

    torch.cuda.empty_cache()
    detected_outputs.append(output_fc)
    iter_enddt = datetime.now()
    iter_minutes_diff = (iter_enddt - iter_startdt).total_seconds() / 60.0
    print(f"The processing of image: {output_fc} took {int(iter_minutes_diff)} minutes")
```

# Deep Learning with Python

Working Environment:

In [4]: `dl_workflow_env = dlw.workflow_env()  
dl_workflow_env`

Where to run deep learning workflows: Use ArcPy libraries to train, fine-tune & inf ▾

In [5]: `widgets_box = dlw.data_inputs()  
widgets_box[0]`

Choose from the below workflows:  Fine-tune Living Atlas Model  
 Fine-tune Local Model  
 Train new model  
 Inference with local model  
 Inference with Living Atlas Model

Imagery layer in ArcGIS Pro TOC `berlin_imagery`

Labelled data in ArcGIS Pro TOC `Berlin`

ArcGIS Pro project directory: `E:\Esri\Useful_Resources\Deep_Learning\Deep Learning to Webmap`

# Deep Learning Resources

Deep Learning with ArcGIS Pro Tips & Tricks:  
Part 1



Scan Me

Deep Learning with ArcGIS Pro Part 3: QA/QC  
Extracted Features



Scan Me

Deep Learning with ArcGIS Pro Tips & Tricks:  
Part 2



Scan Me

Introducing pre-trained geospatial deep  
learning models



Scan Me

Deep Learning with ArcGIS Pro Part 3: QA/QC  
Extracted Features



Scan Me

Pre-trained deep learning models  
update (February 2021)



Scan Me

# Thank you to our sponsors

Co-host



esri®

THE  
SCIENCE  
OF  
WHERE®

