# Machine learning models for the Kadoma-Chegutu greenstone belt using magnetics, radiometrics and mineral occurrences (minoccs) data


***Tenyears Gumede***

**KNOWLEDGE FACTORY**

tenyearsgumede@gmail.com/tenyears@oneiricminerals.com


**1September 2023**

# The Presentation

**Introduction**

Key Concepts

Models

Kadoma – Chegutu data and Models

Conclusions

# Introduction

There is an ongoing digital revolution motivated by the data-driven scientific discovery paradigm.

Machine learning and artificial intelligence (AI) represent two of the trendiest and important topics in geosciences right now.

**Machine learning in the interpreter's toolbox: Unsupervised, supervised, and deep-learning applications**

**Artificial Intelligence:**" It is the science and engineering of making intelligent machines, especially intelligent computer programs." (John McCarthy)

**Machine Learning:** Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

**Supervised Learning:** the term "supervised" refers to a set of samples where both the desired output signals (label) and the predictive variables are already known.

**Unsupervised Learning:** The algorithm has no prior labeled data to learn from i.e., it learns from unlabeled data by finding patterns among examples and grouping them accordingly. The algorithm is provided with a large volume of data and expected to identify hidden patterns

**Training Data:** The training step is used to choose the best-suited algorithm and combination of parameters for the classification problem

**Validaton Data:**

# AI In Mineral Exploration

Advances in enormous computing power to effectively process and analyse massive amounts of data (Big Data)

Access to huge datasets (Big Data), making it difficult for Knowledge Driven Model

- *Big data is a set of data that cannot be managed , processed, or analysed with traditional software/algorithms within reasonable amount of time*
- *It revolves around **Volume, Velocity, Variety, Value , Veracity***

*( Walmart handles over one million purchase transactions per hour )*
*( Facebook processes more than 250 million picture uploads per day)*

Human bias from Interpretation

Find complex relationships between different datasets

Automation of tasks

Parallel computing, and cloud computing coupled with better computer hardware

Building complex geological models based on data

Dwindling rates of discovery in exploration in certain minerals

# Supervised ML

Data is presented as plan maps – 2D Model cells

Cell size should represent well the data at scale of surveys while maintain the problem of reasonable size pertaining computational time

Labels are requiredfor the model to learn by associating them with the input data

In this problem, labels were generated using known mineral occurences/ existing mines

# Training and Validation

Labeled data is portioned into training and validation datasets

The distribution of training and validation datasets are nearly equal

Model learn on training data and can be tested against a similar set of data

# Machine Learning Models

# Model Metrics

Traditional ML Classificaton models metrics are Precision, Recall and Accuracy

Accuracy

Precision

Recall

# PYTHON LIBRARIES

## import key packages

```
import os
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import geopandas as gpd
import rasterio
import rasterio.mask
from rasterio.features import rasterize
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import metrics
from sklearn.metrics import roc_curve, auc
from imblearn.under_sampling import RandomUnderSampler
```

1 Load and inspect data sets

 - mineral occurrence point data -**geopandas**

- magnetic and radiometric data sets **– rasterio**

2 Combine data sets to build a labeled N_pixel, N_layers array for model training

 - inspect differences between proximal vs. distal to mineralisation pixels

3 Train using a **random forest classifier** and apply to all pixels, visualise results

4 evaluate performance with a randomly selected testing subset

 - repeat with stratified classes

5 Used a **checkerboard** data selection procedure, train and evaluate models

6 Investigate occurrence holdout models with a spatially clustered approach

# Kadoma-Chegutu Greenstone Belts Datatsets

Geology data – Two ZGS Bulletins Cover the Area of Study (B64 and B34)

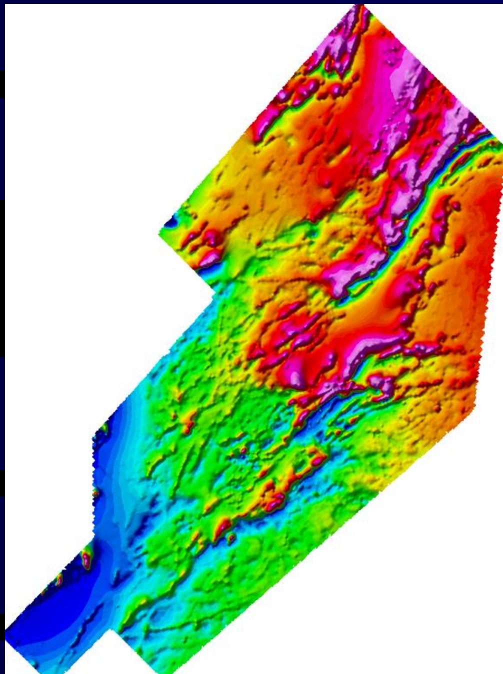
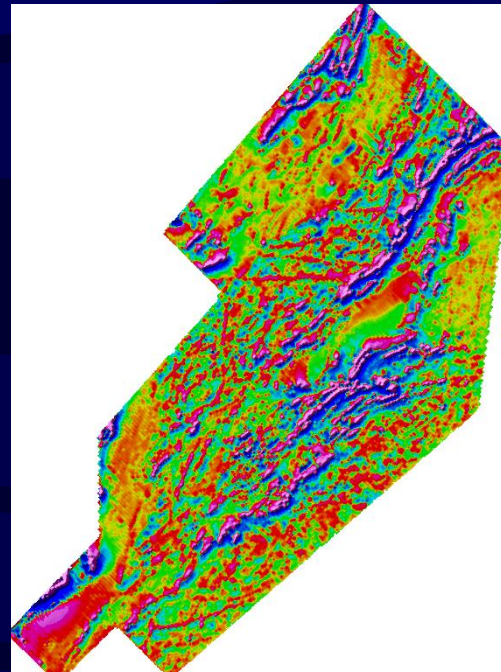
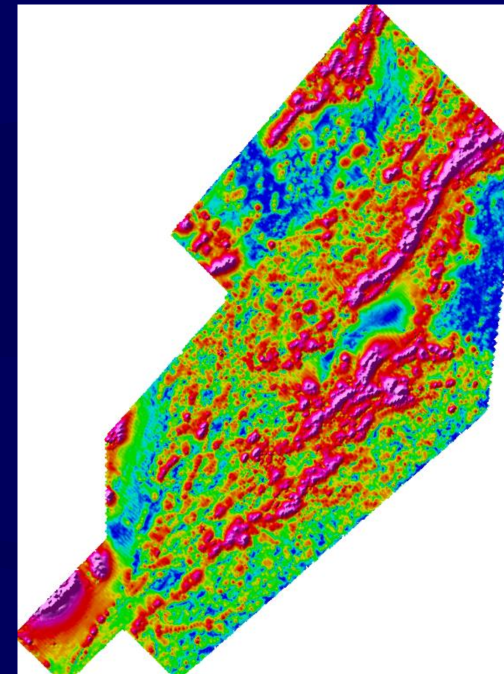
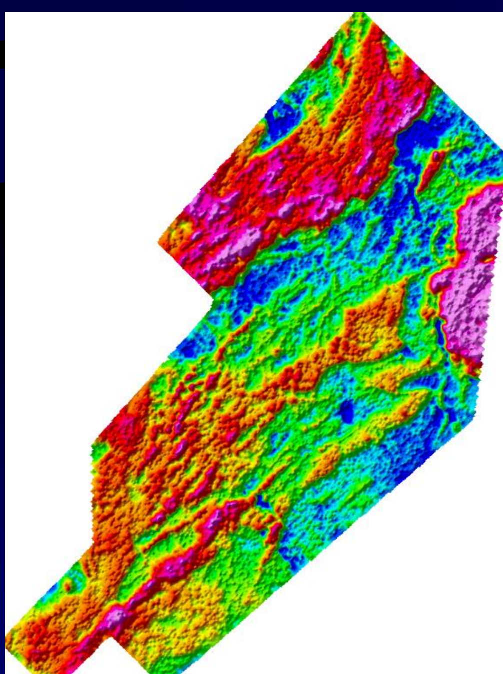
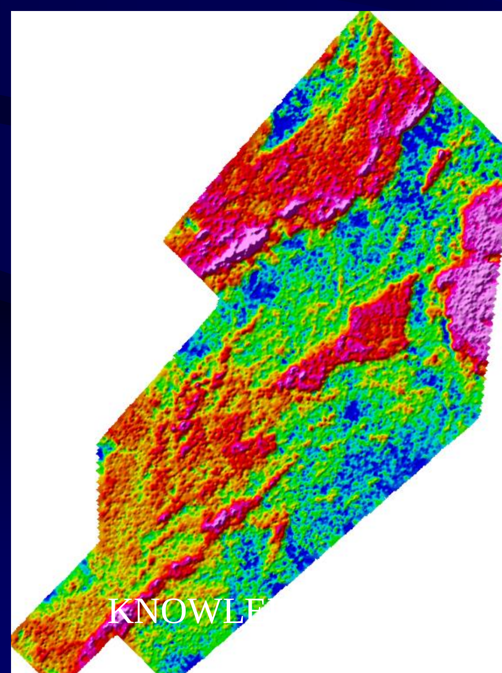
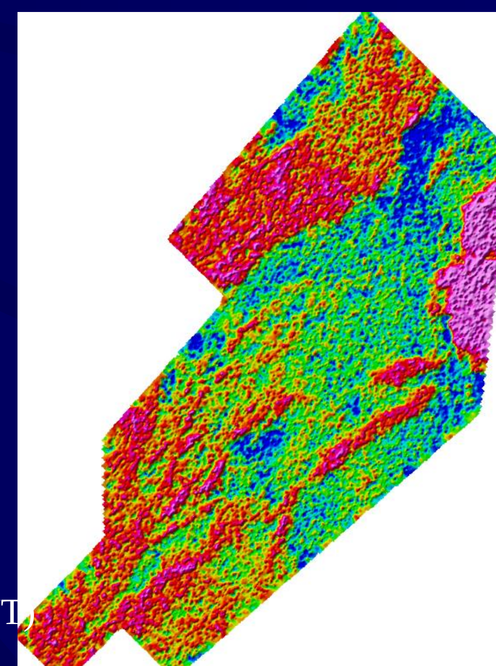
Geophysical Data - Airborne Magnetics that yielded the following deliverables:

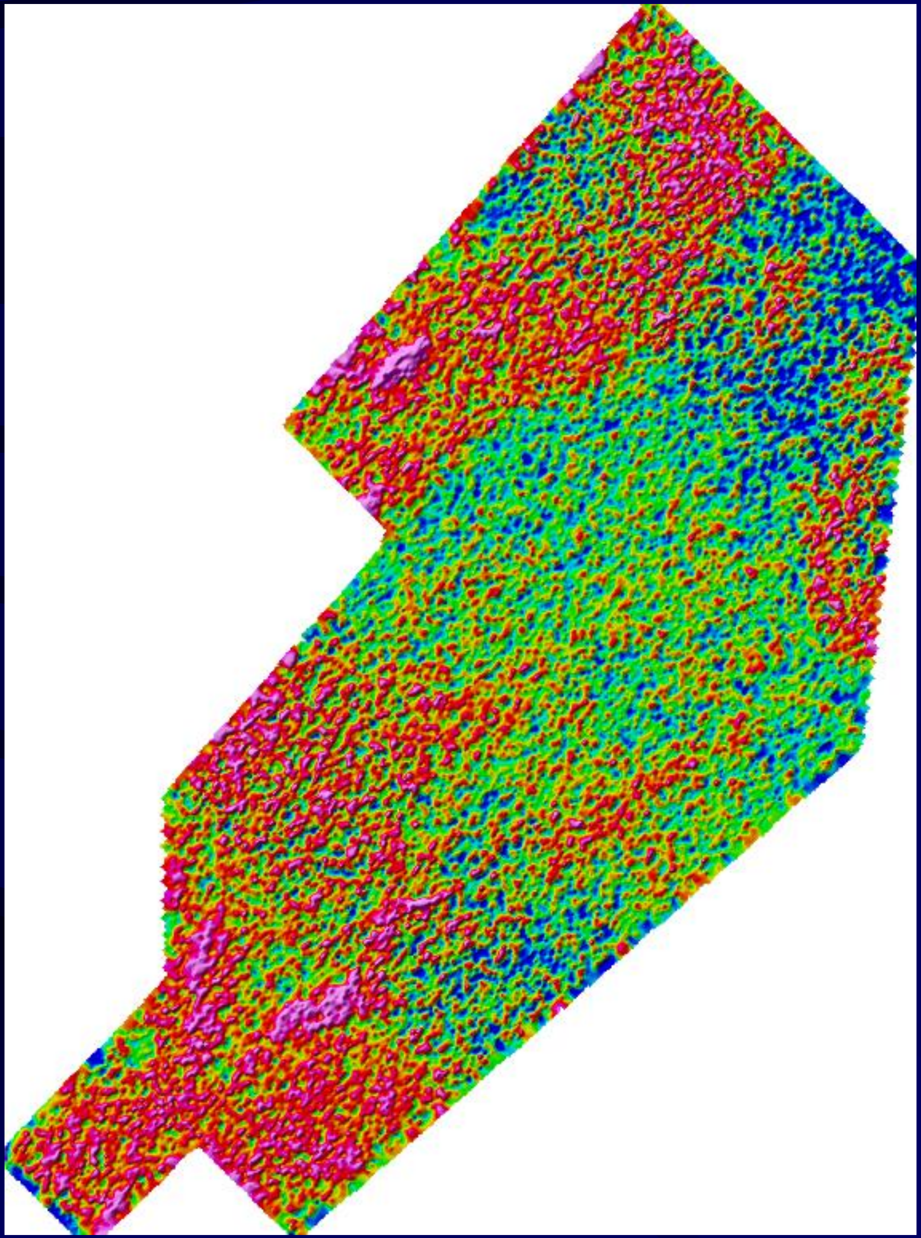
- ✓ Total Magnetic Intensity (TMI)
- ✓ Vertical Derivative (VD)
- ✓ Analytical Signal (AS)
- ✓ Total Count Radiometrics (TC)
- ✓ Potassium Count (KC)
- ✓ Thorium Count (ThC)
- ✓ Uranium Count (UC)

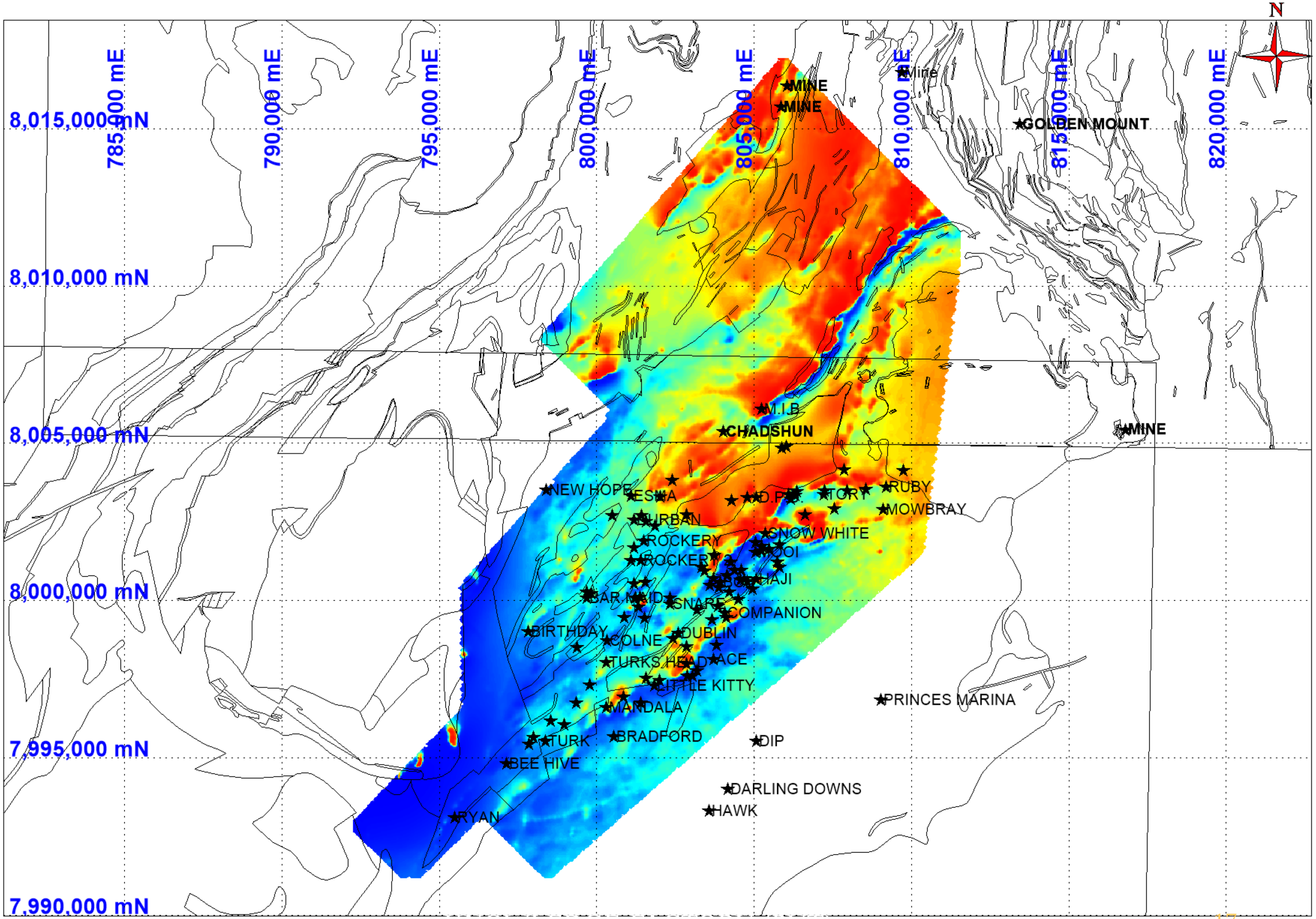
Mineral Occurrence (Minoccs) Data from B64 and B34

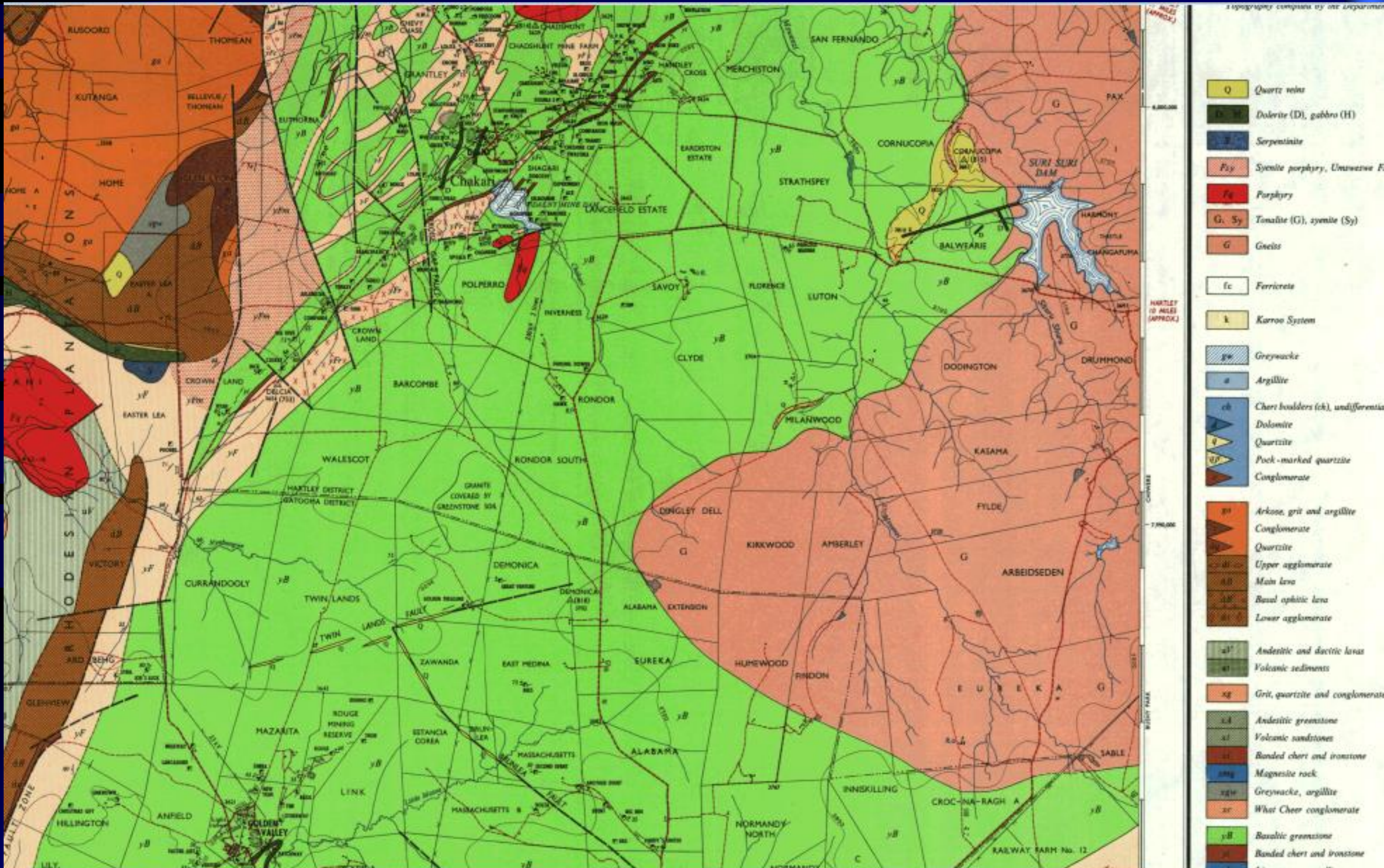
Other Probable Datasets that are available but at a course Spacing (Differing Resolutions)

- ✓ Gravity Bouger Maps
- ✓ ? Stream sediment Data (ZGS)

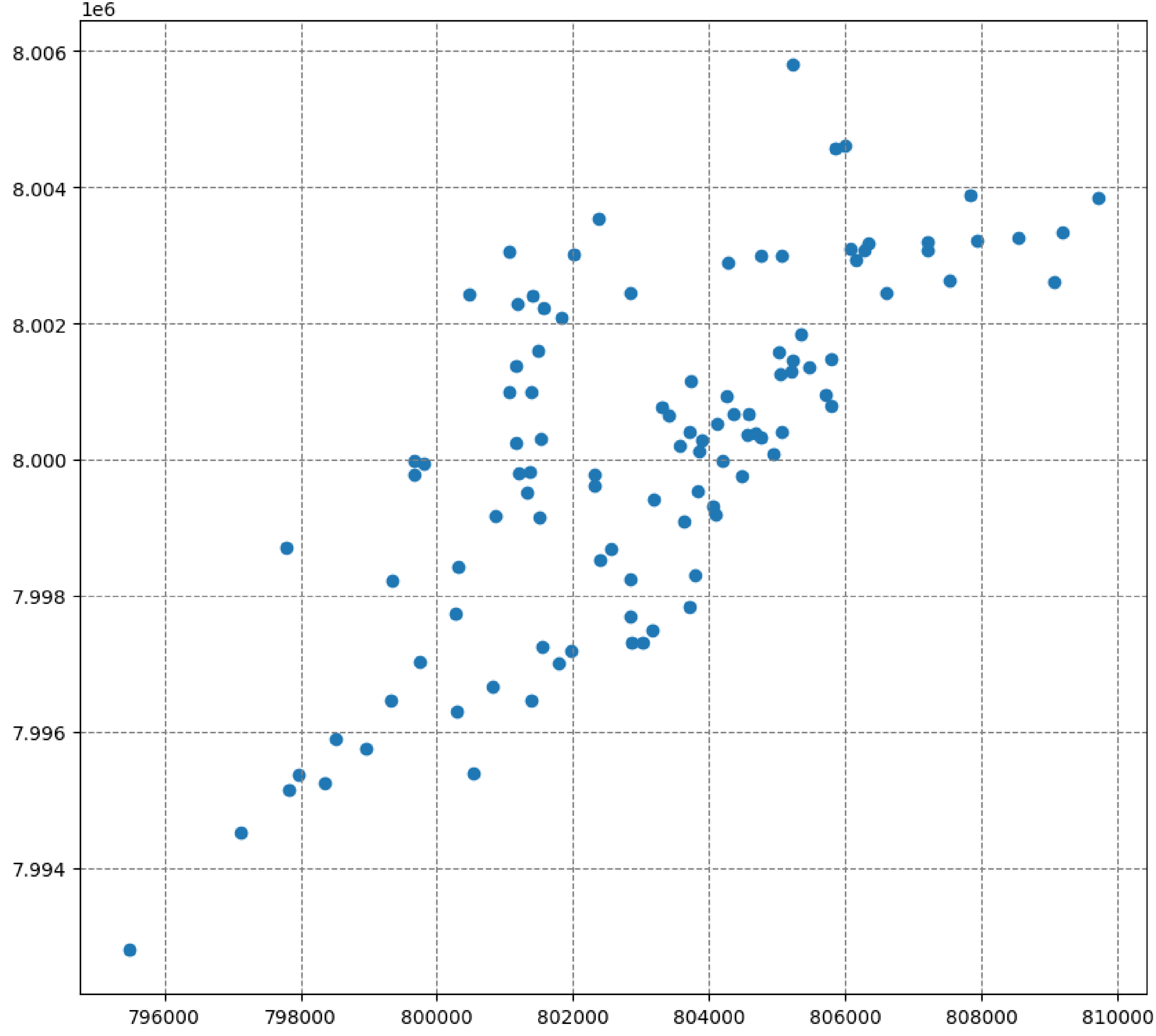










N
785,000 mE
790,000 mE
795,000 mE
800,000 mE
805,000 mE
810,000 mE
815,000 mE
820,000 mE
8,015,000 mN
8,010,000 mN
8,005,000 mN
8,000,000 mN
7,995,000 mN
7,990,000 mN
MINE
MINE
Mine
GOLDEN MOUNT
M.I.B
CHADSHUN
MINE
NEW HOPE
ESMA
D.P.
TORY
RUBY
MOWBRAY
URBAN
SNOW WHITE
ROCKERY
ROCKERY
HAJI
BAR MAID
SNARE
COMPANION
BIRTHDAY
COLNE
DUBLIN
TURKS HEAD
ACE
LITTLE KITTY
MANDALA
PRINCES MARINA
TURK
BRADFORD
DIP
BEE HIVE
DARLING DOWNS
HAWK
RYAN



RUSOORD
THOMEAN
KUTANGA
BELLEVUE/ THOMEAN
HOME
EASTER LEA
CHEVY CHASE
GRANTLEY
CHADSHUNT MINE FARM
HANDLEY CROSS
MERCHISTON
SAN FERNANDO
PAX
CORNUCOPIA
EARDISTON ESTATE
STRATHSPEY
Chakari
LANCEFIELD ESTATE
SURI SURI DAM
HARMONY
BALWEARIE
CHANGAFUMA
POLPERRO
SAVOY
FLORENCE
LUTON
INVERNESS
CROWN LAND
CLYDE
DODINGTON
DRUMMOND
BARCOMBE
RONDOR
MILANWOOD
KASAMA
WALESCOT
RONDOR SOUTH
HARTLEY DISTRICT
GATOOMA DISTRICT
FYLDE
DINGLEY DELL
KIRKWOOD
AMBERLEY
ARBEIDSEDEN
CURRANDOOLY
DEMONICA
TWIN LANDS
ALABAMA
EXTENSION
VICTORY
ZAWANDA
EAST MEDINA
EUREKA
HUMEWOOD
FINDON
E U R E K A
GLENVIEW
MAZARITA
ESTANCIA COREA
MASSACHUSETTS
SABLE
LINK
INNISKILLING
CROC-NA-RAGH
HILLINGTON
ANFIELD
GOLDEN VALLEY
NORMANDY NORTH
RAILWAY FARM No. 12
NORMANDY
LILY
Quartz veins
Dolerite (D), gabbro (H)
Serpentinite
Syenite porphyry, Umswezwe Farm
Porphyry
Tonalite (G), syenite (Sy)
Gneiss
Ferricrete
Karroo System
Greywacke
Argillite
Chert boulders (ch), undifferentiated
Dolomite
Quartzite
Pock-marked quartzite
Conglomerate
Arkose, grit and argillite
Conglomerate
Quartzite
Upper agglomerate
Main lava
Basal ophitic lava
Lower agglomerate
Andesitic and dacitic lavas
Volcanic sediments
Grit, quartzite and conglomerate
Andesitic greenstone
Volcanic sandstones
Banded chert and ironstone
Magnesite rock
Greywacke, argillite
What Cheer conglomerate
Basaltic greenstone
Banded chert and ironstone
Limestone, crystalline

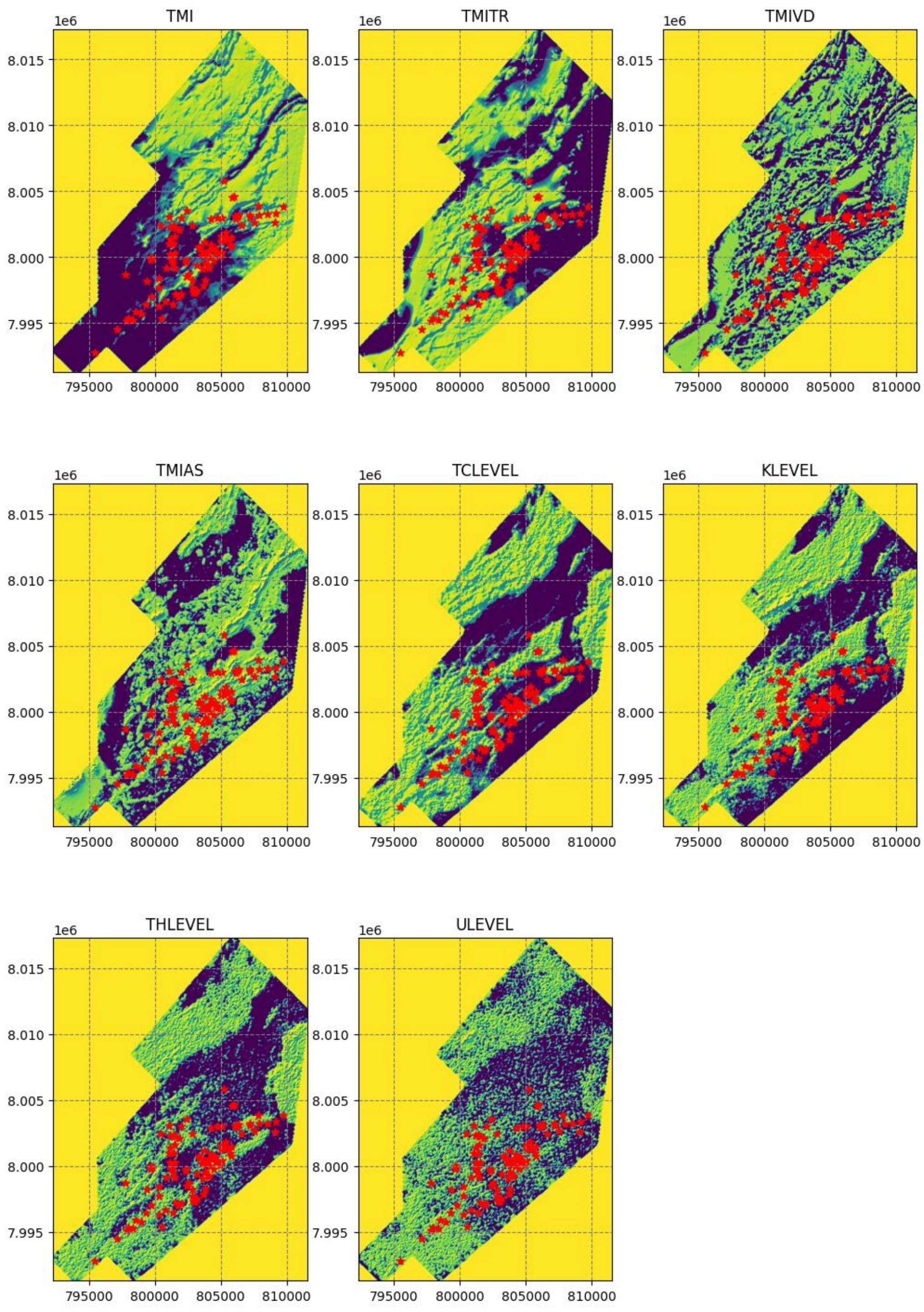

1e6
8.006
8.004
8.002
8.000
7.998
7.996
7.994
796000
798000
800000
802000
804000
806000
808000
810000

```python
df.head(10)
```

| | Lat | Long | GID | X | Y | | Name | COMMODITYS | geometry |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 29.882463 | -18.014840 | 1 | 805224.3564 | 8005796.740 | NaN | M.I.B | Au | POINT (805224.356 8005796.740) |
| 1 | 29.881301 | -18.040077 | 1 | 805057.7159 | 8003003.623 | NaN | D.P.D. | Au | POINT (805057.716 8003003.623) |
| 2 | 29.890845 | -18.039077 | 1 | 806070.6590 | 8003098.687 | NaN | ADRIATIC | Au | POINT (806070.659 8003098.687) |
| 3 | 29.893419 | -18.038326 | 1 | 806344.6487 | 8003177.560 | NaN | CUTLET | Au | POINT (806344.649 8003177.560) |
| 4 | 29.892883 | -18.039255 | 1 | 806286.2276 | 8003075.515 | NaN | STELLA | Au | POINT (806286.228 8003075.515) |
| 5 | 29.891632 | -18.040542 | 1 | 806151.4414 | 8002935.065 | NaN | GLOUCESTER | Au | POINT (806151.441 8002935.065) |
| 6 | 29.901498 | -18.039077 | 1 | 807199.2896 | 8003081.005 | NaN | TORY | Au | POINT (807199.290 8003081.005) |
| 7 | 29.901605 | -18.038004 | 1 | 807212.5159 | 8003199.598 | NaN | TIMES | Au | POINT (807212.516 8003199.598) |
| 8 | 29.908397 | -18.037790 | 1 | 807932.4952 | 8003212.045 | NaN | DIKER | Au | POINT (807932.495 8003212.045) |
| 9 | 29.914152 | -18.037289 | 1 | 808543.1434 | 8003257.870 | NaN | MELTON | Au | POINT (808543.143 8003257.870) |

```python
df.tail(10)
```

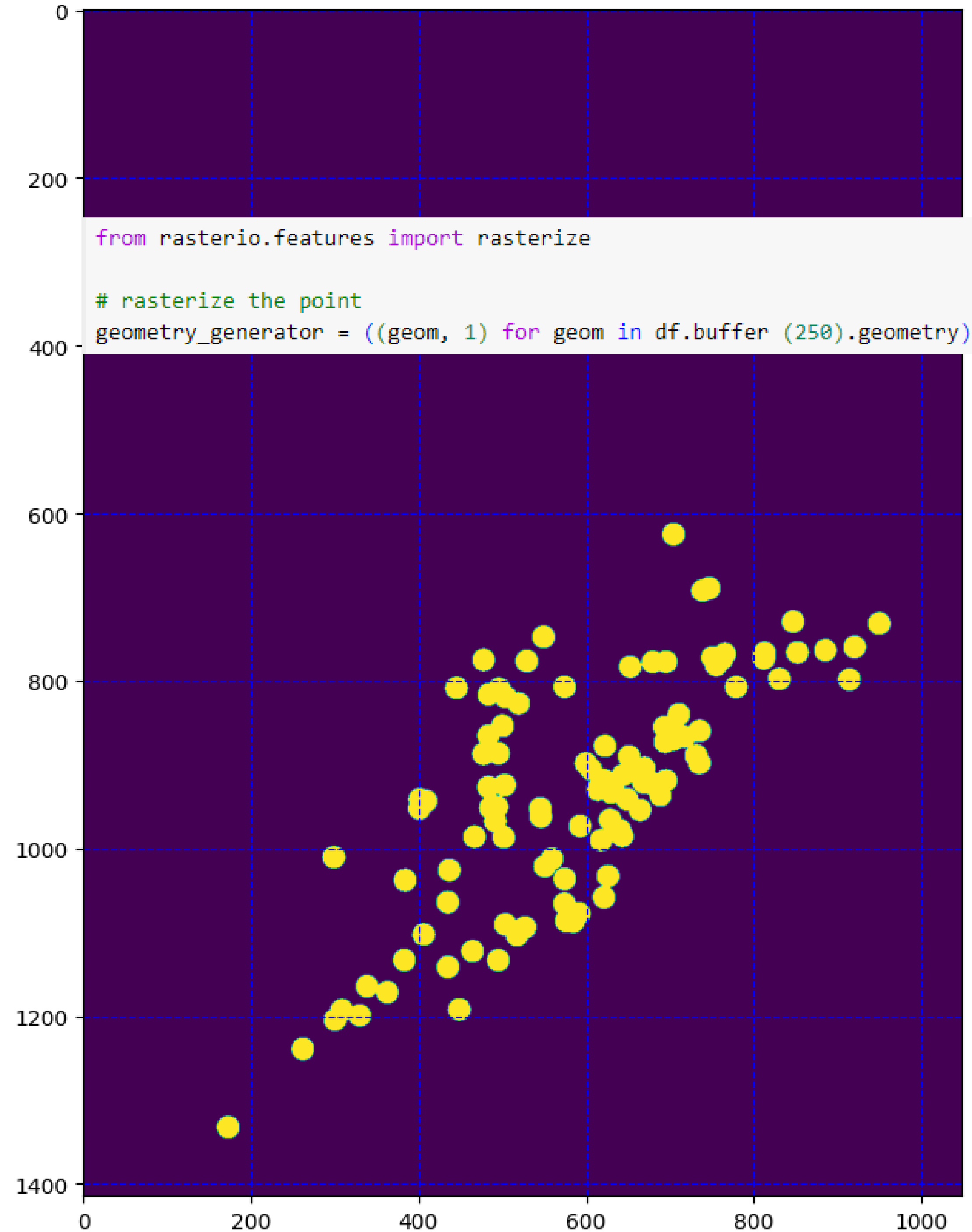
| | Lat | Long | GID | X | Y | | Name | COMMODITYS | geometry |
|---|---|---|---|---|---|---|---|---|---|
| 96 | 29.838123 | -18.045917 | 1 | 800473.2251 | 8002427.658 | NaN | H.W.J. | Au | POINT (800473.225 8002427.658) |
| 97 | 29.843616 | -18.040231 | 1 | 801064.9247 | 8003048.402 | NaN | ESMA | Au | POINT (801064.925 8003048.402) |
| 98 | 29.852529 | -18.040347 | 1 | 802009.0058 | 8003021.089 | NaN | SISTER MARRY | Au | POINT (802009.006 8003021.089) |
| 99 | 29.860443 | -18.045341 | 1 | 802838.9041 | 8002455.020 | NaN | SUNDOWN | Au | POINT (802838.904 8002455.020) |
| 100 | 29.873890 | -18.041153 | 1 | 804270.6507 | 8002896.688 | NaN | GOLDEN GLADE | Au | POINT (804270.651 8002896.688) |
| 101 | 29.878538 | -18.040155 | 1 | 804764.8705 | 8002999.648 | NaN | JODOL | Au | POINT (804764.870 8002999.648) |
| 102 | 29.889948 | -18.025364 | 1 | 805999.3402 | 8004618.885 | NaN | ROSS | Au | POINT (805999.340 8004618.885) |
| 103 | 29.888604 | -18.025863 | 1 | 805856.0046 | 8004565.798 | NaN | BABY | Au | POINT (805856.005 8004565.798) |
| 104 | 29.855910 | -18.035506 | 1 | 802375.4628 | 8003551.651 | NaN | WELCOME FRIEND | Au | POINT (802375.463 8003551.651) |
| 105 | 29.925177 | -18.031856 | 1 | 809720.7821 | 8003841.138 | NaN | SAN TOY | Au | POINT (809720.782 8003841.138) |




Load the 2D model data using the Rasterio Library

```python
data, names = [], []
for fn in geotiffs:
    with rasterio.open(fn, 'r') as src:

        transform = src.transform
        region = (src.bounds.left, src.bounds.right, src.bounds.bottom, src.bounds.top)

        d = src.read(1).astype('float')
        nodata_mask = d == src.nodata
        d[nodata_mask] = np.nan
        # append data to lists
        data.append(d)
        names.append(os.path.basename(fn).replace('.tif',''))

# stack list into 3D numpy array
data = np.stack(data)
data.shape, names
```

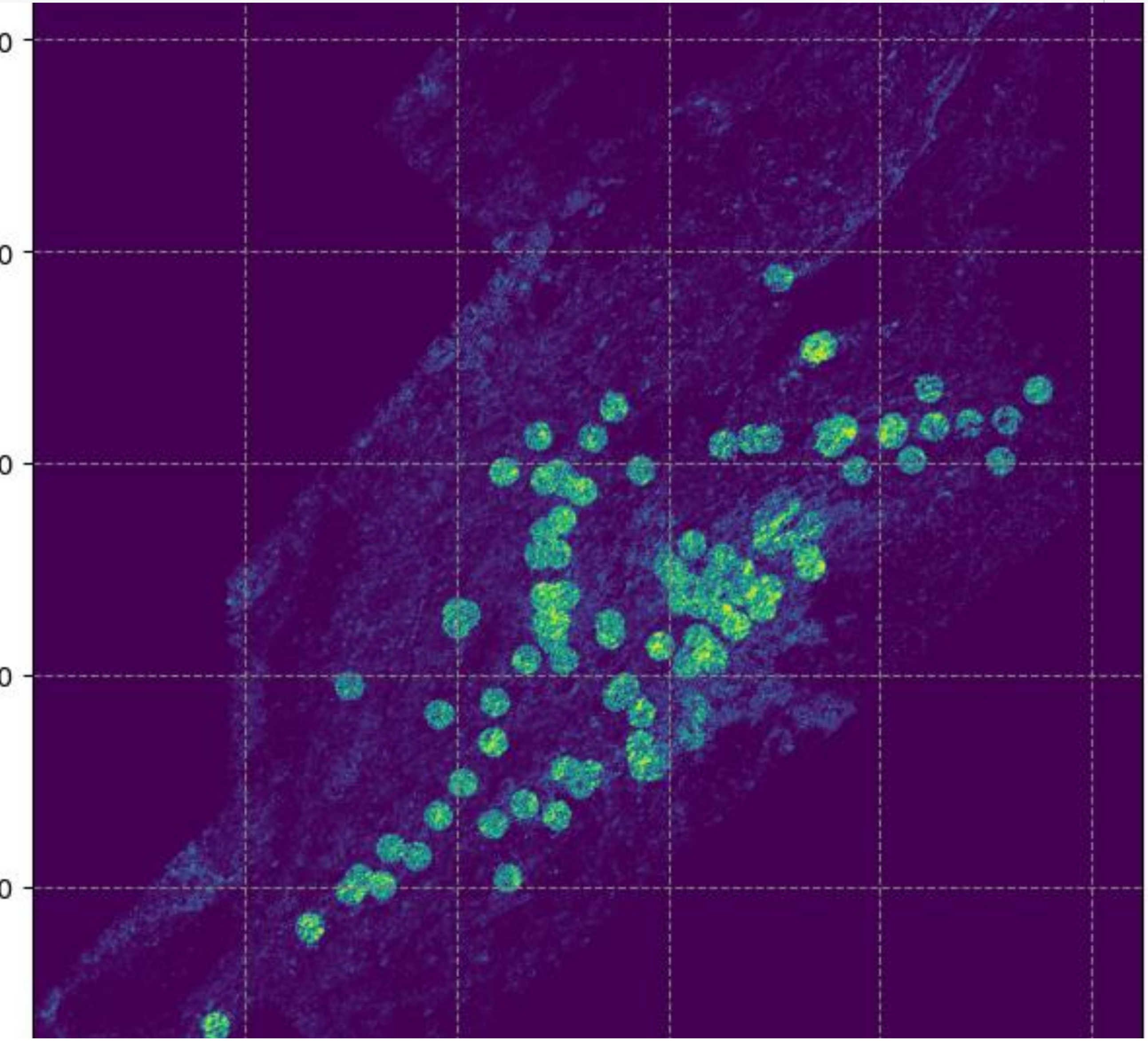
Thomas Ostersen etal

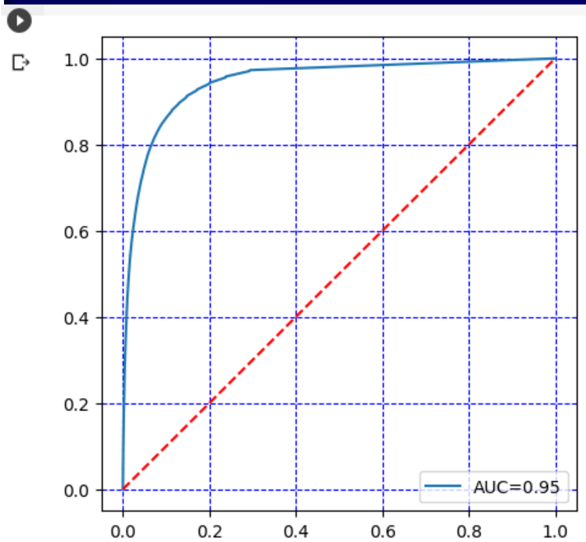



```python
from rasterio.features import rasterize

# rasterize the point
geometry_generator = ((geom, 1) for geom in df.buffer (250).geometry)
```

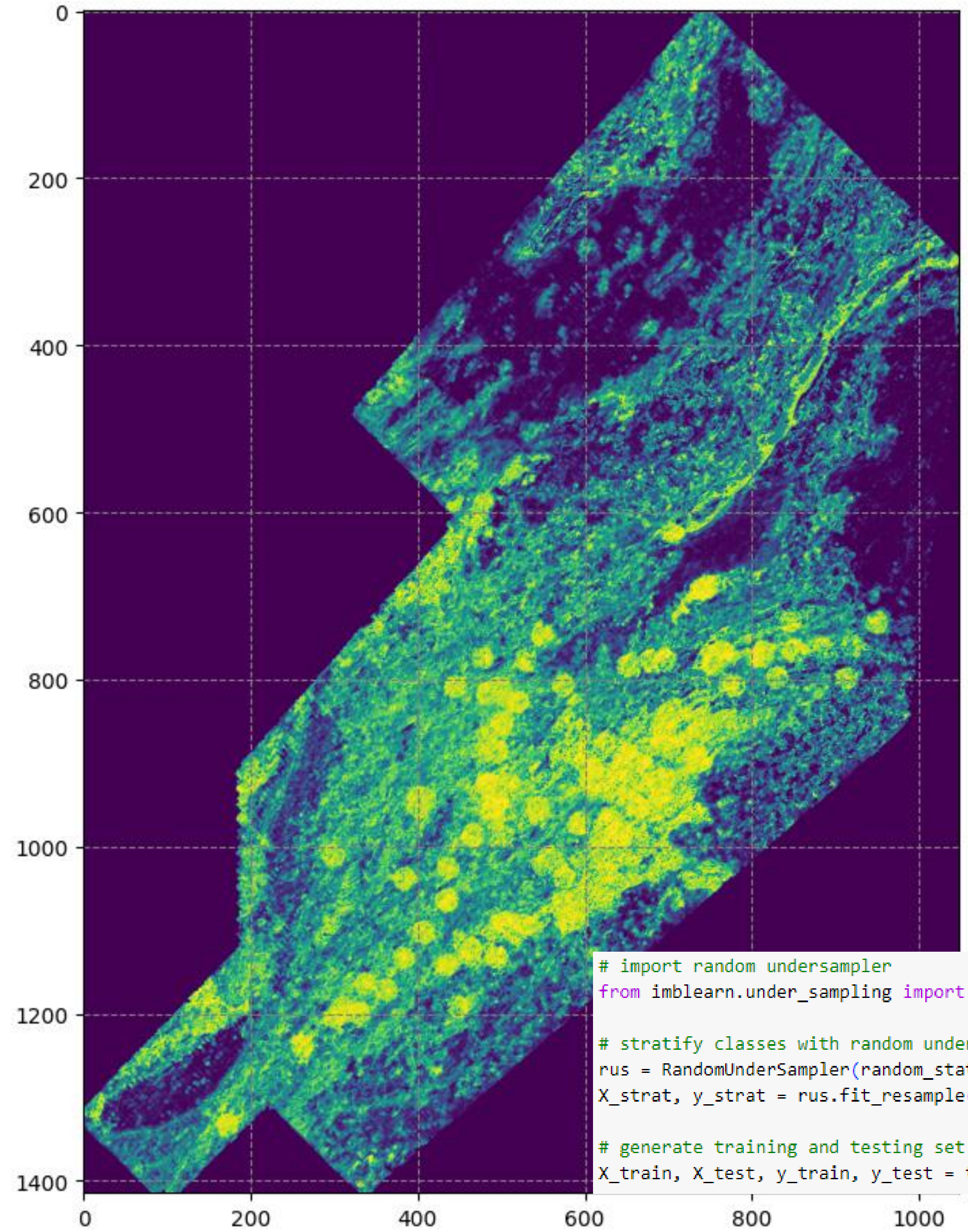
Convert the minnocs point data into a raster map using the rasterise library

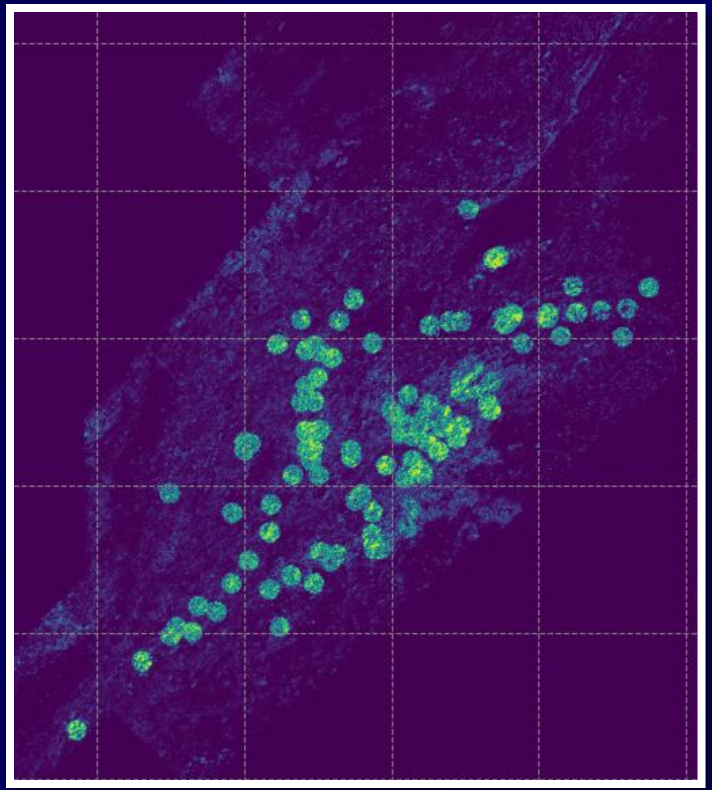
```python
# import modelling modules
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# generate train and testing subsets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=420)
```





## Applying the undersampler to the data



```python
# import random undersampler
from imblearn.under_sampling import RandomUnderSampler

# stratify classes with random undersampler
rus = RandomUnderSampler(random_state=32)
X_strat, y_strat = rus.fit_resample(X, y)

# generate training and testing set
X_train, X_test, y_train, y_test = train_test_split(X_strat, y_strat, test_size=0.33, random_state=42)
```

# CLASSIFICATION

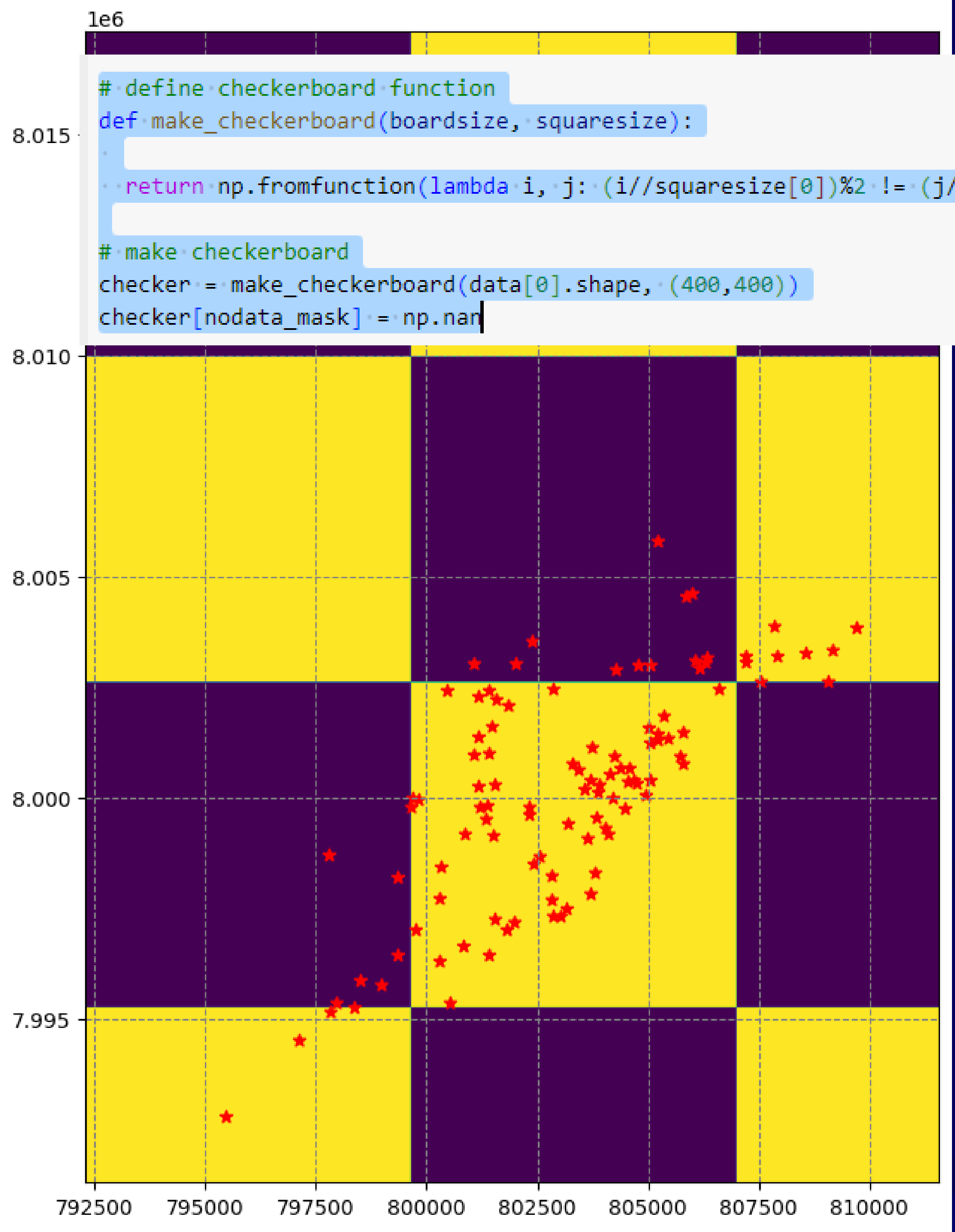
CHECKERBOARD –

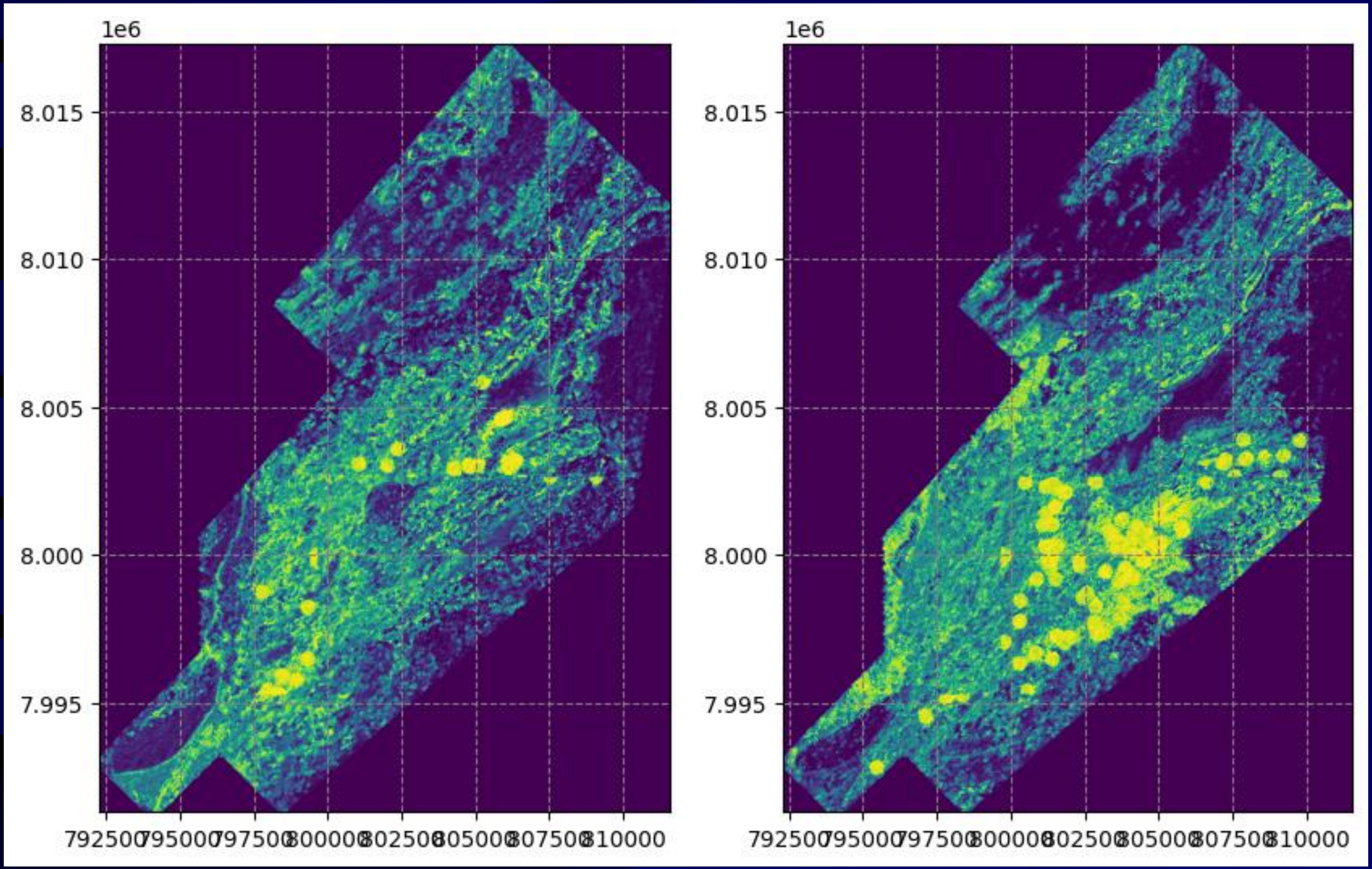
K-MEANS -

RANDOM FOREST -

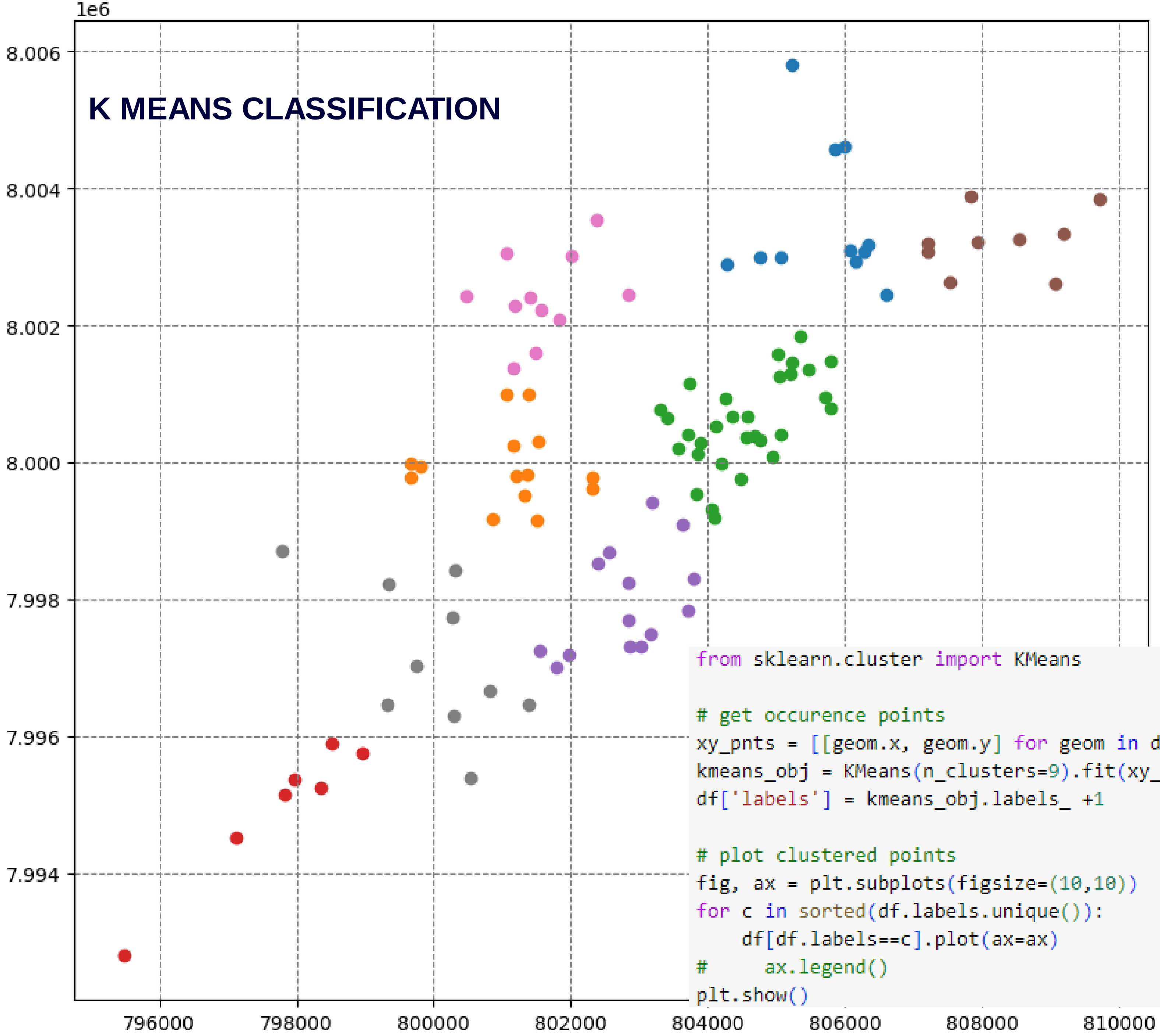
SIMPLE VECTOR MACHINE -

```python
# define checkerboard function
def make_checkerboard(boardsize, squaresize):

    return np.fromfunction(lambda i, j: (i//squaresize[0])%2 != (j//squaresize[1])%2, boardsize).astype('float32')

# make checkerboard
checker = make_checkerboard(data[0].shape, (400,400))
checker[nodata_mask] = np.nan
```



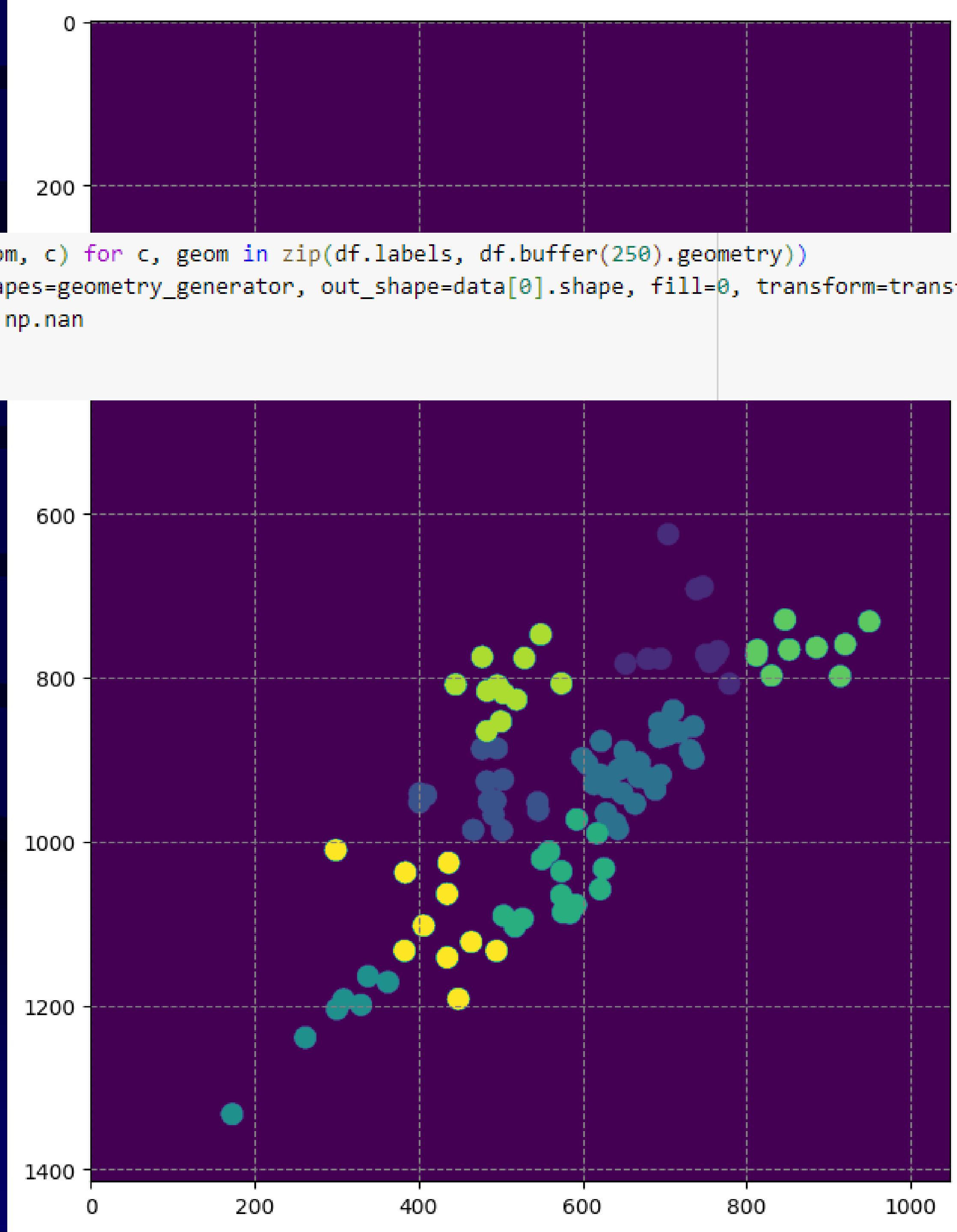

1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000

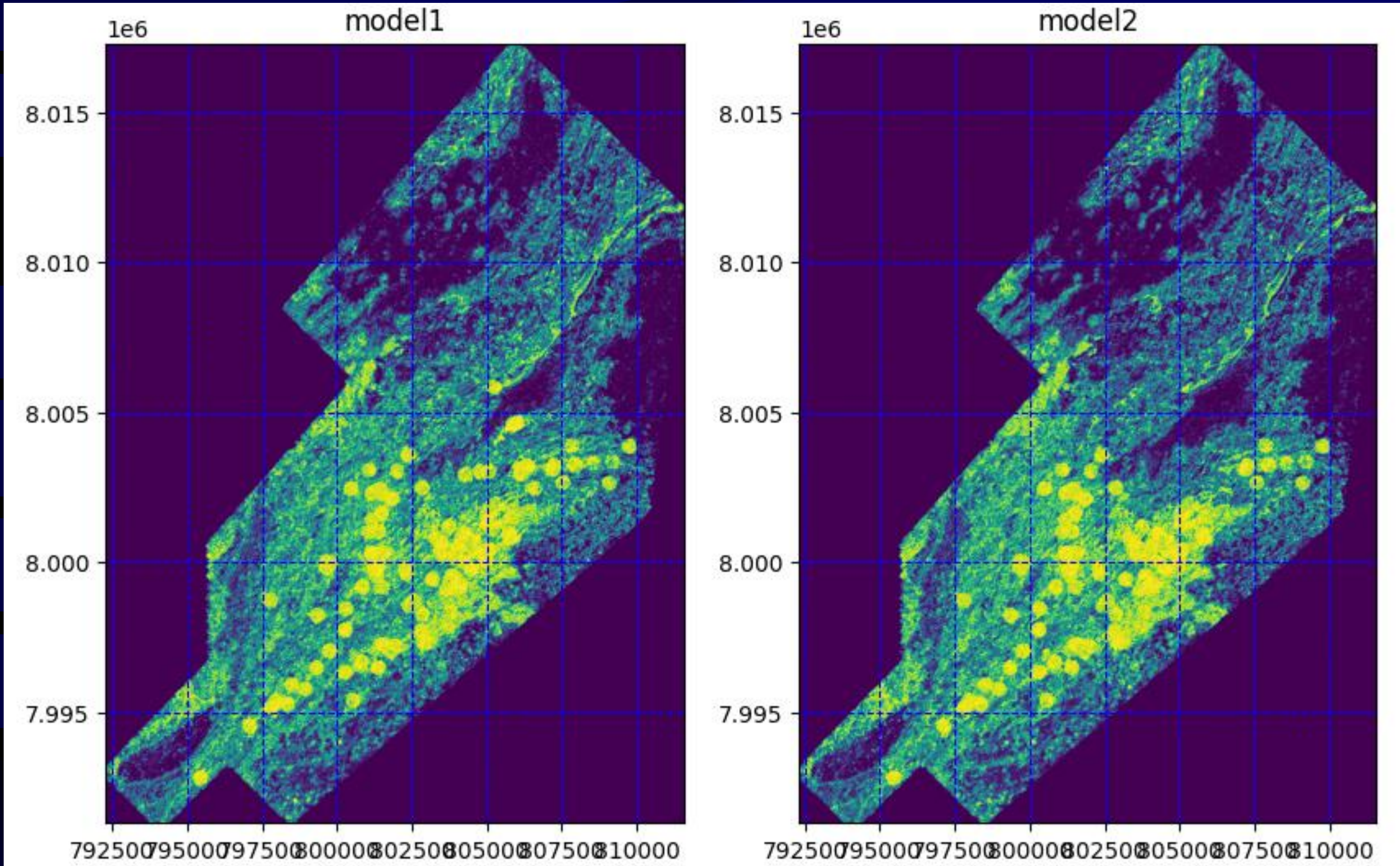
# K MEANS CLASSIFICATION

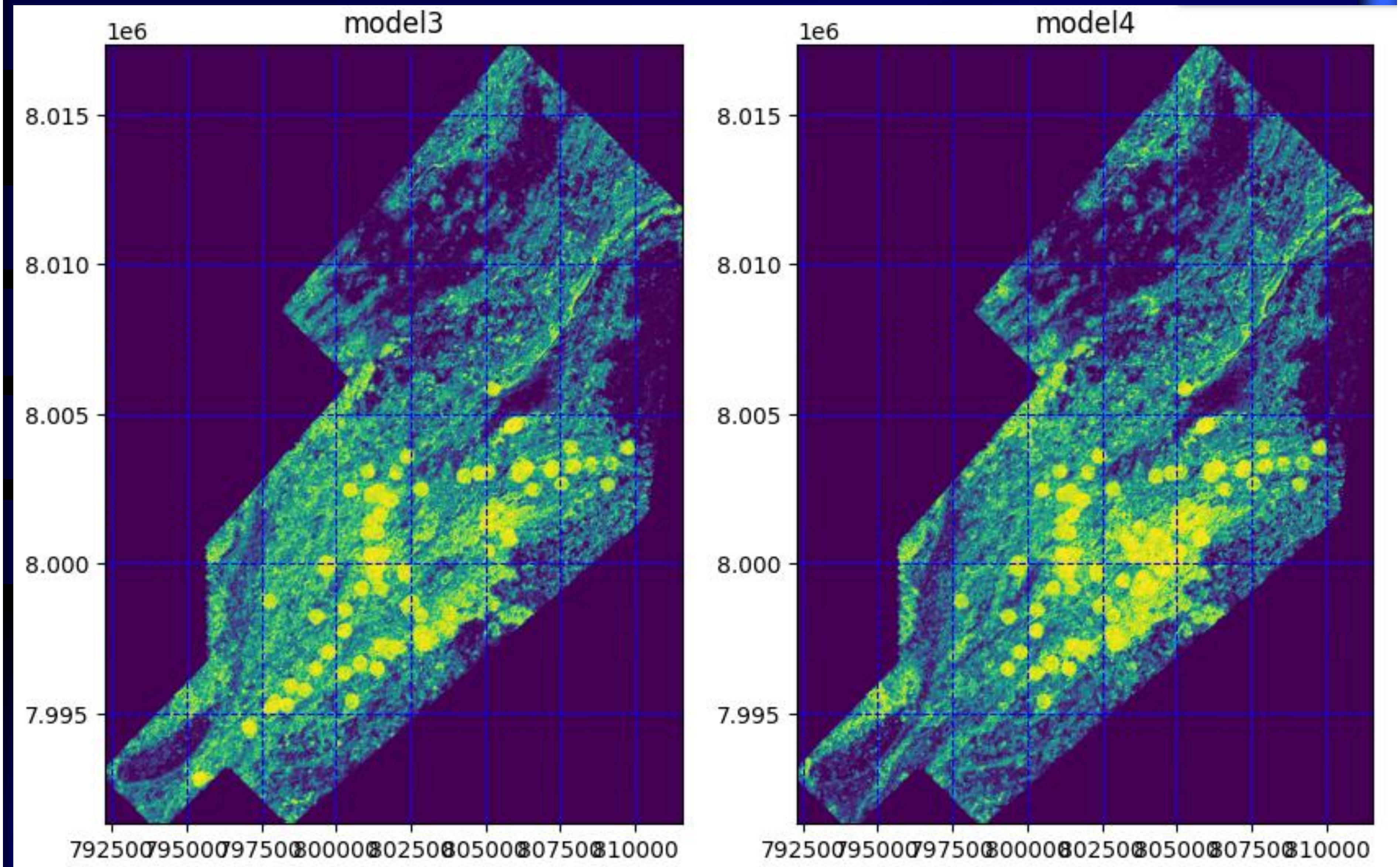



```python
from sklearn.cluster import KMeans

# get occurence points
xy_pnts = [[geom.x, geom.y] for geom in df.geometry]
kmeans_obj = KMeans(n_clusters=9).fit(xy_pnts)
df['labels'] = kmeans_obj.labels_ +1

# plot clustered points
fig, ax = plt.subplots(figsize=(10,10))
for c in sorted(df.labels.unique()):
    df[df.labels==c].plot(ax=ax)
#     ax.legend()
plt.show()
```

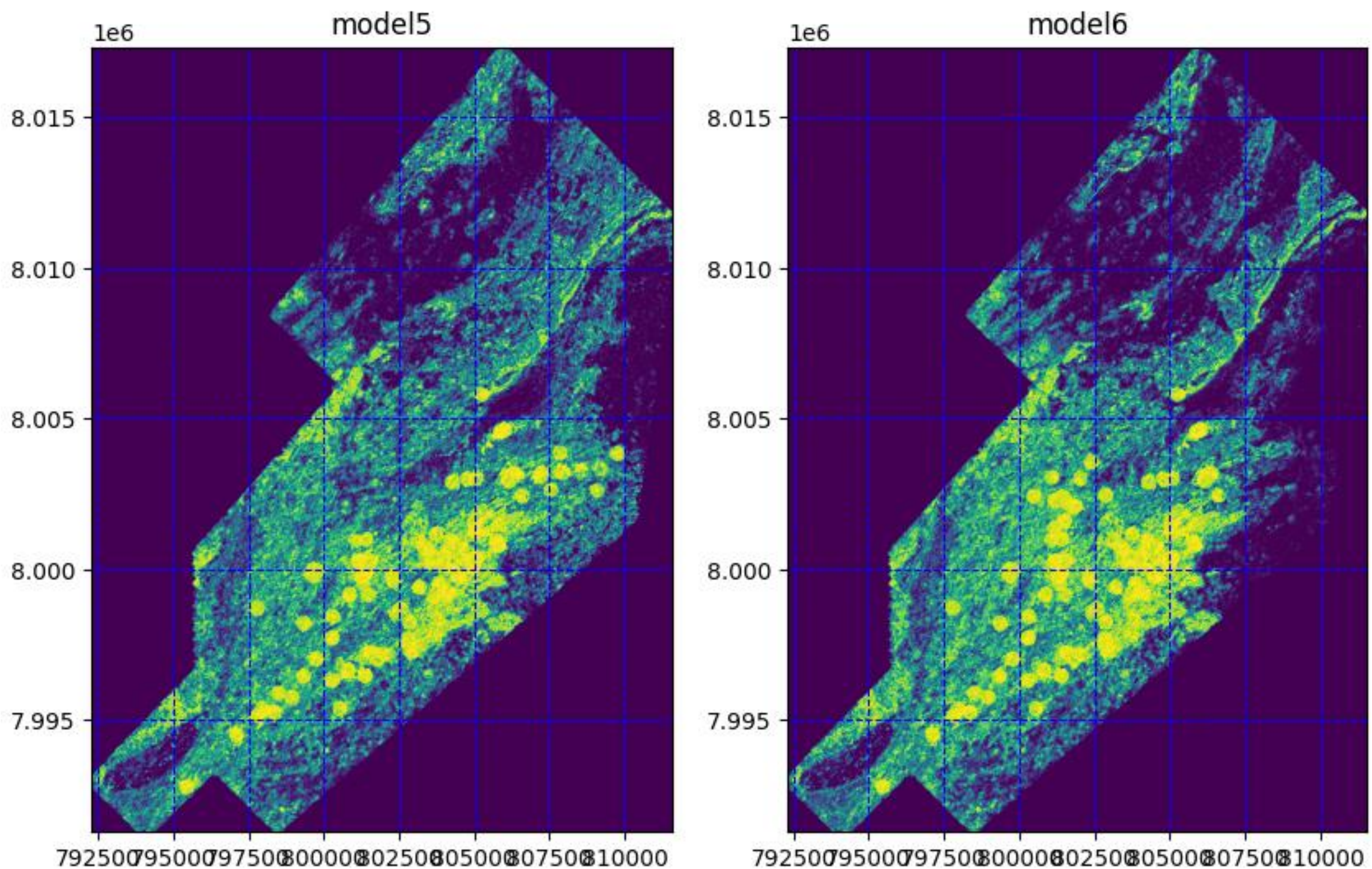
```
geometry_generator = ((geom, c) for c, geom in zip(df.labels, df.buffer(250).geometry))
clustermap = rasterize(shapes=geometry_generator, out_shape=data[0].shape, fill=0, transform=transform).astype('float32')
clustermap[nodata_mask] = np.nan

plt.imshow(clustermap)
```



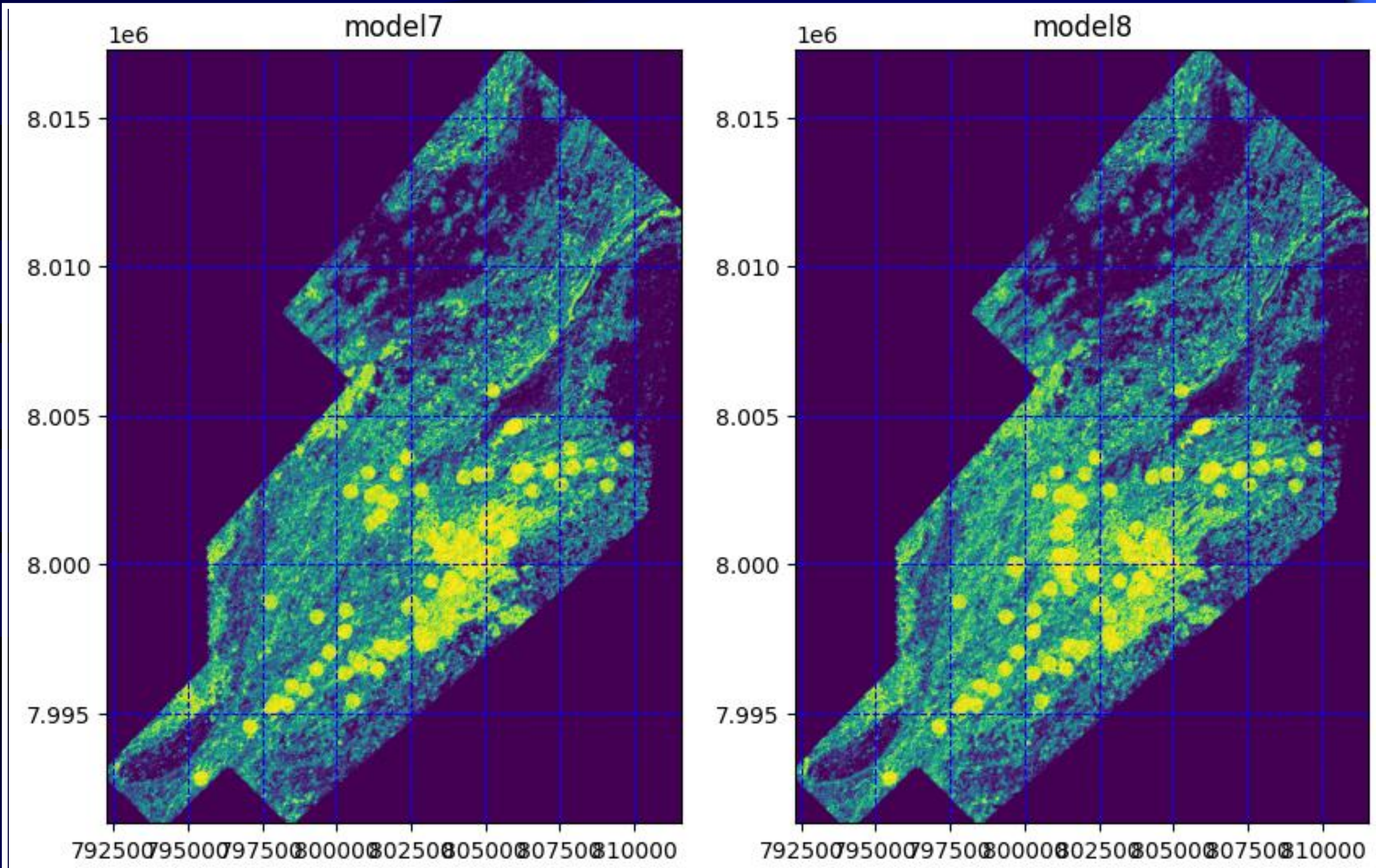

model1
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000
model2
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000

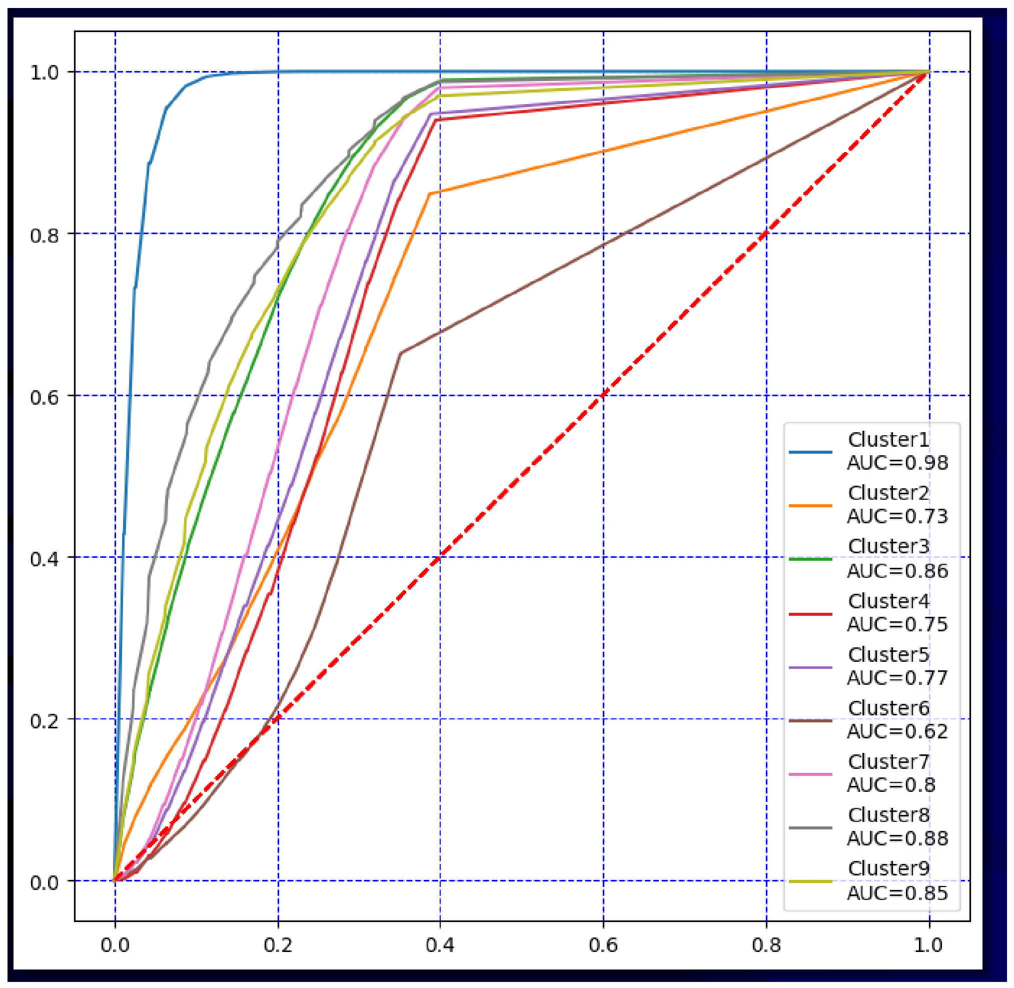


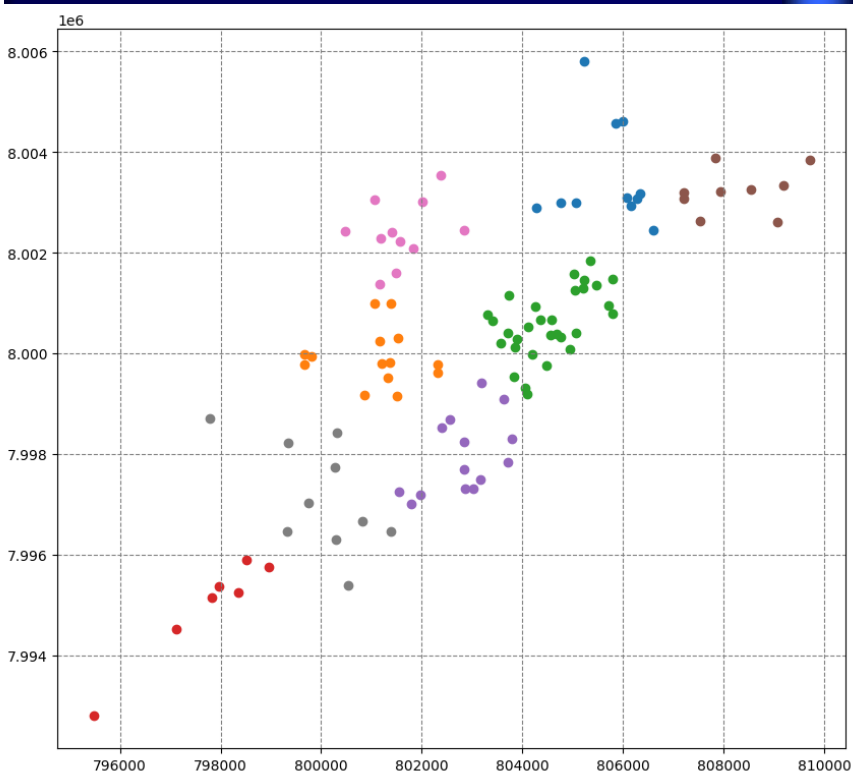
model3
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000
model4
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000



model5
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000
model6
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000


model7
model8
1e6
8.015
8.010
8.005
8.000
7.995
792500
795000
797500
800000
802500
805000
807500
810000


Cluster1
AUC=0.98
Cluster2
AUC=0.73
Cluster3
AUC=0.86
Cluster4
AUC=0.75
Cluster5
AUC=0.77
Cluster6
AUC=0.62
Cluster7
AUC=0.8
Cluster8
AUC=0.88
Cluster9
AUC=0.85
0.0
0.2
0.4
0.6
0.8
1.0


1e6
8.006
8.004
8.002
8.000
7.998
7.996
7.994
796000
798000
800000
802000
804000
806000
808000
810000

# CONCLUSIONS

While implementing various machine learning applications, it became apparent that quality controlling the outcomes plays an important role not only in building confidence in the algorithm but also in addressing two scepticisms: (1) the concern that machines will replace humans and (2) concern over black-box-type algorithms.

A 98% accuracy is attained using the k-means clustering to train and test model