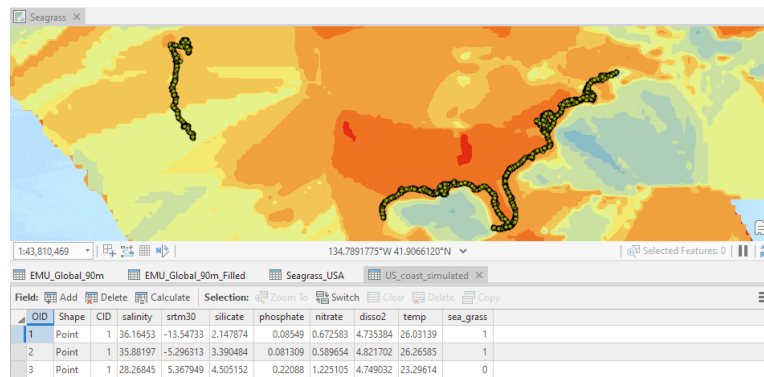


Exercise: Building your own decision tree and random forest models to predict the existence of seagrass

In this exercise, you will build your own decision tree and random forest models to complete a task very similar to what we have demoed in the class. You will work on the same dataset, but your goal will be to predict the existence of seagrass (a classification problem), rather than to predict the concentration of nitrate as in our demo.



(Credit: The exercise dataset is revised based on the tutorial here:
<https://learn.arcgis.com/en/projects/predict-seagrass-habitats-with-machine-learning/>)

Task 1: Load the shapefile data using GeoPandas, show its top 10 records, and plot out the geographic data.

```
import geopandas as gpd
coastline_attr_gpd = gpd.read_file("your Google Drive path")
coastline_attr_gpd.head(10)
coastline_attr_gpd.plot(figsize=(10,8))
```

Task 2: Simple geospatial feature engineering

```
coastline_attr_gpd["x"] = coastline_attr_gpd.geometry.x
coastline_attr_gpd["y"] = coastline_attr_gpd.geometry.y
```

Task 3: Divide the data into 80% for training and 20% for testing

```
training_data = coastline_attr_gpd.sample(frac=0.8, random_state=42)
test_data = coastline_attr_gpd.drop(training_data.index)
```

Task 4: Prepare the training and test data

```
training_label = training_data.pop("sea_grass")
test_label = test_data.pop("sea_grass")
training_data = training_data[["salinity", "srtn30", "silicate", "phosphate", "disso2", "temp", "nitrate", "x", "y"]]
test_data = test_data[["salinity", "srtn30", "silicate", "phosphate", "disso2", "temp", "nitrate", "x", "y"]]
```

Task 5: Create and train decision tree and random forest models

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
decision_tree_model = DecisionTreeClassifier(random_state=42, min_samples_leaf=10)
random_forest_model = RandomForestClassifier(random_state=42, min_samples_leaf=10)
decision_tree_model.fit(training_data, training_label)
random_forest_model.fit(training_data, training_label)
```

Task 6: Make predictions and evaluate the classification accuracy

```
from sklearn.metrics import accuracy_score
y_pred_1 = decision_tree_model.predict(test_data)
accuracy_1 = accuracy_score(test_label, y_pred_1)
y_pred_2 = random_forest_model.predict(test_data)
accuracy_2 = accuracy_score(test_label, y_pred_2)
```