TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN

BÁO CÁO BÀI TẬP LỚN
TRÍ TUỆ NHÂN TẠO – Tổng Kết

Nhóm 8:

Nguyễn Đam Trường

Mai Tất Thắng

Trịnh Ngọc Du

Nguyễn Thanh Lâm

Hà Nội, ngày 24 tháng 12 năm 2021

Mục tiêu báo cáo :

I. Giới thiệu vấn đề

II. Chuẩn bị dữ liệu đào tạo

III. Giới thiệu bộ đào tạo dữ liệu

IV. Kết quả đào tạo bộ dữ liệu

Lời cảm ơn

Cảm ơn cô và anh đã giúp đỡ bọn em hoàn thành bài tập lớn này

I. Giới thiệu vấn đề

1.1. Nêu ra vấn đề

Ngày nay, với sự gia tăng về mặt dân số và sự suy thoái của môi trường rừng và một số vấn đề liên quan khác đều cần có sự quan trọng của bản đồ. Và việc phân loại đặc tính trên bản đồ, phân loại các vùng trên bản đồ đóng vai trò quan trọng trong việc phát triển định hướng. Trên thực tế mọi người lấy việc phân lớp bằng cách đi thực địa và ghi chép lại những gì đo đạc được. Vì vậy, việc thực đi thực địa khảo sát từng vùng để lấy phân loại lớp phủ sẽ rất tốn thời gian và công sức của con người. Không chỉ vậy việc lấy vùng thủ công sẽ rất mất nhiều thời gian nhưng đến khi cần dùng lại phải đi thực địa lại chưa chắc đã đúng với thực trạng hiện tại mà nơi đã lấy mẫu. Với khoa học phát triển, đã có những vệ tinh được phóng lên phục vụ con người trong việc này.

1.2. Giải pháp vấn đề

Những vệ tinh được phóng lên trên tầng khí quyển ghi lại hình ảnh dưới mặt đất và quan sát các hiện tượng thời tiết được nhờ các bức xạ mặt trời phản chiếu xuống mặt đất và hình ảnh vệ tinh thu được trên mỗi vùng khác nhau sẽ có những bức xạ thu được khác nhau. Và những vùng có tính chất giống nhau sẽ thu được các bức xạ có đặc tính gần giống nhau, tiếp theo có thể nhờ đến sự huấn luyện của học máy để phân biệt các vùng hoặc các nhãn mà một vùng đất có thể có. Làm vậy có thể tiết kiệm được rất nhiều thời gian công sức cũng như chi phí để có thể giải quyết vấn đề phân vùng trên mặt đất.

1.3. Bài toán đặt ra

Phân loại lớp phủ dựa bề mặt trên TP.Hồ Chí Minh năm 2020 và dựa theo kết quả của vệ tinh USGS Landsat8 level 2 - Bài toán thuộc bài toán phân loại.

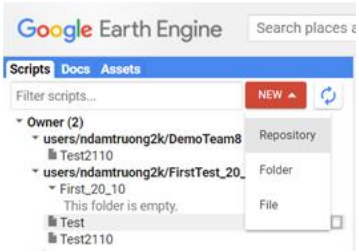
II. Chuẩn bị dữ liệu để đào tạo

2.1. Lấy dữ liệu ảnh khu vực của vệ tinh

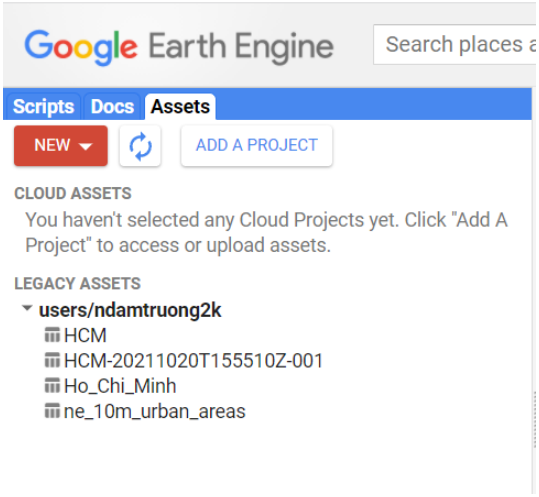
Quá trình thực hành dữ liệu trên Google Earth Engine

1. Tập dữ liệu Landsat8 level 2
2. Khu vực HCM
3. Thời gian 01-01-2020 đến 31-12-2020
4. Áp dụng hệ số scale 10000, không có offset
5. Lọc và cắt theo vùng nghiên cứu HCM

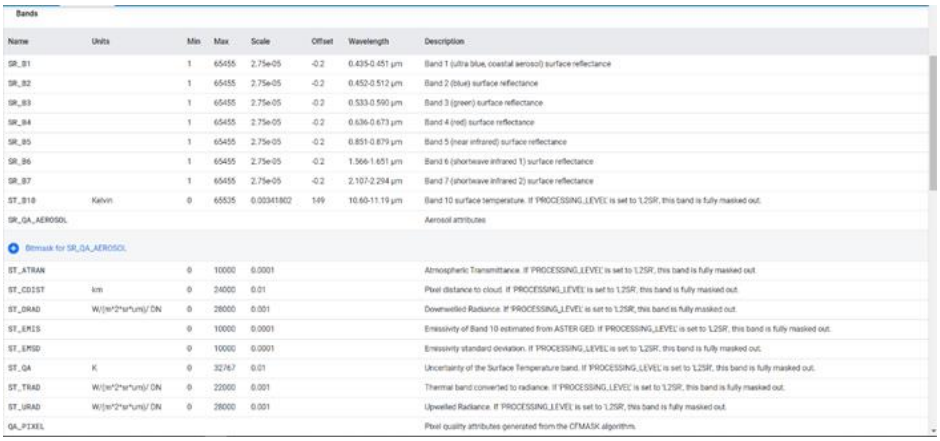
Bước 1: Tạo một New Responsitory trên Google Earth Engine, tiếp đến tạo 1 Folder làm việc File trên đó.



Bước 2: Add các vùng cấu trúc cho sẵn để chia theo từng vùng và nạp chúng, nhóm 8 khảo sát khu vực HCM nên chỉ thêm dữ liệu ở HCM



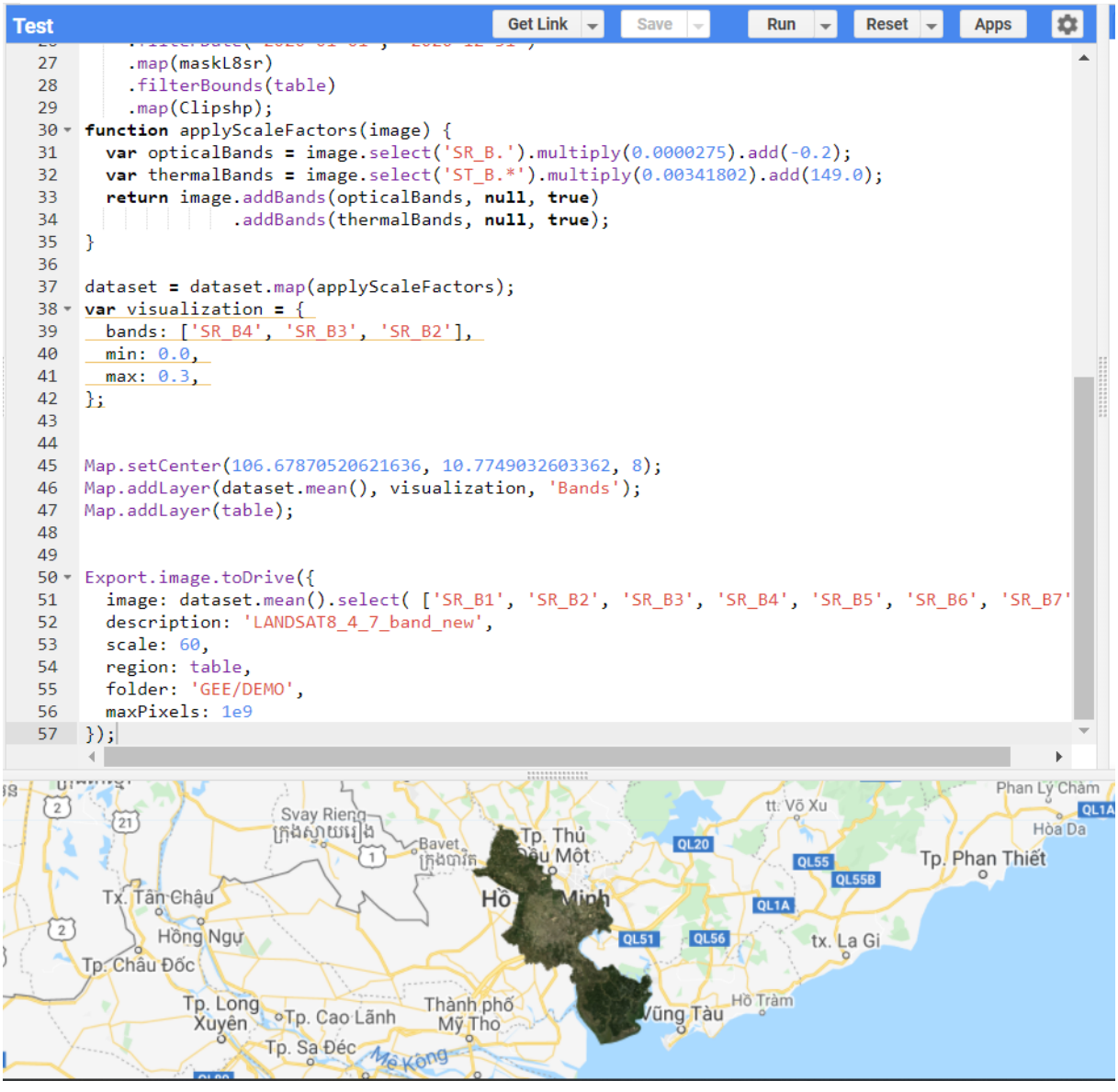
Bước 3: Đọc và lấy thông tin của vệ tinh mà nhóm được giao cụ thể nhóm 8 sử dụng vệ tinh Landsat8 và dữ liệu của [USGS Landsat 8 Level 2, Collection 2, Tier 1]



Hình 2.1: Các Band có trong dữ liệu của vệ tinh Landsat8

Bước 4: Sử dụng thư viện và các thông tin có được từ trang web Landsat8 sẽ code các lệnh

Hình 2.2: Code mẫu có trong Landsat Level2 1



Hình 2.3: Code được trong Google Earth Engine (sử dụng với ban suface SB_B1 đến SB_B7)

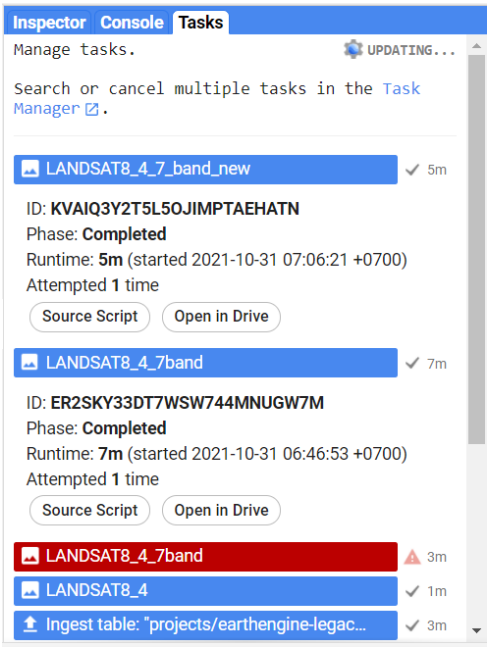
```
function maskL8sr(image) {
    var qaMask = image.select('QA_PIXEL')
        .bitwiseAnd(parseInt('11111', 2)).eq(0);

    var saturationMask = image
        .select('QA_RADSAT').eq(0);

    return image.updateMask(qaMask)
            .updateMask(saturationMask);
}
```

Hình 2.4: Hàm lọc mây

```
function applyScaleFactors(image) {
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
  var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
  return image.addBands(opticalBands, null, true)
              .addBands(thermalBands, null, true);
}
```

Hình 2.5: Hàm ScaleFactor

Bước 5: Lấy ảnh từ thanh task và được lưu trên Google Drive



Hình 2.6: Lưu dữ liệu dưới dạng GeoTiff

2.2. Lấy mẫu từng vùng trên khu vực

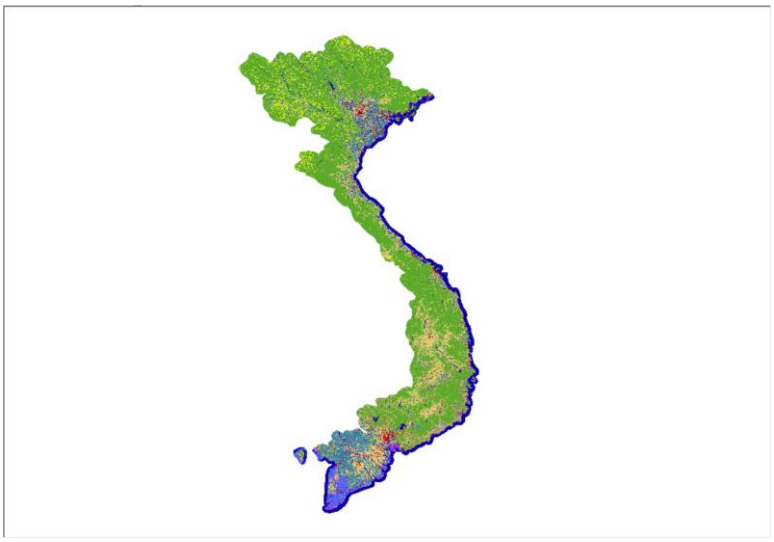
Các công cụ cần có để phục vụ cho việc lấy mẫu hiệu quả :

1. ARC Map
2. Google Earth Pro
3. Giá trị của mỗi pixel chỉ ra một loại đất che phủ được phân loại

Các bước thể hiện quy trình lấy mẫu:

Bước 1:

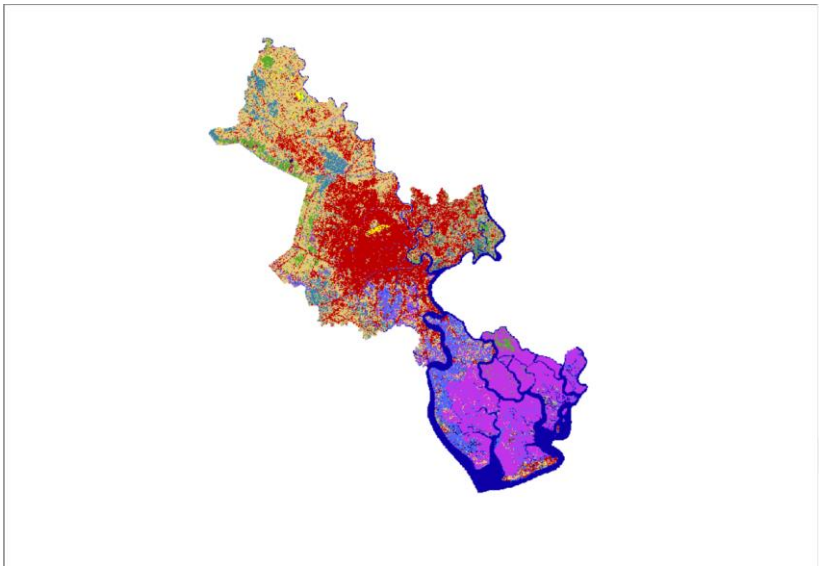
Lấy ảnh chứa giá trị của mỗi pixel chỉ ra một loại đất che phủ phân loại:



Hình 2.7: Phân loại lớp phủ bản đồ Việt Nam (chưa tính Hoàng Sa và Trường Sa)

Bước 2:

Cắt khu vực được giao để lấy mẫu (Cụ thể nhóm 8 được giao khu vực TP.HCM):



Hình 2.8: Phân loại lớp phủ của khu vực TP.HCM

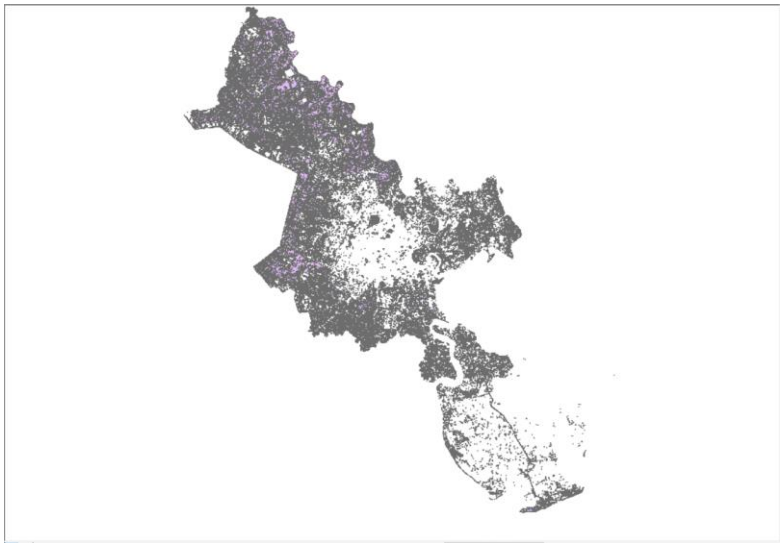
Bước 3:

Sử dụng tool Raster to Polygon để biến đổi ảnh thành các polygon:

Hình 2.9: Mẫu Polygon từ TP.HCM

Bước 4:

Trích xuất ra giá trị của pixel mong muốn dựa trên bảng giá trị:

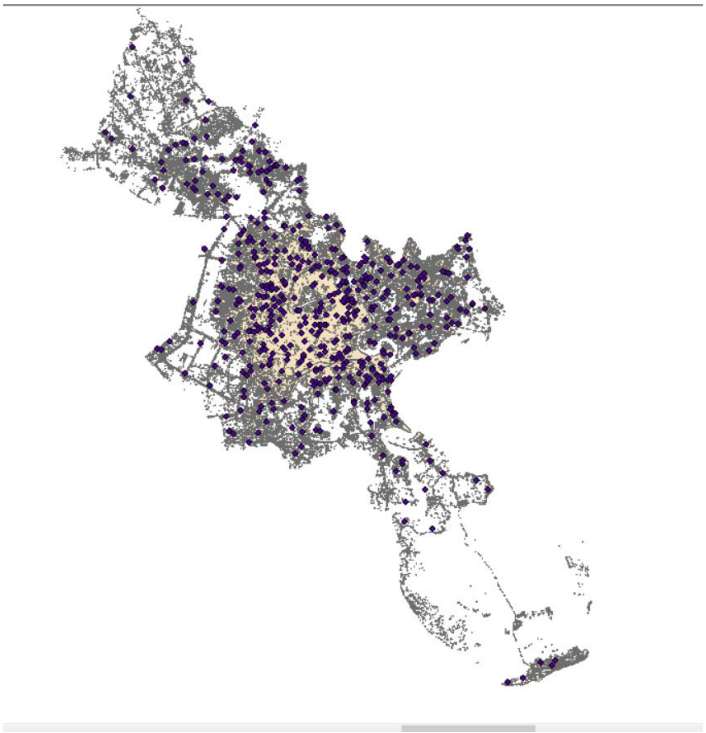


Level-1
#1 and #2: Residential land
#3: Rice paddies
#4, #5, and #6: Croplands
#7: Grassland
#8: Barren land
#9: Scrub/Shrub
#10, #11, #12, 14#, and #20: Forests
#15: Wetlands
#18: Open water
#19: Aquaculture

Hình 2.10: Lựa chọn mẫu từ khu vực TP.HCM và Bảng Level các giá trị lớp phủ

Bước 5:

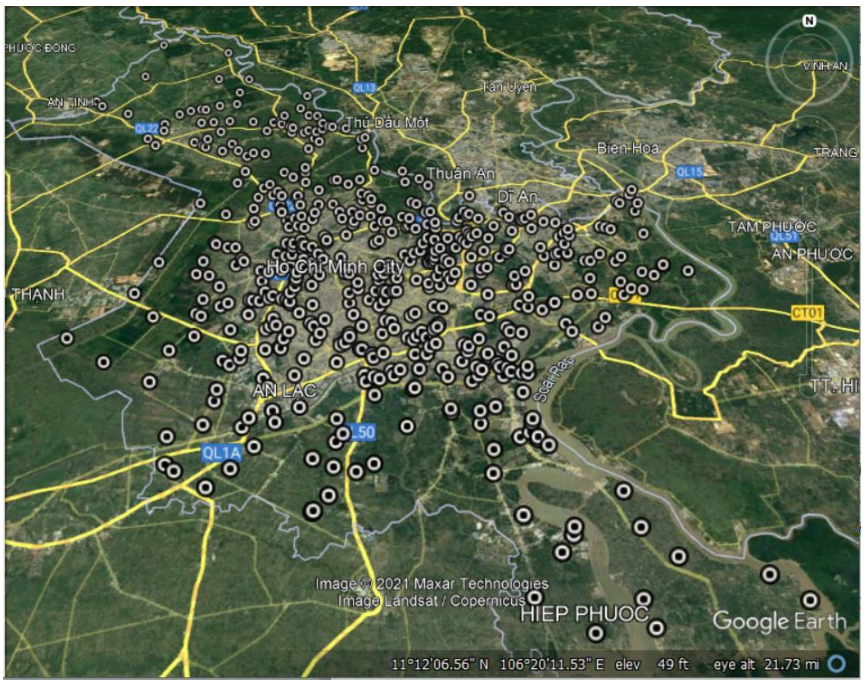
Hợp nhất các polygon và lấy các điểm ngẫu nhiên trên Polygon dùng tool Random Point :



Hình 2.11: 150-200 điểm ngẫu nhiên thuộc phân lớp Residential Lands trên TP.HCM

Bước 6:

Export các điểm và đưa vào Google Earth Pro để kiểm tra và chỉnh sửa các điểm:



Hình 2.12: Các điểm đã qua chỉnh sửa trên Google Earth Pro

Bước 7:

Xuất các điểm lại vào ARC Map và đổi hệ tọa độ sang UTM và export vào Excel:

	OBJECTID *	Shape *	OID_	Name	FolderPath	SymbolID	AltMode	Base	Snippet	PopupInfo	HasLabel	LabelID	X	Y
▸	369	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	649255.2	649255.2
	449	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	649787.8	649787.8
	229	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	650549.1	650549.1
	487	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	650620.6	650620.6
	400	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	650723.9	650723.9
	184	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	650777	650777
	103	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	651449.5	651449.5
	285	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	652613	652613
	175	Point ZM	0	Placemark	Croplands_samples.shp/Croplands_samples	0	-1	0			-1	0	652794.5	652794.5

Hình 2.13: Bảng số liệu thống kê được lấy trên các điểm

	A	B	C
1	Label	X	Y
2	Residential_Land	673180	1213718
3	Residential_Land	690909	1182155
4	Residential_Land	683742	1189353
5	Residential_Land	687055	1201077
6	Residential_Land	690292	1172595
7	Residential_Land	667824	1210160
8	Residential_Land	673352	1188191
9	Residential_Land	700476	1202367
10	Residential_Land	683676	1196631
11	Residential_Land	677657	1203044
12	Residential_Land	680127	1191384
13	Residential_Land	681900	1184756
14	Residential_Land	675571	1192713
15	Residential_Land	681306	1185776
16	Residential_Land	690097	1176928
17	Residential_Land	688093	1182720
18	Residential_Land	690558	1197981
19	Residential_Land	685784	1198012
20	Residential_Land	661156	1211991
21	Residential_Land	677408	1190601
22	Residential_Land	691019	1192335
23	Residential_Land	674311	1179635
24	Residential_Land	685394	1190095
25	Residential_Land	689737	1200517
26	Residential_Land	666308	1210744
27	Residential_Land	700703	1205042
28	Residential_Land	706126	1148046
29	Residential_Land	671276	1192067
30	Residential_Land	690903	1195153
31	Residential_Land	662022	1214244

Hình 2.14: Xuất bảng số liệu ra Excel

	A	B	C	D	E	F	G	H	I	J
1	Label	X	Y	Band_1	Band_2	Band_3	Band_4	Band_5	Band_6	Band_7
2	Residentia	673180.3	1213718	0.066463	0.081176	0.121876	0.123805	0.275585	0.24341	0.181944
3	Residentia	690909.2	1182155	0.062819	0.073528	0.10536	0.100736	0.255782	0.180208	0.123422
4	Residentia	683741.9	1189353	0.095798	0.107778	0.138688	0.147527	0.21325	0.230234	0.186807
5	Residentia	687054.5	1201077	0.093326	0.104977	0.139403	0.138774	0.27514	0.257691	0.201702
6	Residentia	690291.8	1172595	0.06112	0.07042	0.101802	0.102755	0.265555	0.207426	0.130775
7	Residentia	667824.1	1210160	0.074178	0.090385	0.129022	0.134269	0.244648	0.250215	0.206212
8	Residentia	673352	1188191	0.071336	0.083377	0.114586	0.120859	0.191391	0.213027	0.18472
9	Residentia	700476.3	1202367	0.074868	0.09198	0.134103	0.155825	0.23663	0.237593	0.186358
10	Residentia	683675.6	1196631	0.090688	0.105787	0.14049	0.157069	0.192187	0.19303	0.163603
11	Residentia	677657.2	1203044	0.092189	0.116249	0.171194	0.168739	0.253779	0.246467	0.209833
12	Residentia	680126.8	1191384	0.08468	0.098451	0.126579	0.127884	0.177871	0.220553	0.197637
13	Residentia	681899.6	1184756	0.090533	0.0998	0.134137	0.153946	0.243386	0.268082	0.216462
14	Residentia	675571.2	1192713	0.105609	0.12337	0.159218	0.160172	0.22791	0.25092	0.224859
15	Residentia	681305.9	1185776	0.058827	0.069024	0.101617	0.094095	0.295007	0.199898	0.123742
16	Residentia	690096.6	1176928	0.125808	0.138624	0.146696	0.142279	0.224261	0.271837	0.268378
17	Residentia	688093.2	1182720	0.045708	0.056556	0.086287	0.082451	0.194038	0.132038	0.08626
18	Residentia	690558.1	1197981	0.060863	0.069037	0.097698	0.09429	0.247512	0.177505	0.122535
19	Residentia	685783.6	1198012	0.078839	0.089719	0.121181	0.12544	0.23808	0.222976	0.168411
20	Residentia	661155.5	1211991	0.061088	0.07978	0.124693	0.131986	0.276649	0.257878	0.194042

Hình 2.15: Kết quả thu được trên Excel sau cùng

Sau khi lấy mẫu các vùng khác và thực hiện so sánh trên Google Earth Pro. Chúng tôi đã có một tệp dữ liệu đầy đủ và cần kiểm tra lại tệp dữ liệu xem trong quá trình trích xuất các band có thể không có hoặc một số band có thể bị lỗi cần xóa bỏ vị trí và hàng dữ liệu tại đó.

Tên vùng	*Gán nhãn*	*Số lượng*
Aquaculture	0	115
Barren Land	1	144
Crop Land	2	108
Forests	3	141
Grasslands	4	142
Residential Land	5	151
Rice paddies	6	124
Scrub/Shrub	7	152
Open Water	8	150

Bảng 1: Bảng giá trị số lượng mẫu của từng vùng.

Tuy nhiên có hạn chế rằng một số vùng rất khó nhìn và lấy mẫu vì Google Earth Pro ở trên cao lấy mẫu xuống rất khó để quan sát màu sắc để có thể phân biệt được như một số như: Vùng nước với vùng nuôi trồng thủy sản, hoặc giữa vùng đất trồng với khu vực lúa,…

III. Giới thiệu bộ dữ liệu đào tạo

Khi có một tệp các data cần huấn luyện thì tiếp đó sẽ phải chọn các mô hình huấn luyện. Các mô hình huấn luyện khác nhau sẽ cho ta các kết quả về hiệu suất là

khác nhau. Một số mô hình có tham số ta cần tìm ra tham số tối ưu phù hợp với bộ dữ liệu mà ta muốn xem xét. Ở đây chúng tôi sẽ nêu ra một vài bộ huấn luyện đã áp dụng.

3.1. Sử dụng mô hình Random Forests để huấn luyện bộ dữ liệu

RandomForests là một thuật toán học có giám sát. Nó được sử dụng trong cả hai bài toán phân loại và hồi quy. Nó cũng là một thuật toán linh hoạt và dễ dùng nhất.

RandomForests sử dụng nhiều cây quyết định và đưa ra dự đoán bằng cách lấy trung bình dự đoán của từng cây thành phần. Nó thường có độ chính xác dự đoán tốt hơn nhiều so với một cây quyết định đơn lẻ và nó hoạt động tốt với các tham số mặc định. Và nếu thử thay đổi tham số kết quả có thể sẽ tốt hơn.

Các tham số quan trọng của RandomForestClassifier:

Các tham số tăng hiệu quả của thuật toán:

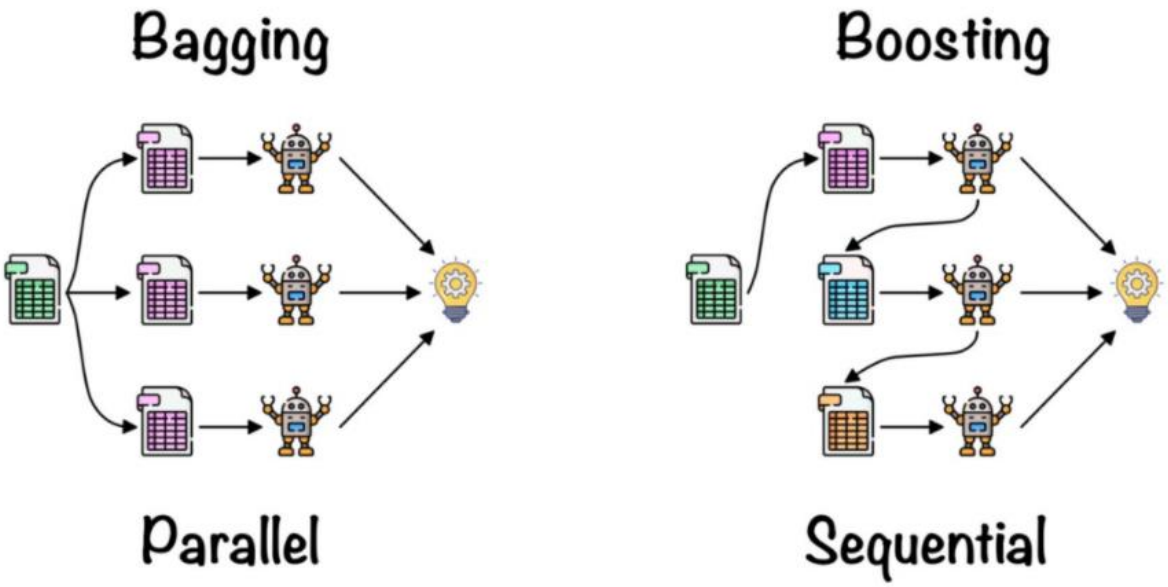
1. n_estimators : Số lượng cây mà thuật toán xây dựng trước khi tính trung bình các dự đoán.
2. max_features: Số lượng feartures tối đa mà thuật toán xem xét việc tách nút.
3. mini_sample_leaf: Số lá tối thiểu cần tách một node bên trong.

Các tham số tăng tốc độ của thuật toán:

1. n_jobs: Có bao nhiêu bộ xử lý cho phép sử dụng. Nếu giá trị n_jobs là 1 thì là 1 bộ xử lý nếu là -1 thì là không giới hạn.
2. random_state: Kiểm soát tính ngẫu nhiên.
3. oob_score: oob có nghĩa là Out Of the Bag là xác nhận chéo randomforests. Trong đây 1/3 mẫu không sử dụng để huấn luyện mà là để đánh giá hiệu suất của thuật toán.

3.2. Sử dụng mô hình XGBoost để huấn luyện thuật toán

XGBoost là thuật toán có sự nhất quán vượt trội so với các thuật toán khác với hiệu suất tốt trên nhiều bộ dữ liệu, nhiều tham số để tinh chỉnh cập nhật như cross-validation, tree parameters trong thư viện scikit-learn. Thuật toán được xây dựng tuần tự để mỗi cây tiếp theo nhằm mục đích giảm các lỗi của cây trước đó. Mỗi cây sau học hỏi cây tiền nhiệm và cập nhật các lỗi còn sót lạ. Do dó cây tiếp theo trong chuỗi sẽ học hỏi từ phiên bản cập nhật của cây còn lại.



Hình 3.2: Cách làm việc lần lượt của RandomForest và XGBoost

XGBoost có hỗ trợ hai mô hình đó là:

1. XGBoost.train
2. XGBoost.XGBRegressor và XGBoost.XGBClassifier

XGBoost.train là API cấp thấp để đào tạo mô hình thông qua phương pháp tăng độ dốc.

XGBoost.XGBRegressor và XGBoost.XGBClassifier là các trình bao bọc (các trình bao bọc giống như Scikit-Learn) chuẩn bị DMatrix và truyền vào các tham số và hàm mục tiêu tương ứng.

Cuối cùng, lệnh gọi đơn giản là:

self._Booster = train(params, dmatrix, self.n_estimators, evals=evals,

early_stopping_rounds=early_stopping_rounds,

evals_result=evals_result, obj=obj, feval=feval,

verbose_eval=verbose)

Trong đó:

- *params (dict) – Booster params.*
- *dtrain (DMatrix) – Data để train.*
- *num_boost_round (int) – Số lần lặp của hàm.*
- *evals (list of pairs (DMatrix, string)) – danh sách của validation ước tính metrics trong quá trình train, Validation metrics giúp theo dõi hiệu xuất mô hình.*
- *obj (function) – Customized objective function.*

- *feval (function) – Customized evaluation function.*
- *maximize (bool) – Whether to maximize feval.*
- *early_stopping_rounds (int) dừng hàm tại lần lặp và trả về*
- *evals_result (dict) –lưu trữ kết quả đánh giá của danh sách các biến trước đó theo kiểu dict*

Điều này có nghĩa là mọi thứ có thể được thực hiện với XGBRegressor và XGBClassifier đều có thể thực hiện được thông qua hàm XGBoost.train. Ngược lại, điều đó rõ ràng là không đúng, chẳng hạn như một số tham số hữu ích của XGBoost.train không được hỗ trợ trong XGBModel API.

Điểm khác biệt khi áp dụng XGBoost.train:

XGBoost.train cho phép thiết lập các lệnh gọi lại được áp dụng ở cuối mỗi lần lặp.

XGBoost.train cho phép tiếp tục đào tạo thông qua tham số xgb_model.

XGBoost.train không chỉ cho phép giảm thiểu hàm eval mà còn cho phép tối đa hóa.

IV. Kết quả bộ đào tạo dữ liệu

4.1. Nhận xét và Đánh giá

Tổng bộ dữ liệu gồm 1200 mẫu được kết hợp và chia đều cho các vùng được chia ra 3 tập: Tập Train với 720 mẫu, tập Validation với 240 mẫu, tập Test với 240 mẫu.

a. Với mô hình RandomForests và các tham số
Với RandomForests chúng tôi chỉ sử dụng hai tệp train và test.
Sử dụng random_state = 9 trong mô hình RandomForestClassifier.
Cùng với đó là StratifiedKFold được dùng làm trình xác thực chéo K-Folds được phân tầng. Cung cấp các chỉ số đào tạo / kiểm tra để chia dữ liệu trong các tập hợp đào tạo/ kiểm tra.
Và GridSreachCV tìm kiếm hoàn chỉnh trên các giá trị tham số được chỉ định cho công cụ ước tính ta xét các tham số của RamdomForest gồm:

'n_estimators': [10, 100],
'max_features': ['auto', 'sqrt'],
'criterion' : ['gini','entropy'],

```
Accuracy score - Test dataset: 0.625531914893617
==========================================================================================
Score =
[0.76190476 0.39393939 0.27777778 0.89285714 0.64285714 0.71428571
 0.4375     0.55263158 0.94444444]
==========================================================================================
F1_board =
[[16.  0.  1.  2.  0.  0.  1.  0.  1.]
 [ 2. 13.  1.  0.  2.  7.  3.  4.  1.]
 [ 0.  1.  5.  1.  1.  0.  5.  5.  0.]
 [ 1.  0.  0. 25.  0.  0.  0.  1.  1.]
 [ 1.  2.  2.  0. 18.  0.  1.  4.  0.]
 [ 1.  5.  1.  0.  0. 25.  1.  2.  0.]
 [ 1.  1.  3.  0.  3.  0.  7.  1.  0.]
 [ 1.  3.  6.  1.  2.  1.  3. 21.  0.]
 [ 0.  0.  0.  1.  0.  0.  0.  0. 17.]]
==========================================================================================
```

Hình 4.1: Kết quả đánh giá dữ liệu trên model RandomForest

Với độ chính xác mô hình rơi vào khoảng: 62.6%. Trong đó nhãn phân biệt tốt nhất với tỷ lệ chính xác lên đến 94.44% là nước, Nước phân biệt cũng như có các chỉ số band khác so với phần còn lại. Các vùng khó phân biệt hơn đó là : 'Barren Land': 39.4% và 'Crop Land': 27.78%. Có thể là do khi chỉnh sửa trên Google Earth Pro không thể nhìn rõ phần đó có phải phần đất hai vùng trên không trong khi khó có thể nhìn rõ bằng mắt thường.

b. Với mô hình XGBoost
   Mô hình XGBoost sử dụng 3 tệp: train, validation và test.
   Cùng với đó các tham số của XGBoost được sử dụng gồm: 'max_depth': 2, 'eta': 0.3, 'objective': 'multi:softmax', 'nthread': 4, 'num_class': 9, 'shuffle': True.

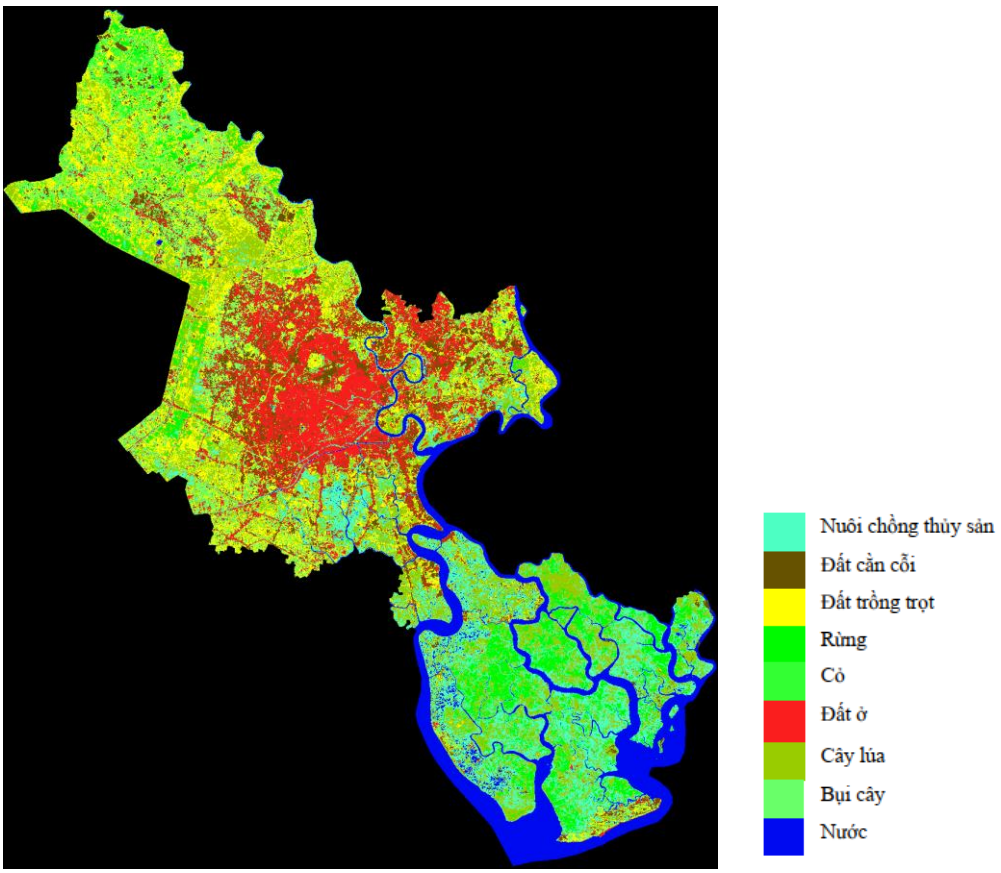
```
================================================================================
Prediction Accuracy: 62 %
================================================================================
Score =
[0.66666667 0.39393939 0.38888889 0.85714286 0.60714286 0.71428571
 0.5        0.57894737 0.94444444]
================================================================================
F1_board =
[[14.  3.  2.  1.  0.  0.  0.  0.  1.]
 [ 2. 13.  2.  0.  5.  5.  2.  4.  0.]
 [ 0.  1.  7.  0.  2.  0.  5.  3.  0.]
 [ 2.  0.  0. 24.  0.  0.  0.  1.  1.]
 [ 1.  3.  1.  0. 17.  0.  1.  5.  0.]
 [ 0.  4.  0.  0.  2. 25.  2.  2.  0.]
 [ 1.  3.  2.  0.  1.  0.  8.  1.  0.]
 [ 2.  3.  5.  0.  1.  2.  3. 22.  0.]
 [ 0.  0.  0.  1.  0.  0.  0.  0. 17.]]
================================================================================
```

Hình 4.2: Kết quả sau khi huấn luyện mô hình XGBoost

Với độ chính xác mô hình rơi vào khoảng: 62.9%. Trong đó nhãn phân biệt tốt nhất với tỷ lệ chính xác lên đến 94.44% là nước, cũng giống với mô hình trong RandomForests. Các vùng khó phân biệt hơn đó là : 'Barren Land': 39.4% và 'Crop Land': 38.89%. Tuy giống nhau về Barren Land nhưng trong nhãn CropLand thì XGBoost đã có sự cải thiện hơn so với RandomForests.

4.2. Hình ảnh nhận diện điểm trên bản đồ TP.Hồ Chí Minh

Sau khi huấn luyện model, được áp dụng để nhận diện hiển thị các vùng trên TP.Hồ Chí Minh



Hình 4.3: Bản đồ phân lớp phủ tại TP.HCM sử dụng mô hình XGBoost

Đường link github nơi chứa thông tin mà nhóm đã làm:

https://github.com/ndamtruong2k/TTNT-Artifical-Intelligence