


Máquinas de aprendizaje para análisis de datos geoespaciales en procesos de clasificación de cobertura terrestre e índices en el CDCol

- **Presentación**
- **¿Qué es el Data Cube (CDCol)?**
- **Estructura de datos**
- **Análisis de datos espaciales**
- **Máquinas de aprendizaje en coberturas terrestre**
- **Conclusiones**

Yilsey Benavides Miranda
Ingeniera Topográfica
Universidad Distrital Francisco José de Caldas

 **/yilsey-benavides**
/yilsey/



CUBO DE DATOS DE IMÁGENES DESATÉLITE PARA COLOMBIA CDCol



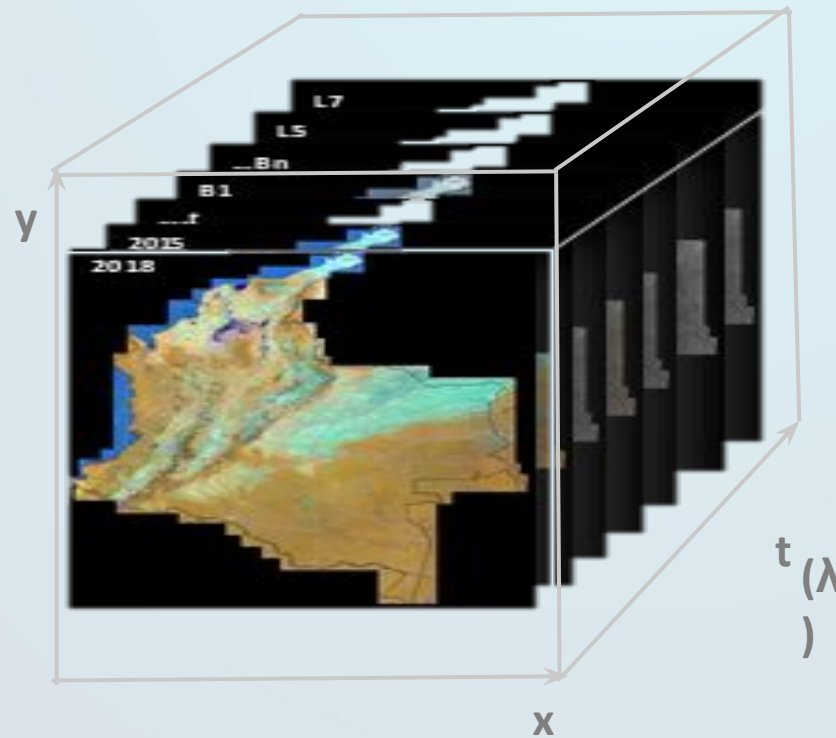
Colaboradores



¿Qué es el Data Cube (CDCol)?

Centralización y estandarización de datos– ARD (1460 escenas por año, 20 years)

Soporta diferentes insumos raster

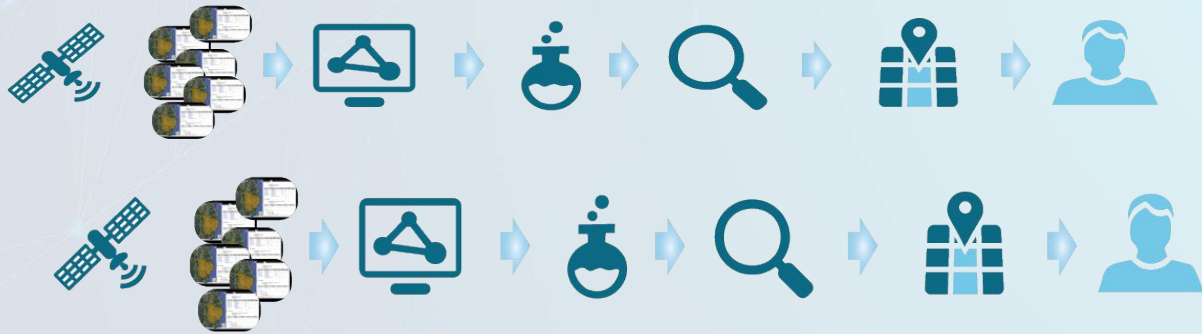


Base de datos multidimensional (x, y, t)

Optimización de procesos en tiempo

Análisis de datos

Estructura de datos



Proceso Tradicional

Analysis ready data ARD



CUBO DE DATOS DE IMÁGENES DESATÉLITE PARA COLOMBIA CDCol



Estructura de datos



Jupyter Notebook

Landsat-7 ETM+ Bands (µm)			Landsat-8 OLI and TIRS Bands (µm)		
			30 m Coastal/Aerosol	0.435 - 0.451	Band 1
Band 1	30 m Blue	0.441 - 0.514	30 m Blue	0.452 - 0.512	Band 2
Band 2	30 m Green	0.519 - 0.601	30 m Green	0.533 - 0.590	Band 3
Band 3	30 m Red	0.631 - 0.692	30 m Red	0.636 - 0.673	Band 4
Band 4	30 m NIR	0.772 - 0.898	30 m NIR	0.851 - 0.879	Band 5
Band 5	30 m SWIR-1	1.547 - 1.749	30 m SWIR-1	1.566 - 1.651	Band 6
Band 6	60 m TIR	10.31 - 12.36	100 m TIR-1	10.60 - 11.19	Band 10
			100 m TIR-2	11.50 - 12.51	Band 11
Band 7	30 m SWIR-2	2.064 - 2.345	30 m SWIR-2	2.107 - 2.294	Band 7
Band 8	15 m Pan	0.515 - 0.896	15 m Pan	0.503 - 0.676	Band 8
			30 m Cirrus	1.363 - 1.384	Band 9



UUID



Numpy
Xarray
Sklearn
Matplotlib
Gdal

xarr0

<xarray.Dataset>

Dimensions: (latitude: 3687, longitude: 3705, time: 113)

Coordinates:

* time (time) datetime64[ns] 2010-01-07T14:57:33 2010-01-07T14:57:57

...

* latitude (latitude) float64 5.0 5.0 4.999 4.999 4.999 4.999 4.998 ...

* longitude (longitude) float64 -73.0 -73.0 -73.0 -73.0 -73.0 -73.0 -73.0

...

Data variables:

blue (time, latitude, longitude) float64 nan nan nan nan nan nan ...

green (time, latitude, longitude) float64 nan nan nan nan nan nan ...

red (time, latitude, longitude) float64 nan nan nan nan nan nan ...

nir (time, latitude, longitude) float64 nan nan nan nan nan nan ...

swir1 (time, latitude, longitude) float64 nan nan nan nan nan nan ...

swir2 (time, latitude, longitude) float64 nan nan nan nan nan nan ...

Attributes:

crs: EPSG:4326

Estructura de datos

ACCESO A DATOS DE OBSERVACIÓN DE LA TIERRA LANDSAT

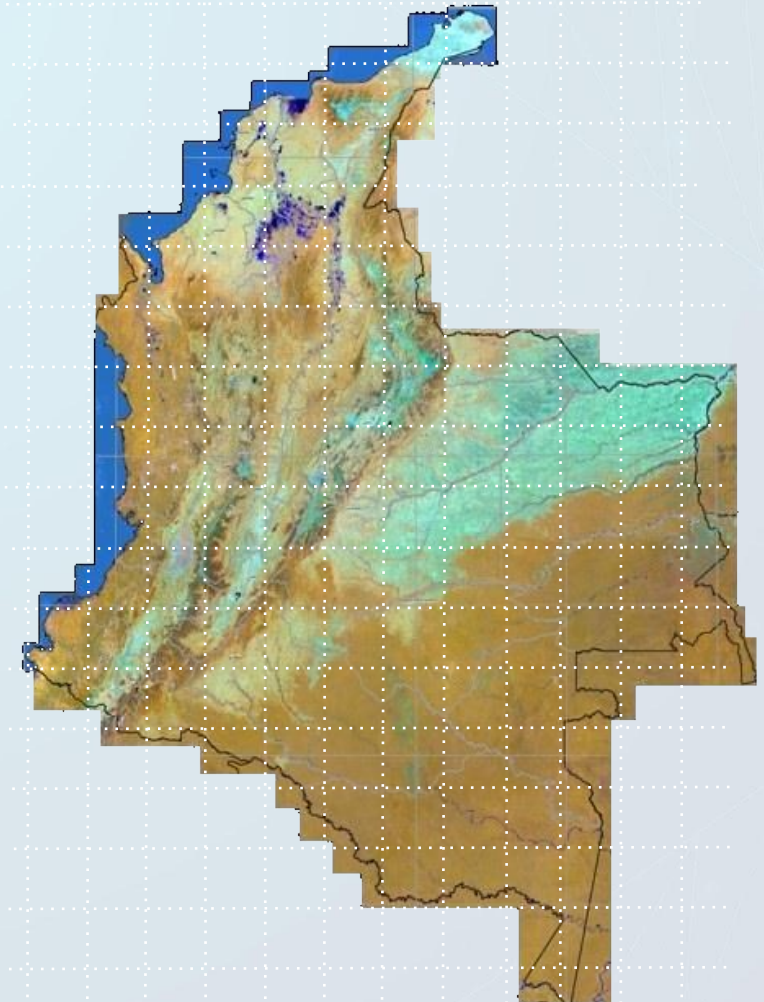


3 sensores
LANDSAT 5/7/8v

20 años
2000-2019

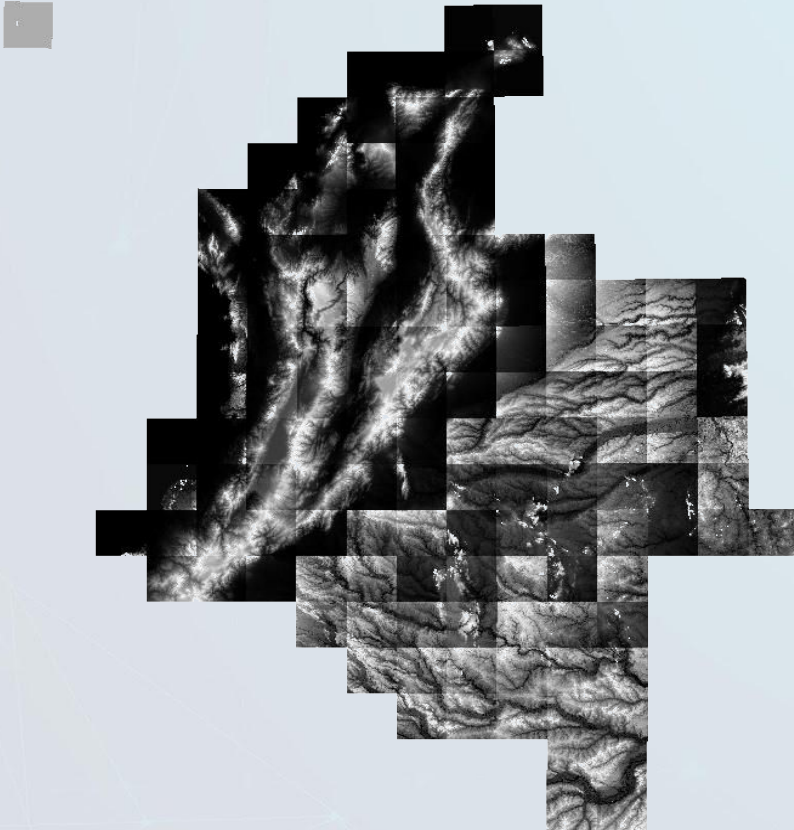
30 metros
RESOLUCION PIXEL

22,057 escenas ingestadas
LANDSAT 5/7/8

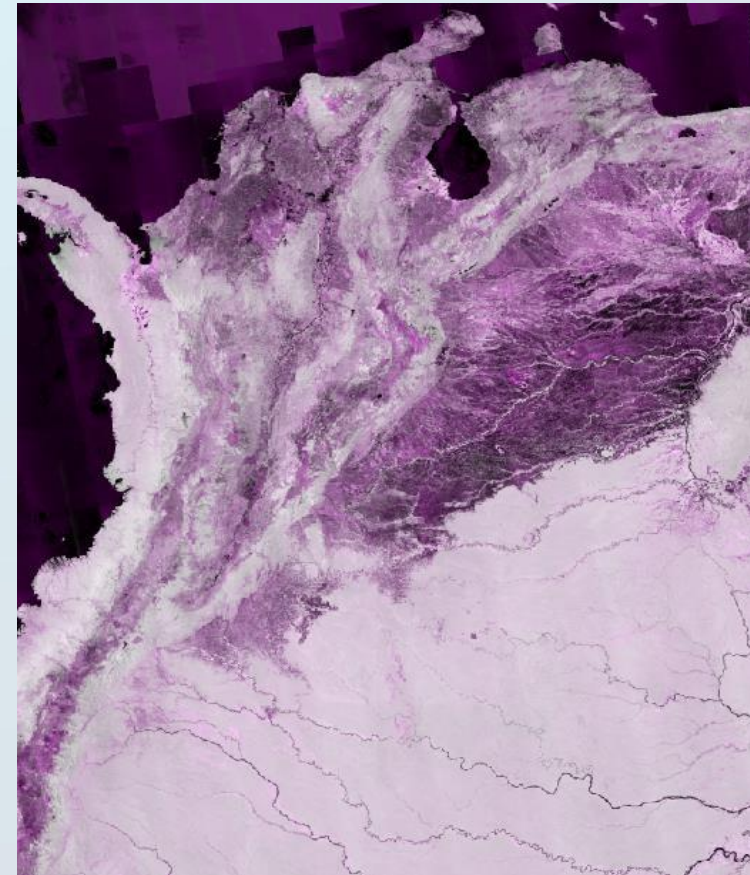


Estructura de datos

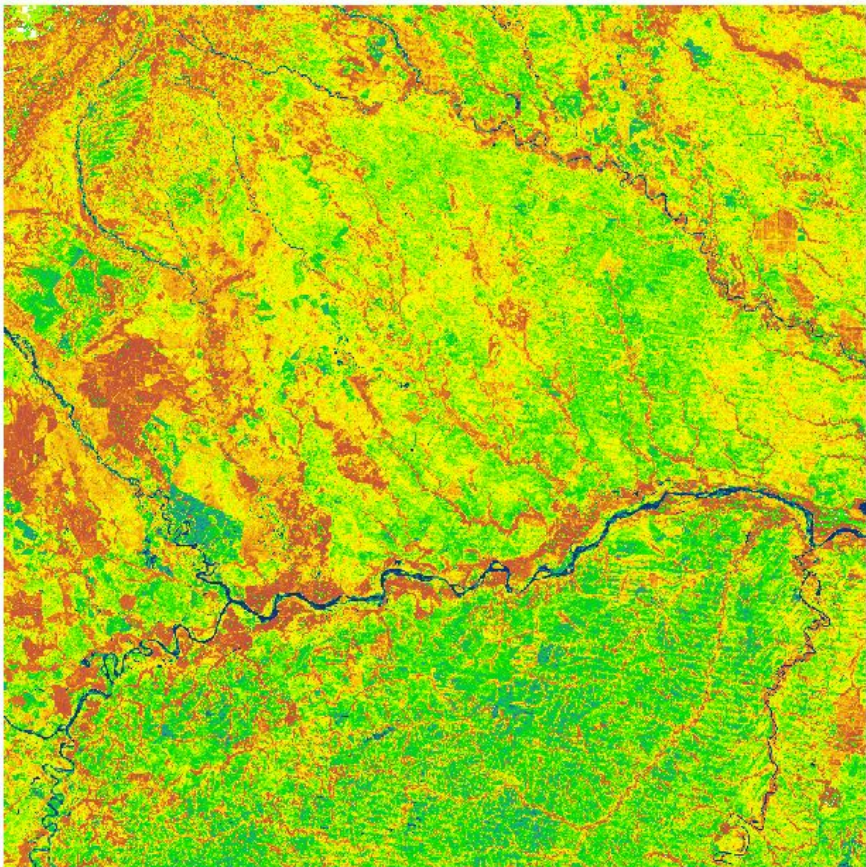
**ACCESO A DATOS DE
OBSERVACIÓN DE LA TIERRA
MOSAICO DSM Next Map World**



**ACCESO A DATOS DE
OBSERVACIÓN DE LA TIERRA
MOSAICO ALOS 2 PALSAR 2**

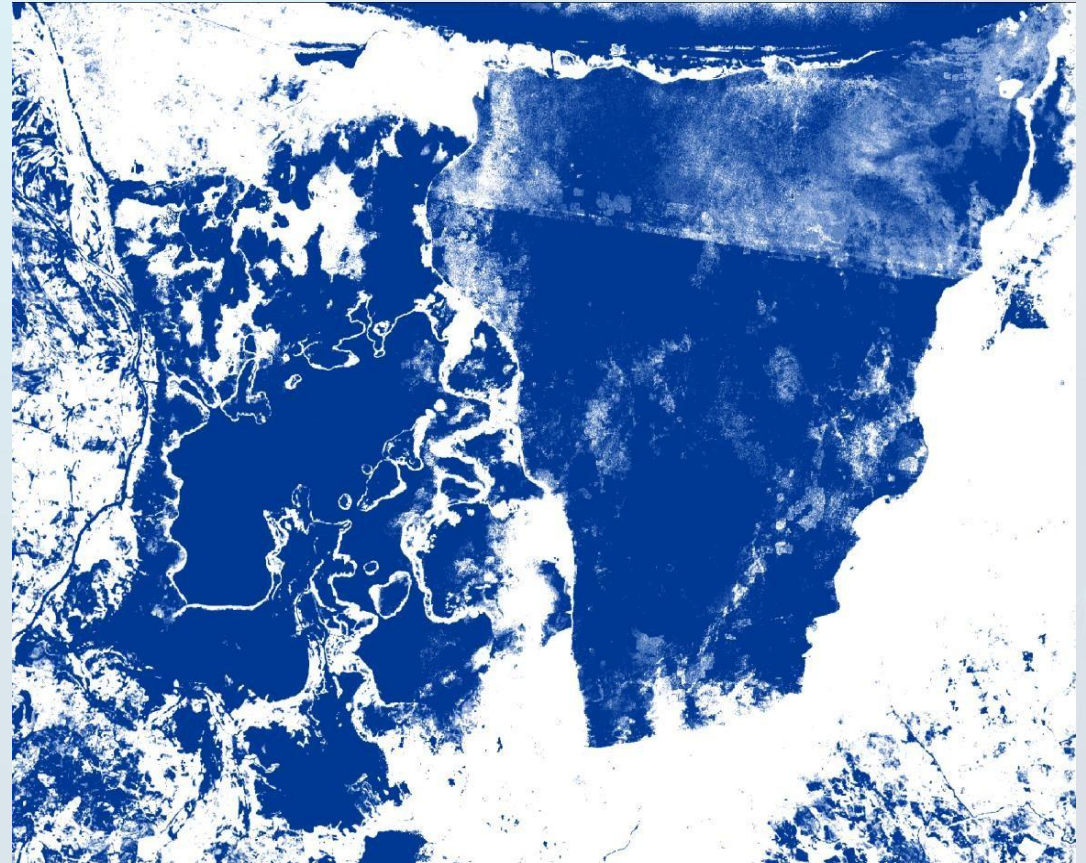


Índice de Normalizado de Vegetación NDVI



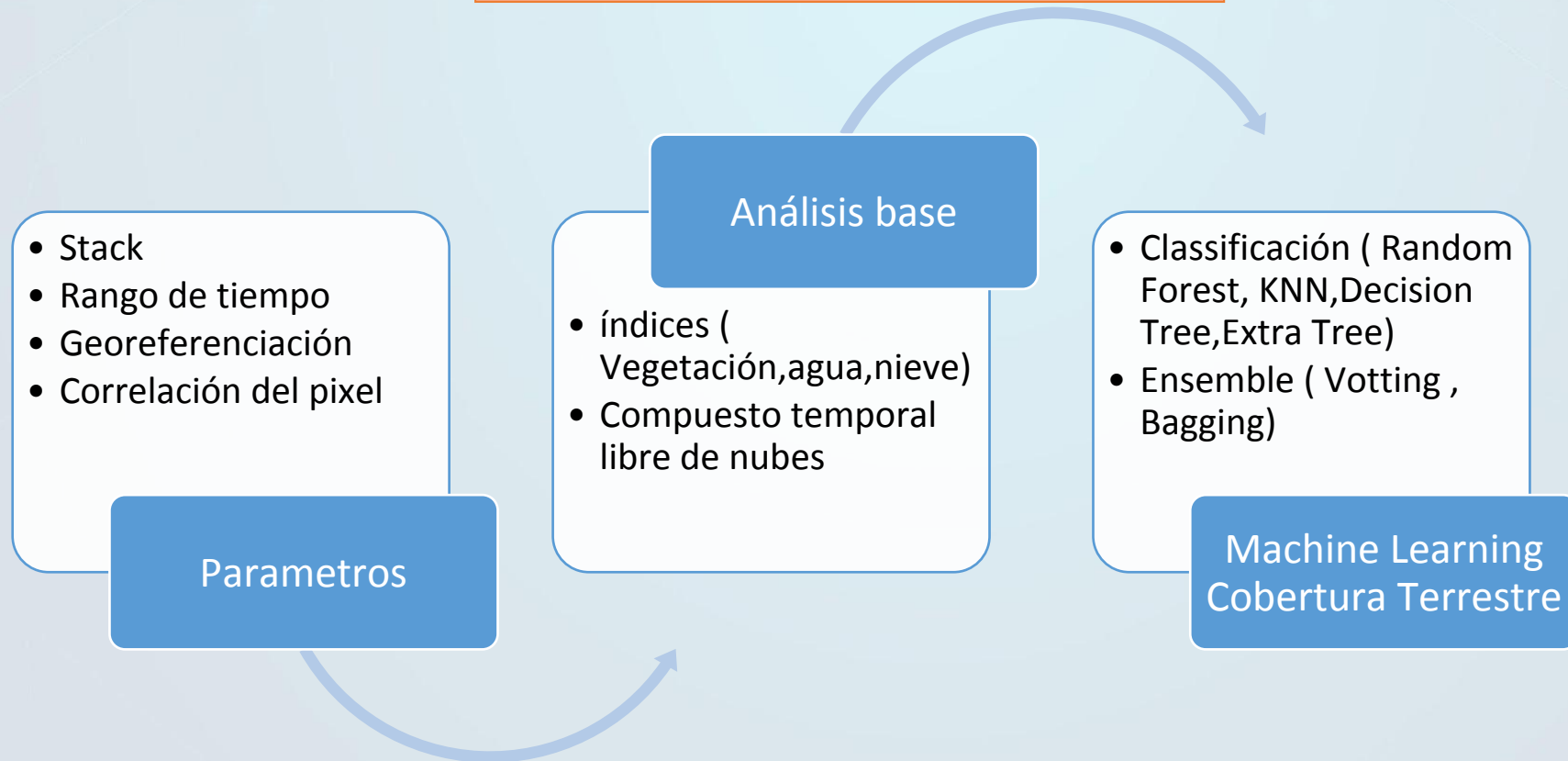
2010-2012

Índice Normalizado de cobertura de Agua NDWI



2000-2017

Análisis de datos espaciales



Análisis de datos espaciales

Imágenes de satélite

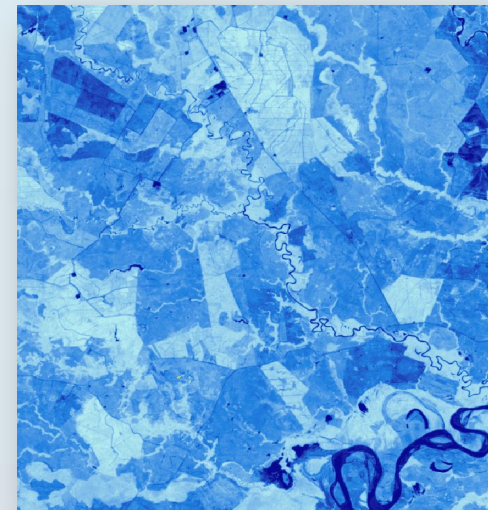
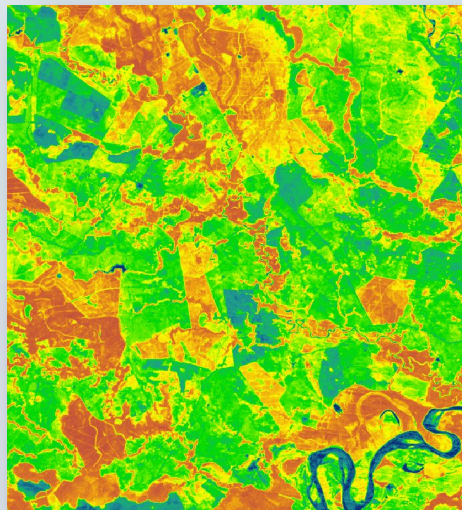


DEM – Digital elevation
model



Stack datos espaciales

```
medians = {}
for band in bands:
    datos = xarr0[band].values
    allNan = ~np.isnan(datos)
    if normalized:
        m = np.nanmean(datos.reshape((datos.shape[0], -1)), axis=1)
        st = np.nanstd(datos.reshape((datos.shape[0], -1)), axis=1)
        datos2 = ((datos - m[:, np.newaxis, np.newaxis])/st[:, np.newaxis,
        datos=(datos2)* np.nanmean(st) + np.nanmean(m)
    medians[band] = np.nanmedian(datos, 0)
    medians[band][np.sum(allNan, 0) < minValid] = np.nan
```



Análisis de datos espaciales

Datos de entrenamiento

Parámetros:

- Georeferencia
- Shapefile
- Index attribute field



```
files = [f for f in os.listdir(train_data_path) if f.endswith('.shp')]
classes = [f.split('.')[0] for f in files]
shapefiles = [os.path.join(train_data_path, f) for f in files if f.endswith(
shapefiles.sort()
shapefiles
```

```
['/home/cubo/jupyter/500/TRAIN/Bosque.shp',
'/home/cubo/jupyter/500/TRAIN/Cuerpos Agua.shp',
'/home/cubo/jupyter/500/TRAIN/Cultivos.shp',
'/home/cubo/jupyter/500/TRAIN/Pastos.shp',
'/home/cubo/jupyter/500/TRAIN/Zona_Pantanososa.shp']
```

```
labeled_pixels = rasterizar_entrenamiento(shapefiles, rows, cols, geo_transform,
```

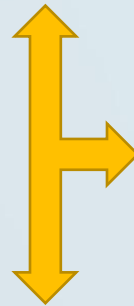
```
is_train = np.nonzero(labeled_pixels)
training_labels = labeled_pixels[is_train]
bands_data=[]
for band in bands:
    bands_data.append(medians[band])
bands_data = np.dstack(bands_data)
training_samples = bands_data[is_train]
rows, cols, n_bands = bands_data.shape
np.isfinite(training_samples)
_msk=np.sum(np.isfinite(training_samples),1)>1
training_samples= training_samples[_msk,:]
training_labels=training_labels[_msk]
#mascara valores nan por valor no data
mask_nan=np.isnan(training_samples)
training_samples[mask_nan]=-9999
```


Máquinas de aprendizaje en coberturas terrestre

Clasificación

Datos de entrenamiento

Stack datos espaciales



Decision Tree

SVM

Random Forest

Extra Tree

KNN

Neural Network

Temporal Median Composite (L7/L8)
Green-Red-NIR-Swir1-Swir2

Collecting training samples (shp)

CDCol WUI – Parameters

Classification model (.pkl model)

CDCol WUI – Algorithm execution

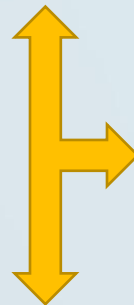
Geotiff

Thematic accuracy assessment

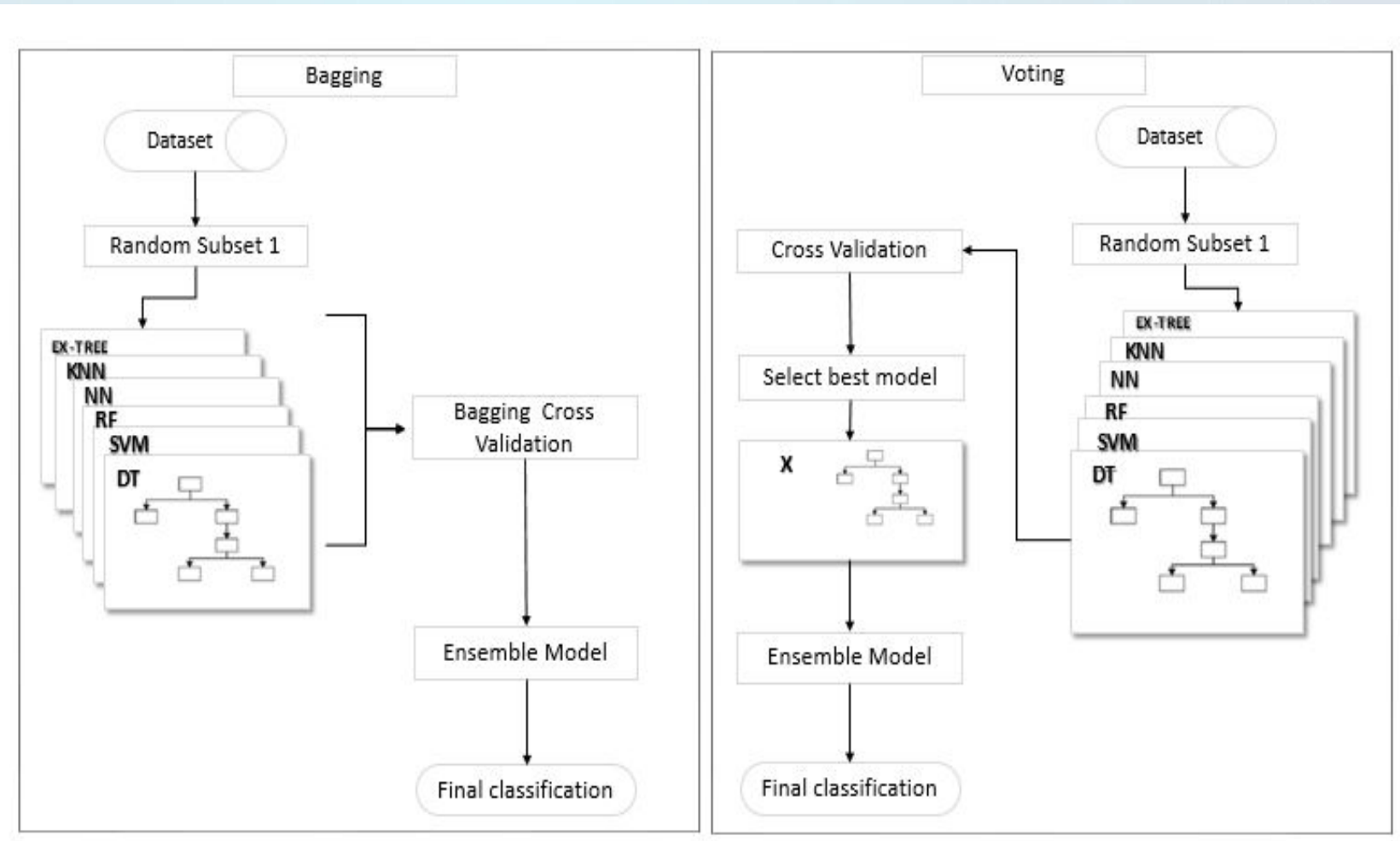
Thematic accuracy assessment

Máquinas de aprendizaje en coberturas terrestre

Datos de entrenamiento

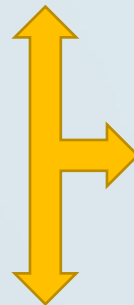


Stack datos espaciales



Máquinas de aprendizaje en coberturas terrestre

Datos de entrenamiento



Stack datos espaciales

Bagging Classifier

```
for clf in clf_array:
    vanilla_scores = cross_val_score(clf, training_samples, training_labels, cv=5)
    bagging_clf = BaggingClassifier(clf)
    bagging_scores = cross_val_score(bagging_clf, training_samples, training_labels, cv=5,
                                     n_jobs=-1)

    print ("Mean of: {1:.3f}, std: (+/-) {2:.3f} [{0}]"
           .format(clf.__class__.__name__,
                   vanilla_scores.mean(), vanilla_scores.std()))
    print ("Mean of: {1:.3f}, std: (+/-) {2:.3f} [Bagging {0}]\n"
           .format(clf.__class__.__name__,
                   bagging_scores.mean(), bagging_scores.std()))
```

Voting Classifier

```
clf_array=[dtree,svml,knn,nn,rf,extrat]
ecf = VotingClassifier(estimators=[('Decision Tree', dtree), ('SVC', svml), ('KNN', knn)])

for clf_array, label in zip([dtree,svml,knn,nn,rf,extrat,ecf], ['Random Forest', 'SVC', 'KNN', 'NN', 'Decision Tree', 'Voting Classifier']):
    scores = cross_val_score(clf_array, training_samples, training_labels, cv=5)
    print("Accuracy: %0.3f ( %0.3f) [%s]" % (scores.mean(), scores.std(), label))
```

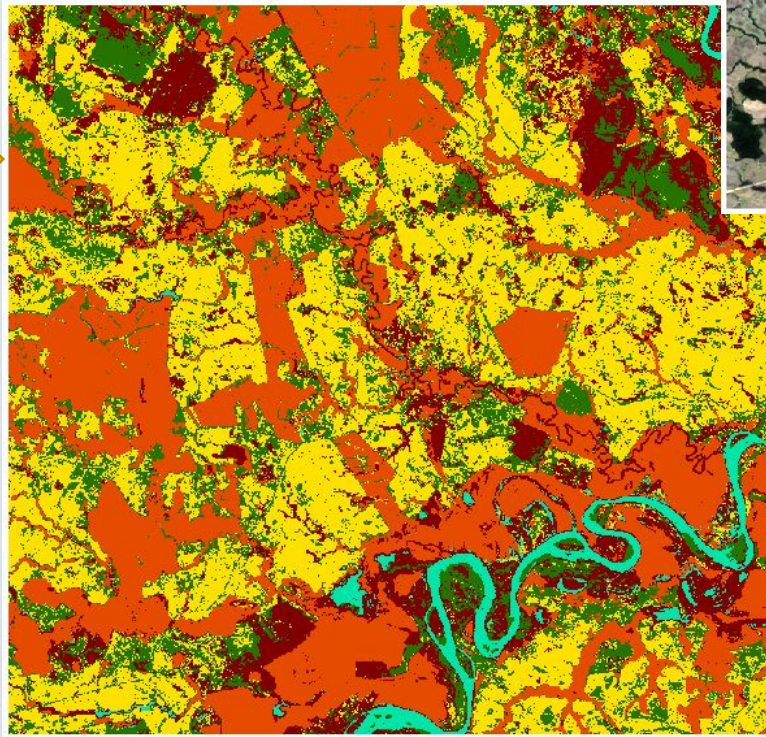
Accuracy: 0.699 (0.031) [Random Forest]
 Accuracy: 0.745 (0.055) [Decision Tree]
 Accuracy: 0.716 (0.025) [SVC]
 Accuracy: 0.286 (0.083) [KNN]

Bagging

Voting

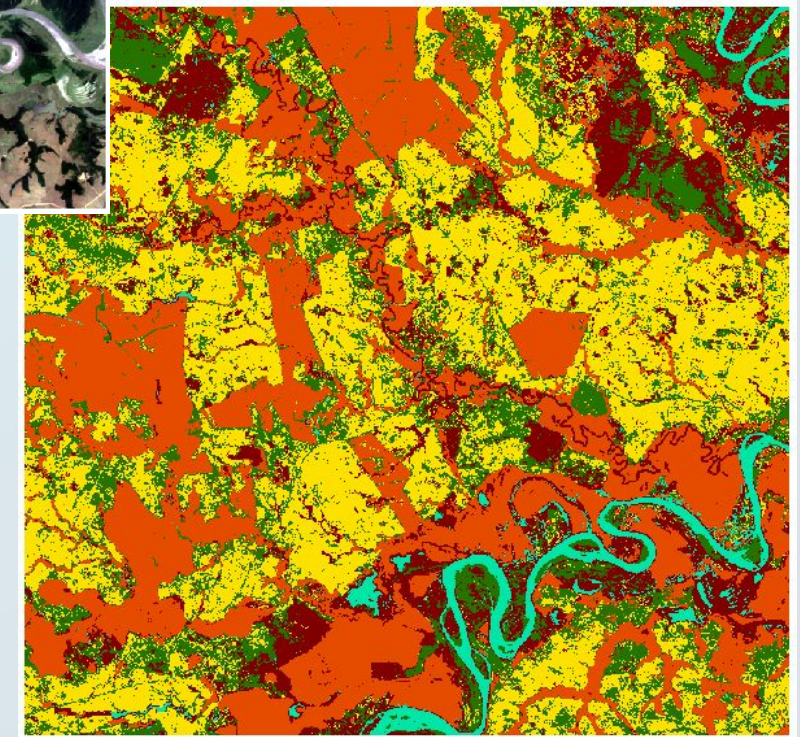
Validación
cruzada

Validación
temática



83.55

Área de estudio
CDCol



83.02

Máquinas de aprendizaje en coberturas terrestre

Validación cruzada

Bagging Classifier

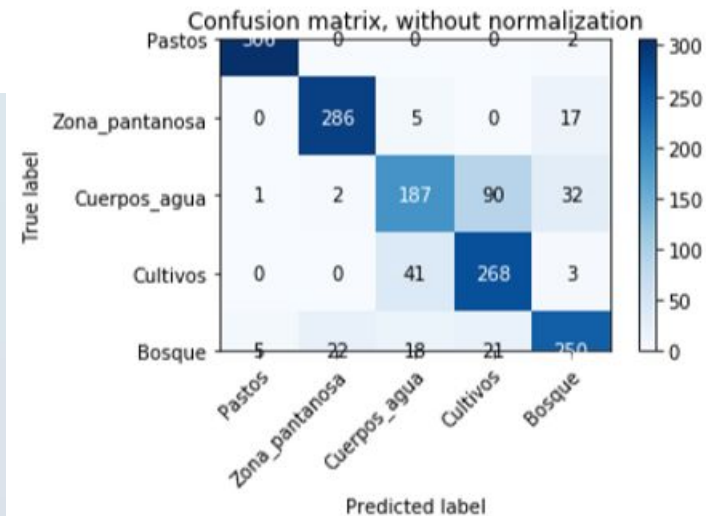
Voting Classifier

Validación temática

```
plot_confusion_matrix(verification_labels, predicted_labels, classes=classe,
                      title='Confusion matrix, without normalization')
```

Confusion matrix, without normalization

```
[[306  0  0  0  2]
 [  0 286  5  0 17]
 [  1  2 187 90 32]
 [  0  0 41 268  3]
 [  5 22 18 21 250]]
```



```
###Exactitud
vrf = 0
for i in list(range(0,len(mconfu))):
    for j in list(range(0,len(mconfu))):
        vrf += mconfu[i][j]
    #print(mconfu[d][d])
diag = 0
for d in list(range(0,len(mconfu))):
    #for j in list(range(0,len(mconfu))):
        diag += mconfu[d][d]
    #print(mconfu[d][d])
```

```
(diag/vrf)*100
```

83.3547557840617

CUBO DE DATOS DE IMÁGENES DESATÉLITE PARA COLOMBIA CDCol



Validación cruzada

Máquinas de aprendizaje en coberturas terrestre

Validación temática

Bagging							
.	Bosque	Cuerpos Agua	Cultivos	Pastos	Zona Pantanosa	Total	ACC User
Bosque	304	0	0	0	4	308	98,7%
Cuerpos Agua	0	285	4	0	19	308	92,5%
Cultivos	0	0	182	100	30	312	58,3%
Pastos	0	0	33	273	6	312	87,5%
Zona Pantanosa	7	19	18	18	254	316	80,4%
Total	311	304	237	391	313		
ACC Producer	97,7%	93,8%	76,8%	69,8%	81,2%		

Voting							
.	Bosque	Cuerpos Agua	Cultivos	Pastos	Zona Pantanosa	Total	ACC User
Bosque	306	0	0	0	2	308	99,4%
Cuerpos Agua	0	286	5	0	17	308	92,9%
Cultivos	1	1	190	85	35	312	60,9%
Pastos	0	0	44	264	4	312	84,6%
Zona Pantanosa	4	21	19	18	254	316	80,4%
Total	311	308	258	367	312	1556	
ACC Producer	98,4%	92,9%	73,6%	71,9%	81,4%		100,0%

INTERFAZ DE USUARIO



Conclusiones

- La región se caracteriza por una diversidad de ecosistemas y cobertura terrestre(Bosque,sabana,páramos)
- El CDCol permite la implementación de máquinas de aprendizaje para clasificación de cobertura terrestre empleando varios insumos en diferentes rangos de tiempo, generando salidas con una exactitud temática considerable en procesos de generación masiva de productos.
- Mejoras en el tiempo de procesamiento de datos espaciales y la generación de mapas de cobertura terrestre ejecutadas en áreas extensas y a nivel nacional.
- Detección cobertura terrestre empleando índices para identificación de coberturas.

CUBO DE DATOS DE IMÁGENES DESATÉLITE PARA COLOMBIA CDCol

Gracias

yilmiranda@gmail.com
cuboimagenes@ideam.gov.co

