WEBINAR

Recomendaciones para abordar el Desafio AgTech utilizando PYTHON

Martes 10 /11 - 9hs

DESAFÍOS AGTECH

www.desafiosagtech.com

Alfredo Campos

Desarrollador-Investigador-Docente

VISTAGUAY

Matba Rofex · Bolsa de Cereales · Bolsa de Comercio de Rosario

DESAFÍOS AGTECH

Qué es una clasificación y para qué clasificar.

Imágenes satelitales. Fortalezas y Debilidades.

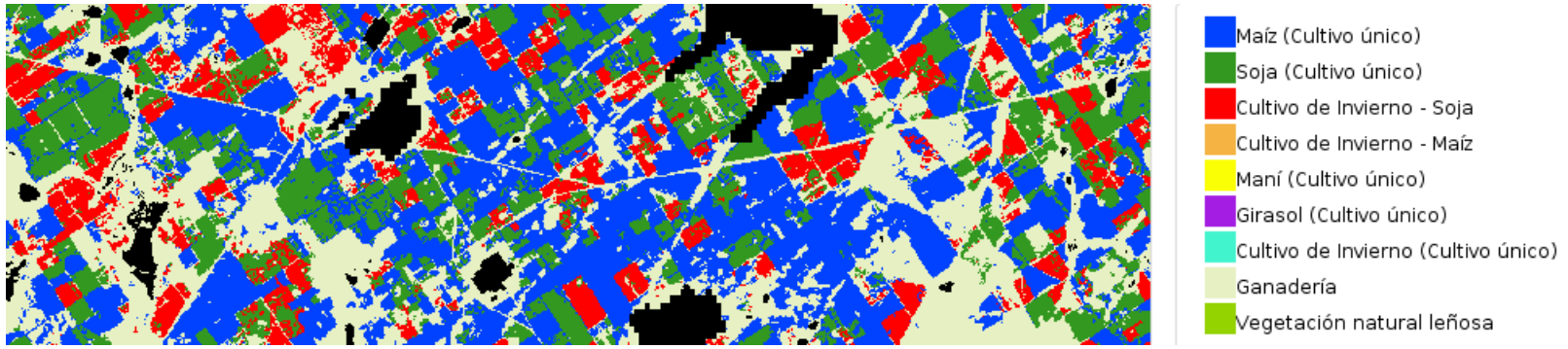
Cómo clasificar con imágenes satelitales.

Desafío AgTech. Dataset. Restricciones.

Formas de encarar el desafío con Python.

DESAFÍOS AGTECH

QUÉ SON?



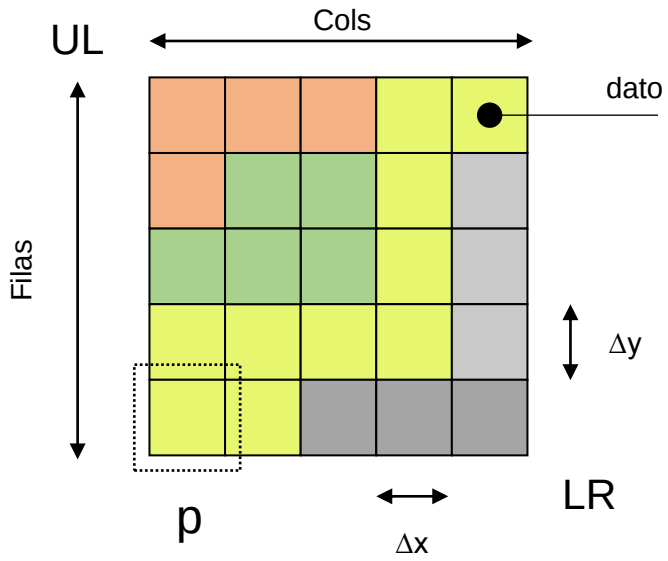
http://www.geointa.inta.gob.ar/2019/09/10/mapa-nacional-de-cultivos-campana-20182019/

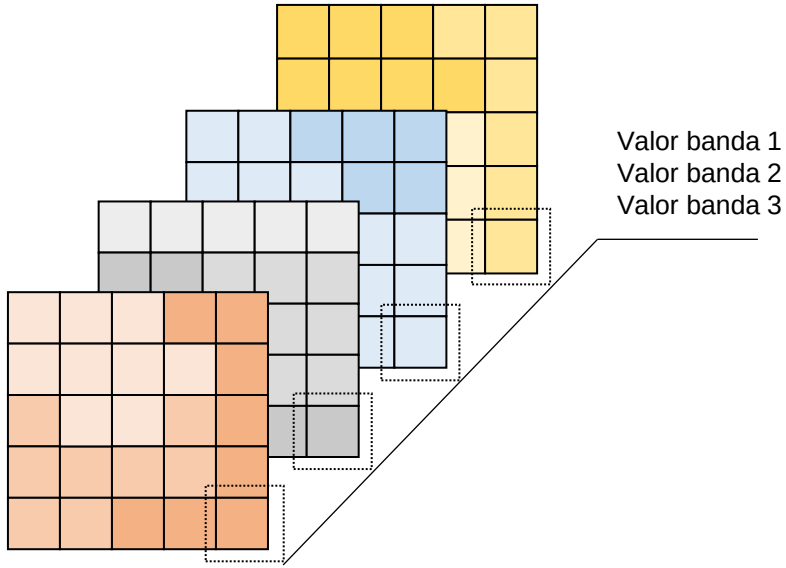
PARA QUÉ SE USAN?

- Optimización de políticas públicas y privadas.
- Cuidado del ambiente.
- 1er paso para estimar la cosecha.

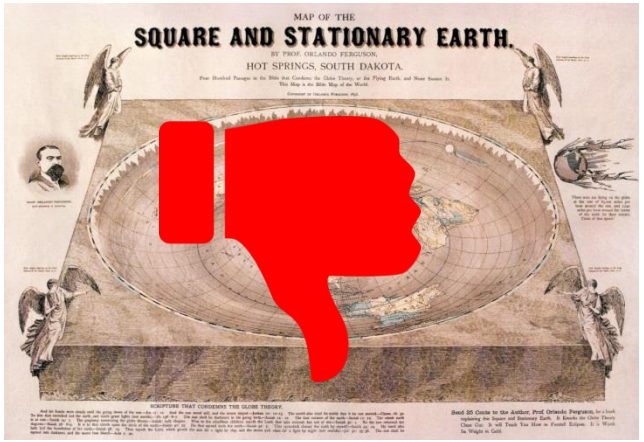
Bondades de las imágenes satelitales

1. Objetiva
2. Metódica
3. Periódica (- nubes)
4. Documento
5. Queda registrada (veo el pasado!)
6. Ven más que el ojo humano

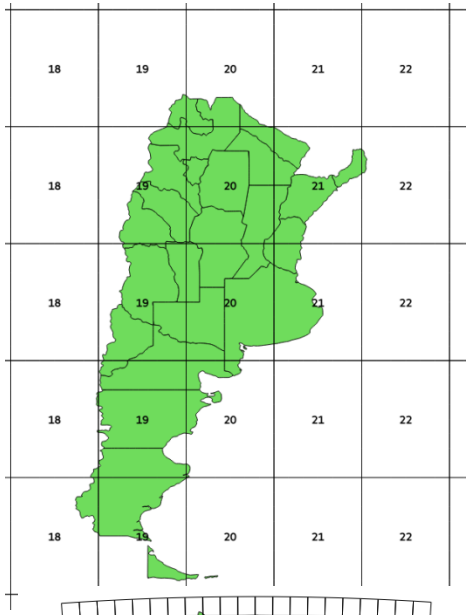

UL
Cols
dato
Filas
Δy
p
LR
Δx

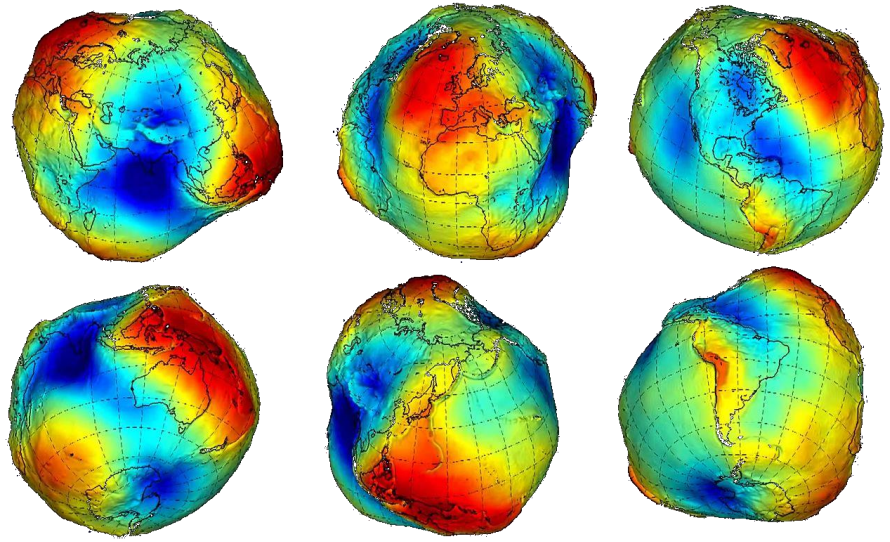

Valor banda 1
Valor banda 2
Valor banda 3

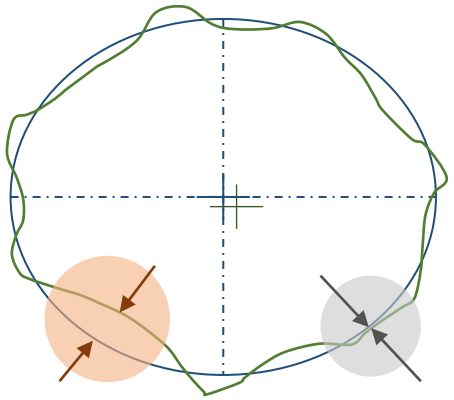
IMÁGENES SATELITALES. CUIDADO CON LOS SISTEMAS DE REFERENCIA.



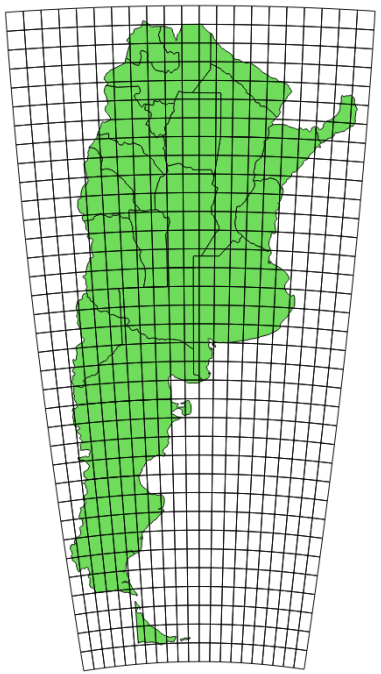
Lat Lon – WGS 84
(EPSG:4326)



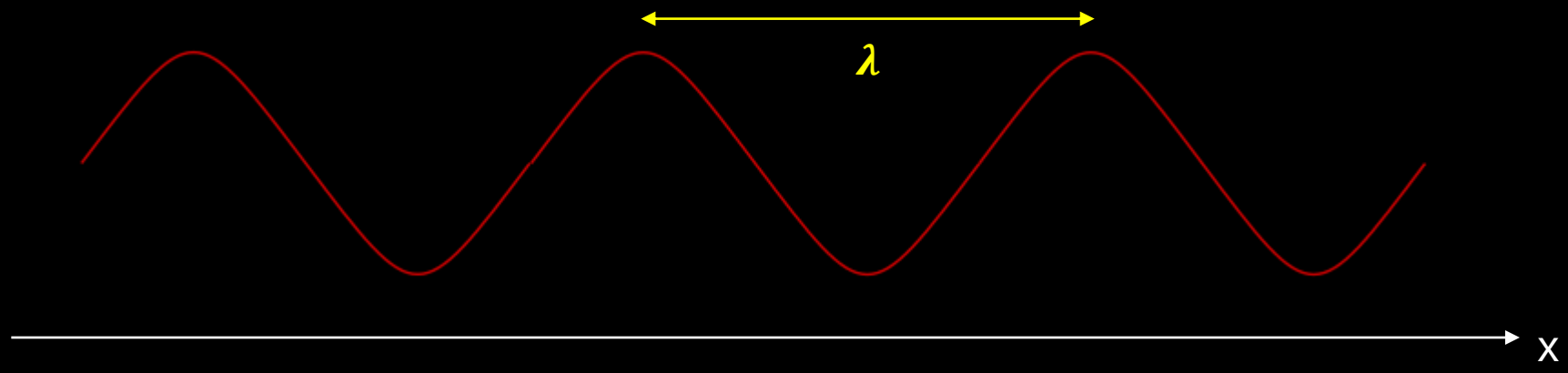


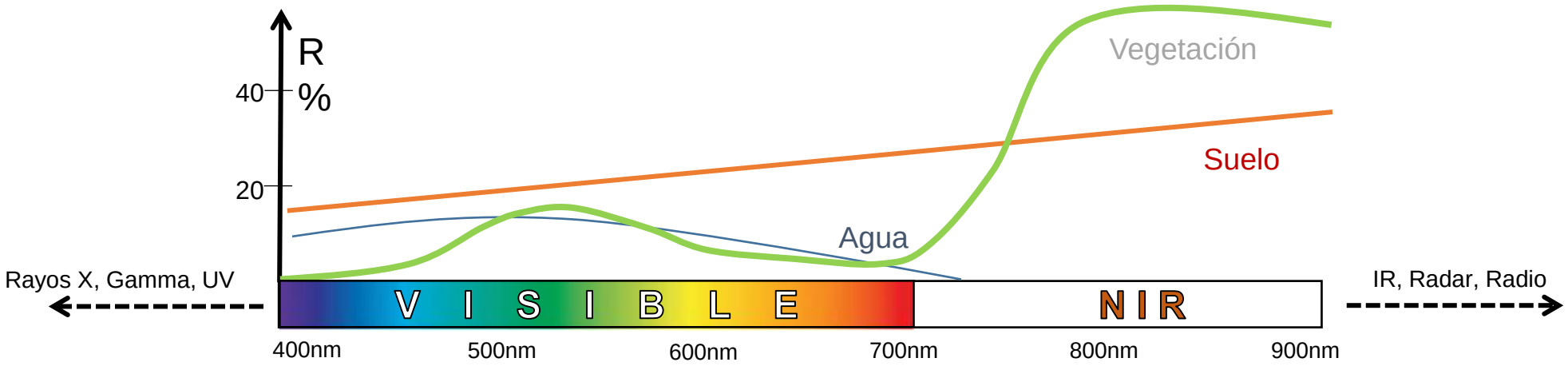


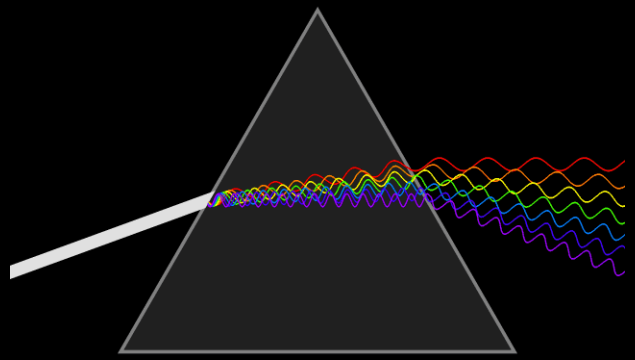
UTM Zona 20S – WGS 84
(EPSG:32720)

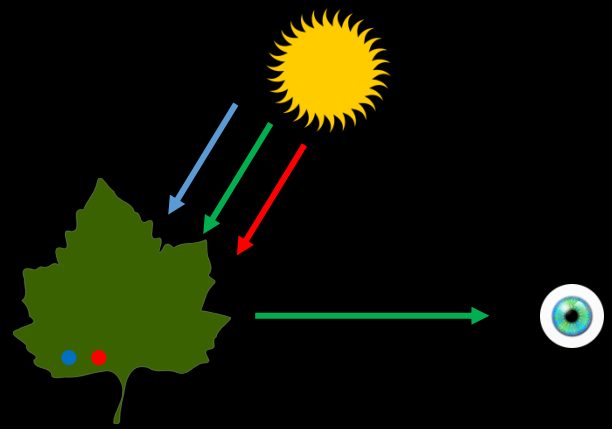


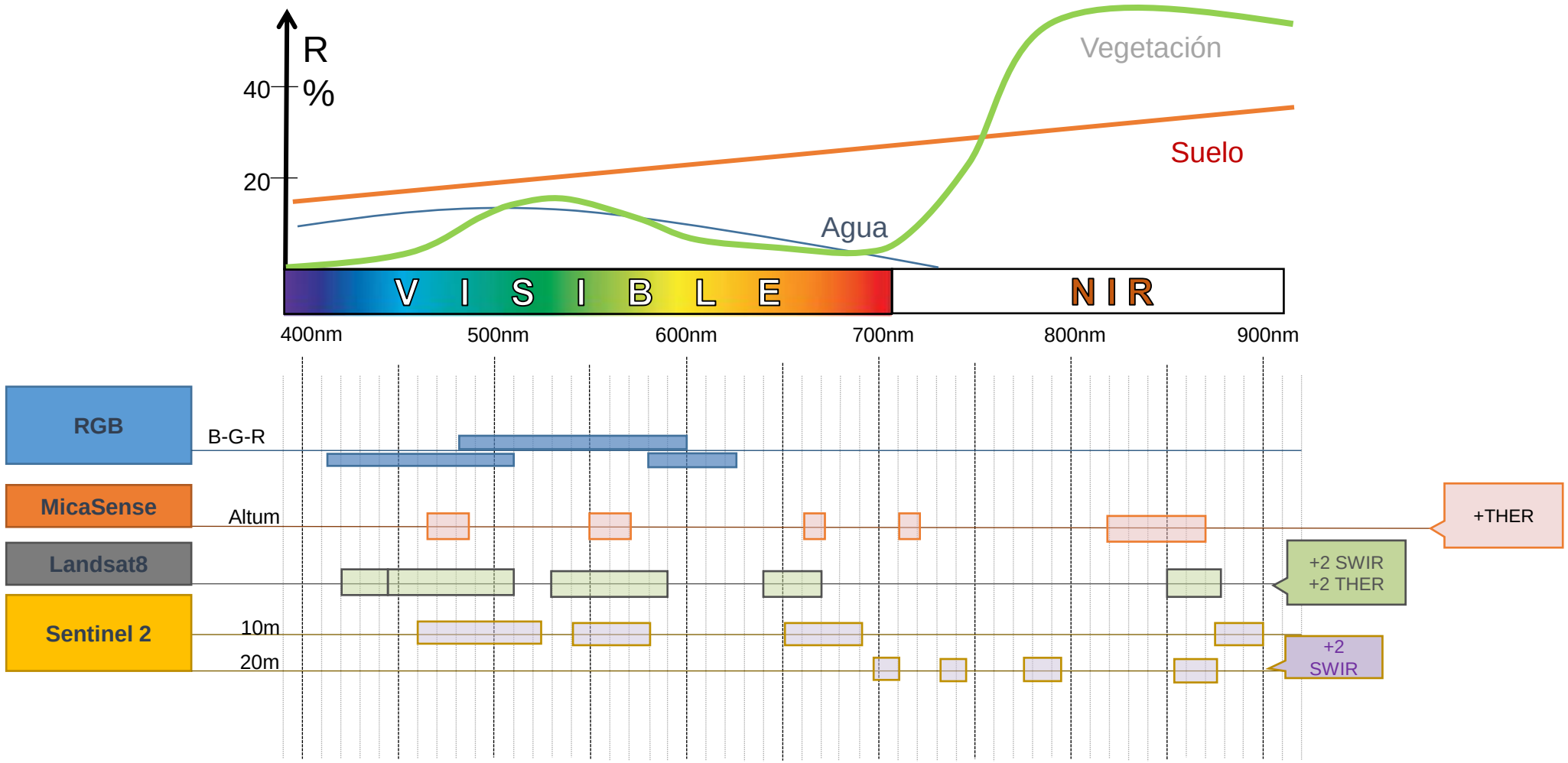

V I S I B L E
400nm
500nm
600nm
700nm

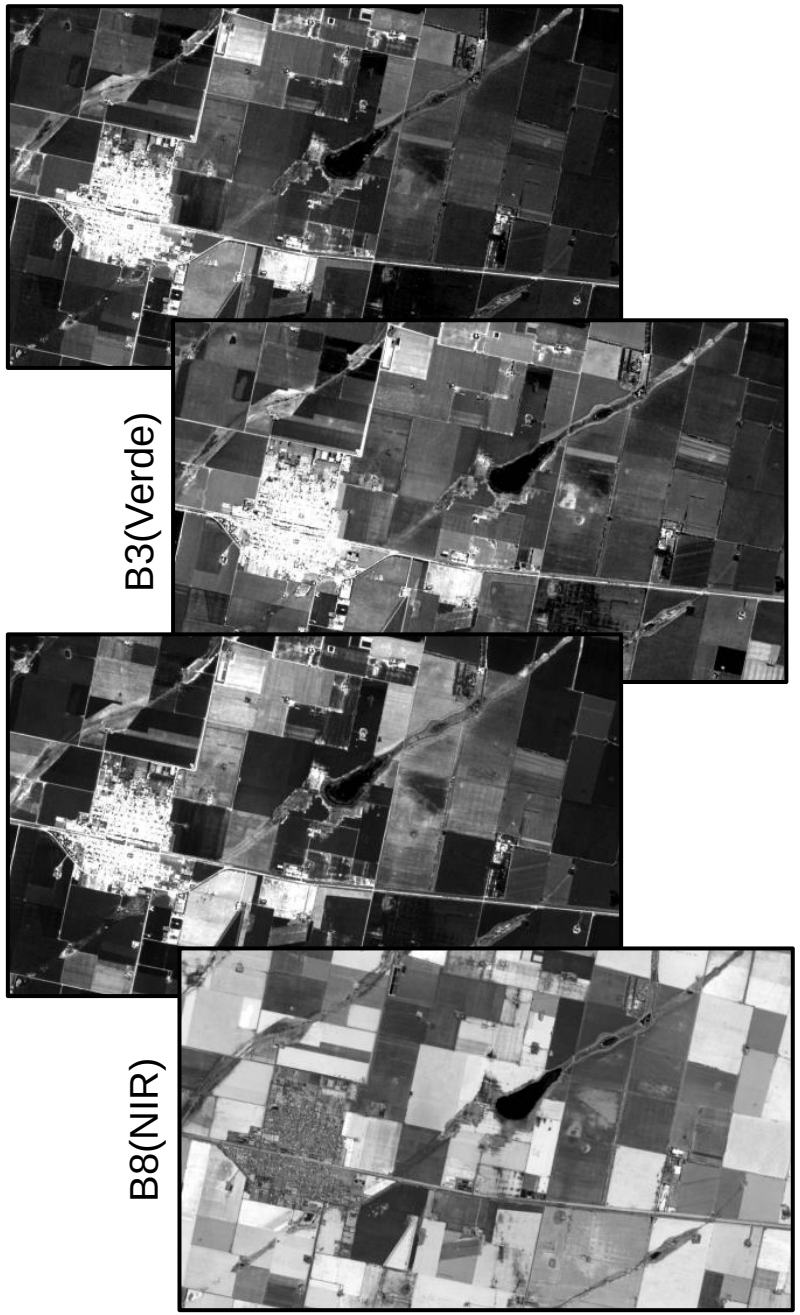

λ
x

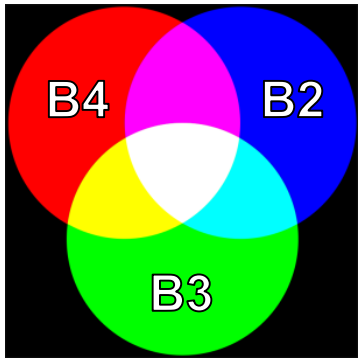

R
%
40
20
Vegetación
Suelo
Agua
Rayos X, Gamma, UV
V I S I B L E
NIR
IR, Radar, Radio
400nm
500nm
600nm
700nm
800nm
900nm






R
%
40
20
Vegetación
Suelo
Agua
V I S I B L E
NIR
400nm
500nm
600nm
700nm
800nm
900nm
RGB
B-G-R
MicaSense
Altum
+THER
Landsat8
+2 SWIR
+2 THER
Sentinel 2
10m
20m
+2
SWIR


B2(Azul)
B3(Verde)
B4(Rojo)
B8(NIR)

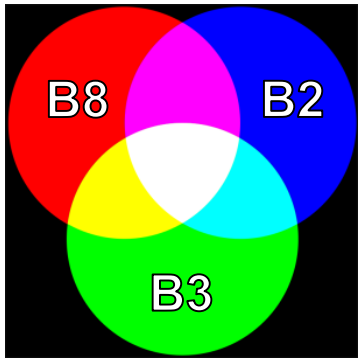

B4
B2
B3

B2(Azul)

B3(Verde)

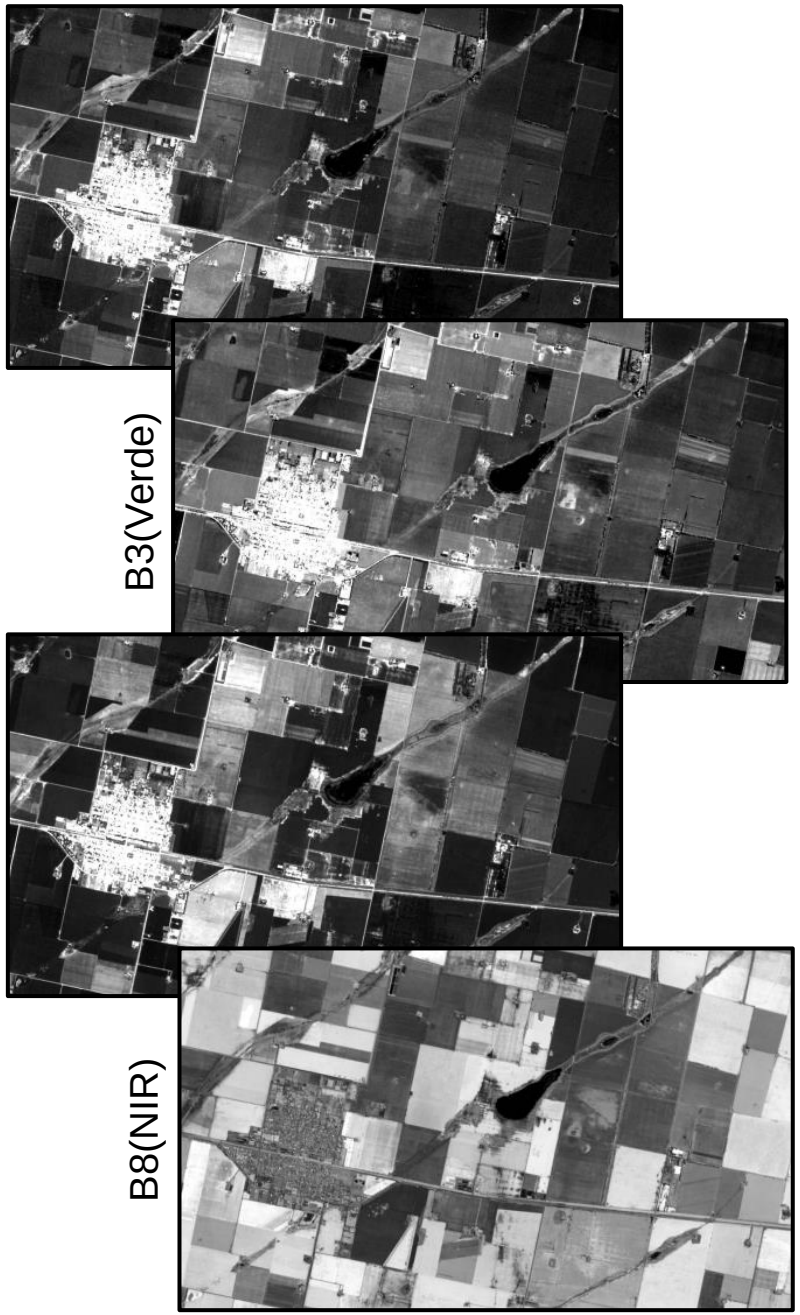
B4(Rojo)

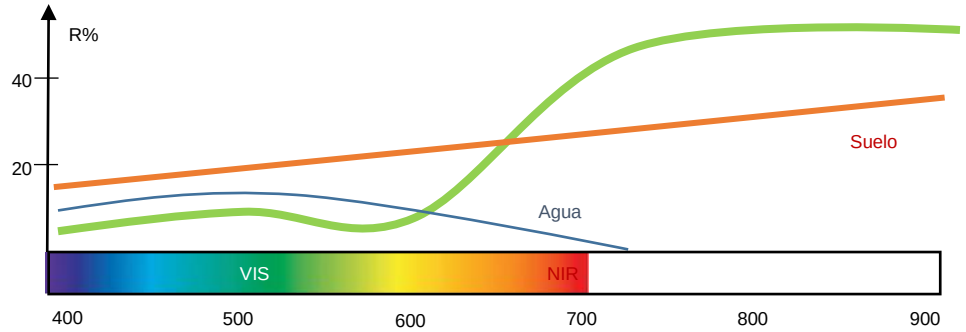
B8(NIR)



IMÁGENES SATELITALES. QUÉ VEMOS EN LAS IMÁGENES

DESAFÍOS AGTECH





$$NDVI = \frac{NIR - Rojo}{NIR + Rojo}$$

DESAFÍOS AGTECH

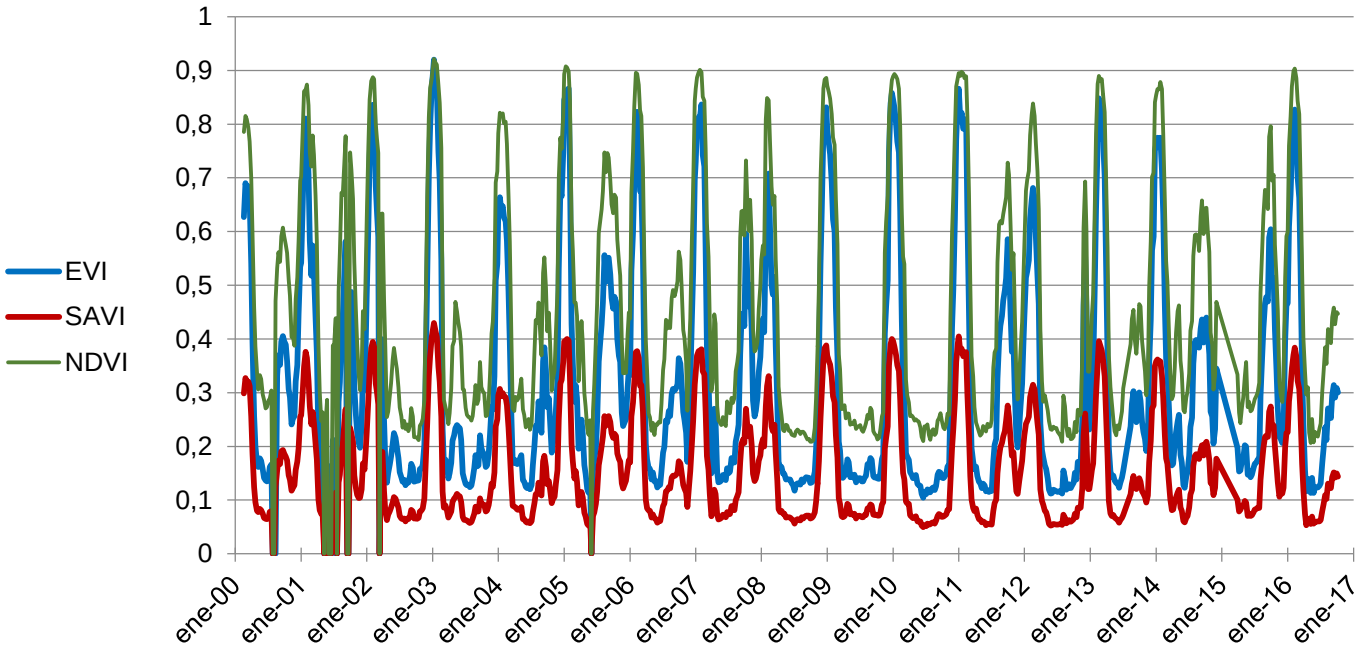
$$NDVI = \frac{NIR - R}{NIR + R} \qquad EVI = \frac{2.5(NIR - R)}{NIR + 6R - 7.5B + 1} \qquad SAVI = \frac{NIR - R}{NIR + R + L}.(1 + L)$$

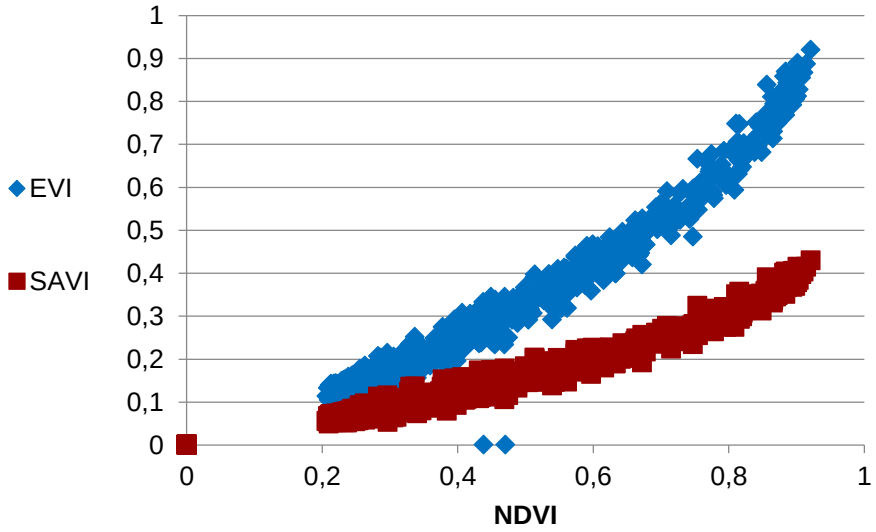
$$TDVI = \sqrt{0.5 + \frac{NIR - R}{NIR + R}} \qquad GNDVI = \frac{NIR - G}{NIR + G} \qquad LAI = 3.618 \ .EVI - 0.118 \qquad SR = \frac{NIR}{R}$$

$$OSAVI = \frac{1.5(NIR - R)}{NIR + R + 0.16} \qquad GSAVI = \frac{NIR - G}{NIR + G + L}.(1 + L)$$

$$VARI = \frac{G - R}{G + R - B} \qquad GRVI = \frac{NIR}{G} \qquad DVI = NIR - R \qquad ENDVI = \frac{NIR + G - 2B}{NIR + G + 2B}$$

$$GDVI = NIR - G \qquad IPVI = \frac{NIR}{NIR + R} \qquad NGRDI = \frac{G - R}{G + R}$$

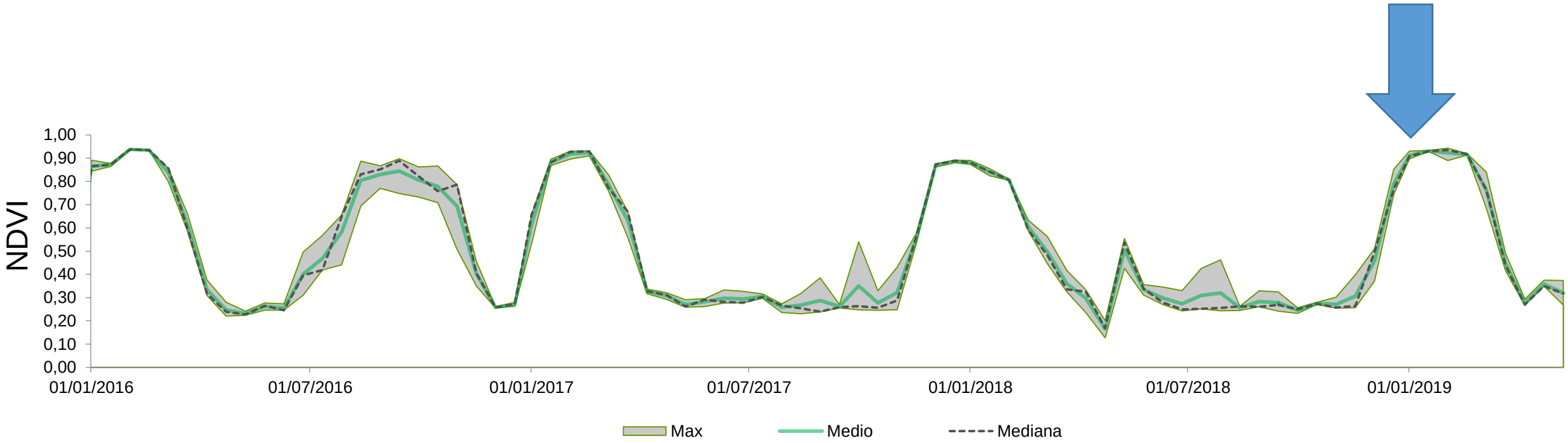




DESAFÍOS AGTECH

Landsat 8	Sentinel II	Planet	Drone
• píxel: 30m • img c/16 días • 11 bandas • $=0*	• Pixel 10m • img c/10 días • 13 bandas • $=0*	• Pixel 3m • img c/1 día • 4 bandas • $ > 0	• Pixel < 5cm • img c/visita • 3-5 bandas • $↑

COMO CLASIFICAR CON IMÁGENES SATELITALES. FECHA.



- INSPECCIÓN VISUAL
- CLASIFICACIÓN NO SUPERVISADA (ej: Kmeans)
- CLASIFICACIÓN SUPERVISADA (ej: SVM, NN, Decision Trees)

metrics.accuracy_score(y_true, y_pred, *[, ...])	Accuracy classification score.
metrics.auc(x, y)	Compute Area Under the Curve (AUC) using the trapezoidal rule
metrics.average_precision_score(y_true, ...)	Compute average precision (AP) from prediction scores
metrics.balanced_accuracy_score(y_true, ...)	Compute the balanced accuracy
metrics.brier_score_loss(y_true, y_prob, *)	Compute the Brier score.
metrics.classification_report(y_true, y_pred, *)	Build a text report showing the main classification metrics.
metrics.cohen_kappa_score(y1, y2, *[, ...])	Cohen's kappa: a statistic that measures inter-annotator agreement.
metrics.confusion_matrix(y_true, y_pred, *)	Compute confusion matrix to evaluate the accuracy of a classification.
metrics.dcg_score(y_true, y_score, *[, k, ...])	Compute Discounted Cumulative Gain.
metrics.f1_score(y_true, y_pred, *[, ...])	Compute the F1 score, also known as balanced F-score or F-measure
metrics.fbeta_score(y_true, y_pred, *, beta)	Compute the F-beta score
metrics.hamming_loss(y_true, y_pred, *[, ...])	Compute the average Hamming loss.
metrics.hinge_loss(y_true, pred_decision, *)	Average hinge loss (non-regularized)
metrics.jaccard_score(y_true, y_pred, *[, ...])	Jaccard similarity coefficient score
metrics.log_loss(y_true, y_pred, *[, eps, ...])	Log loss, aka logistic loss or cross-entropy loss.
metrics.matthews_corrcoef(y_true, y_pred, *)	Compute the Matthews correlation coefficient (MCC)
metrics.multilabel_confusion_matrix(y_true, ...)	Compute a confusion matrix for each class or sample
metrics.ndcg_score(y_true, y_score, *[, k, ...])	Compute Normalized Discounted Cumulative Gain.
metrics.precision_recall_curve(y_true, ...)	Compute precision-recall pairs for different probability thresholds
metrics.precision_recall_fscore_support(...)	Compute precision, recall, F-measure and support for each class
metrics.precision_score(y_true, y_pred, *[, ...])	Compute the precision
metrics.recall_score(y_true, y_pred, *[, ...])	Compute the recall
metrics.roc_auc_score(y_true, y_score, *[, ...])	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
metrics.roc_curve(y_true, y_score, *[, ...])	Compute Receiver operating characteristic (ROC)
metrics.zero_one_loss(y_true, y_pred, *[, ...])	Zero-one classification loss.

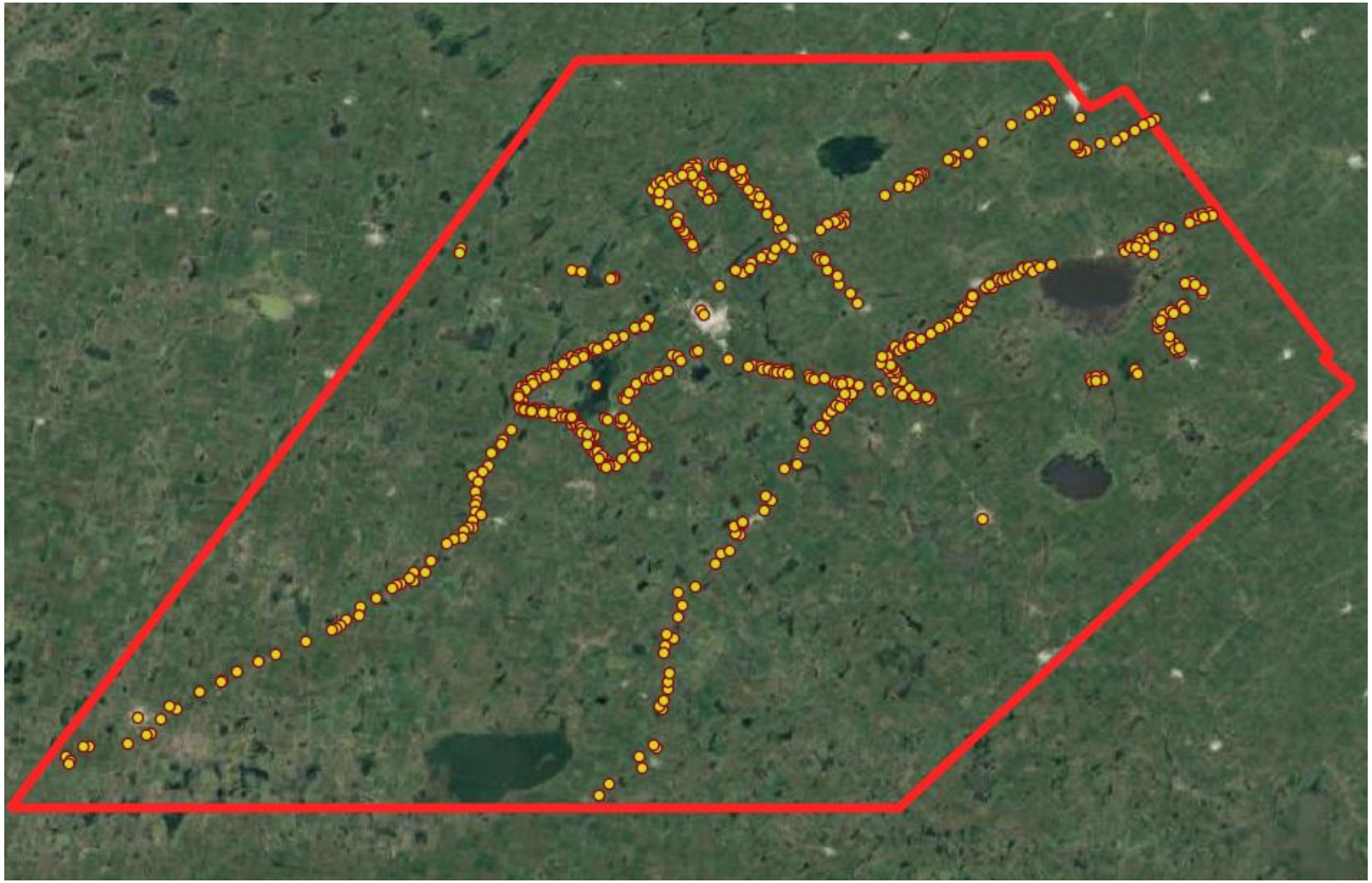
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Google Earth Engine

aws

SENTINEL Hub by SINERGISE

DESAFIO AGTECH. ZONA DE INTERÉS.



DESAFIO AGTECH. QUE SE BUSCA?

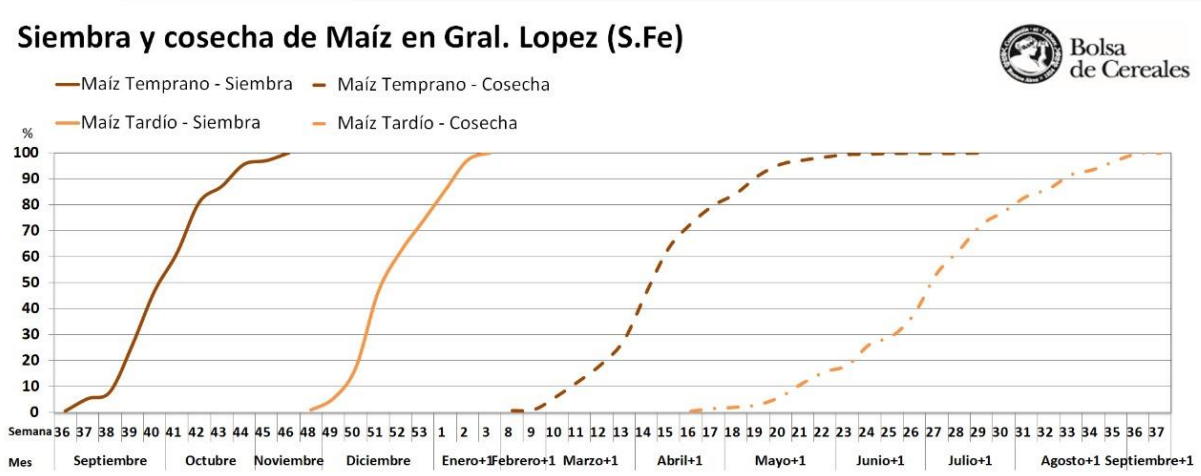
DESAFÍOS AGTECH

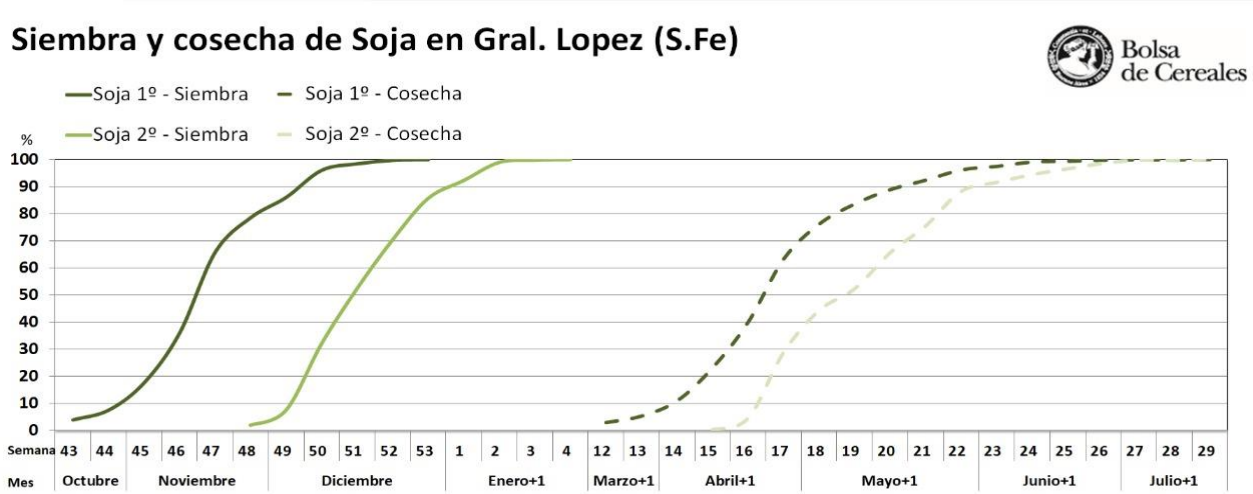
Hacer una clasificación en Gral. López:
- Identificar lo cultivado para ciertas ubicaciones.
- Hacer un código que trabaje de forma automática.

Cultivos: Maíz (1ra y 2da), Soja (1ra y 2da) y Barbechos.
Campañas: 18/19 y 19/20

Datasets: Train, Test, Etiquetas

Método de chequeo: $\texttt{balanced-accuracy}(y, \hat{y}, w) = \frac{1}{\sum \hat{w}_i} \sum_i 1(\hat{y}_i = y_i)\hat{w}_i \qquad \hat{w}_i = \frac{w_i}{\sum_j 1(y_j = y_i) w_j}$





DESAFÍOS AGTECH

EJEMPLOS CON

DISCLAIRMER: ESTE CODIGO ES A MODO DE EJEMPLO DIDÁCTICO, NO CONTIENE CONTROL DE ERRORES, NI SOFISTICACIONES, NI MEJORAS DE PERFORMANCE. TODOS LOS USOS DE LIBRERIAS EXTERNAS PUEDEN SER MEJORADAS EN SU IMPLEMENTACIÓN.

QUE HAREMOS

- Usaremos la librería gdal.
- Vamos a abrir una imagen Sentinel 2 y armaremos una hipermatriz.
- Vamos a visualizar RGB y un «falso color».
- Calcularemos y visualizaremos el ndvi.
- Calcularemos y visualizaremos donde hay lugares con poco o nada de vegetación.

QUE USAREMOS

- Imagen Sentinel descargada con las bandas B-G-R-NIR-SWIR1-SWIR2

QUE HAREMOS

- Usaremos la librería gdal y sklearn.
- Vamos a abrir una imagen Sentinel 2.
- Aplicaremos una clasificación no supervisada (Kmeans).
- Visualizaremos los resultados.

QUE USAREMOS

- Imagen Sentinel 2 descargada con las bandas B-G-R-NIR-SWIR1-SWIR2

QUE HAREMOS

- Usaremos la librería ee (Google Earth Engine).
- Abriremos unos puntos de entrenamiento y obtendremos sus datos asociados.
- Visualizaremos los valores asociados a cada uno de los puntos.

QUE USAREMOS

- Archivo con puntos de entrenamiento (previamente filtrados para apoyar en la img)

QUE HAREMOS

- Usaremos la librería gdal.
- Vamos a abrir una imagen Sentinel 2.
- Abriremos unos puntos de entrenamiento y obtendré sus datos asociados.
- Clasificaremos con Random Forest los puntos de testeo.
- Clasificaremos con Random Forest la imagen completa.

QUE USAREMOS

- Imagen Sentinel 2 descargada con las bandas B-G-R-NIR-SWIR1-SWIR2.
- Archivo con puntos de entrenamiento (previamente filtrados para apoyar en la img)
- Archivo con puntos de testeo (previamente filtrados para apoyar en la img)

QUE HAREMOS

- Usaremos la librería gdal.
- Vamos a abrir una imagen Sentinel 2.
- Abriremos unos puntos de entrenamiento y obtendremos sus datos asociados.
- Clasificaremos con SVM los puntos de testeo.

QUE USAREMOS

- Archivo con puntos de entrenamiento (previamente filtrados para apoyar en la img)
- Archivo con puntos de testeo (previamente filtrados para apoyar en la img)

QUE HAREMOS

- Usaremos la librería gdal.
- Vamos a abrir una imagen Sentinel 2.
- Abriremos unos puntos de entrenamiento y los dividiremos en dos sets.
- Clasificaremos con Random Forest, SVM y Aleatoriamente los puntos.
- Correremos algunas metricas.

QUE USAREMOS

- Archivo con puntos de entrenamiento (previamente filtrados para apoyar en la img)

GRACIAS!

https://github.com/camposalfredo/

Consultas? slack