

WEBINAR

Recomendaciones para
abordar el Desafío AgTech
utilizando **PYTHON**

Martes 10 /11 - 9hs

DESAFÍOS
AGTECH

www.desafiosagtech.com



Alfredo Campos

Desarrollador-Investigador-Docente



Matba Rofex



Bolsa
de Cereales



BOLSA
DE COMERCIO
DE ROSARIO

Qué es una clasificación y para qué clasificar.

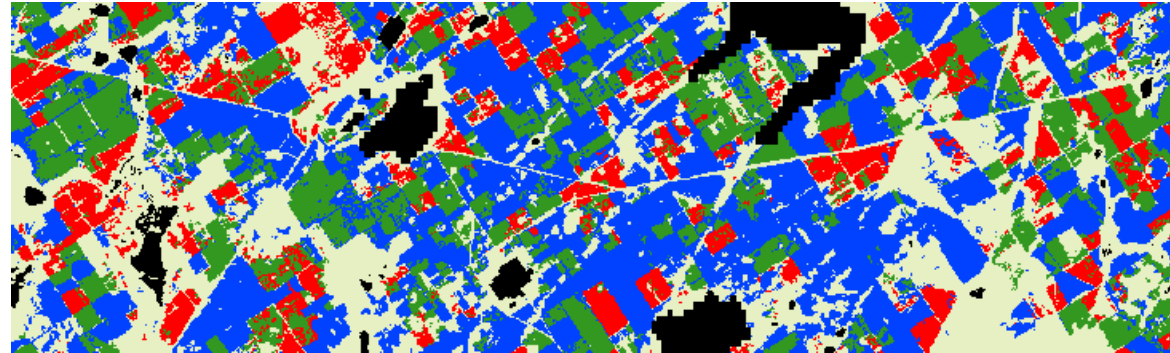
Imágenes satelitales. Fortalezas y Debilidades.

Cómo clasificar con imágenes satelitales.

Desafío AgTech. Dataset. Restricciones.

Formas de encarar el desafío con Python.

QUÉ SON?



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

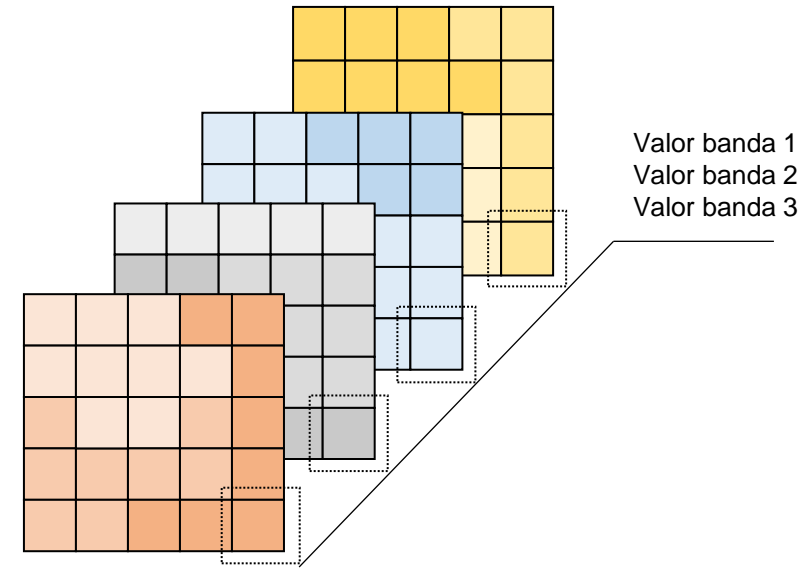
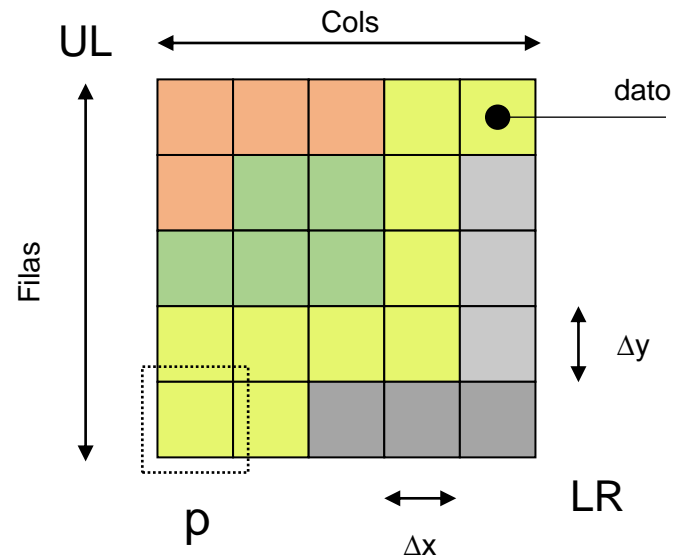
PARA QUÉ
SE USAN?

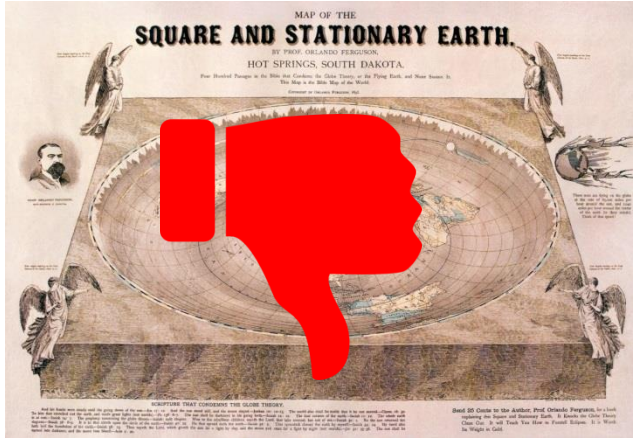
- Optimización de políticas públicas y privadas.
- Cuidado del ambiente.
- 1er paso para estimar la cosecha.



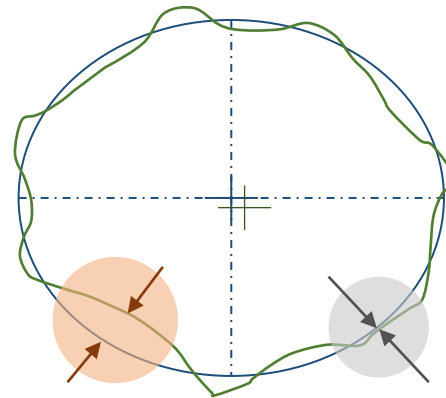
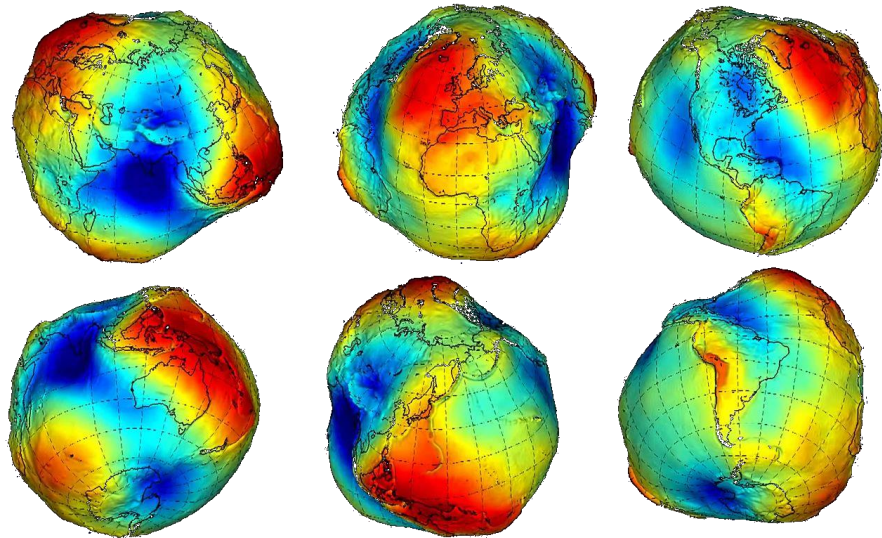
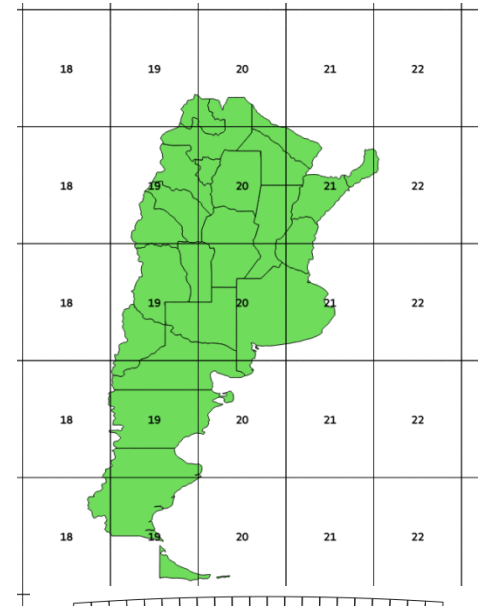
Bondades de las imágenes satelitales

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

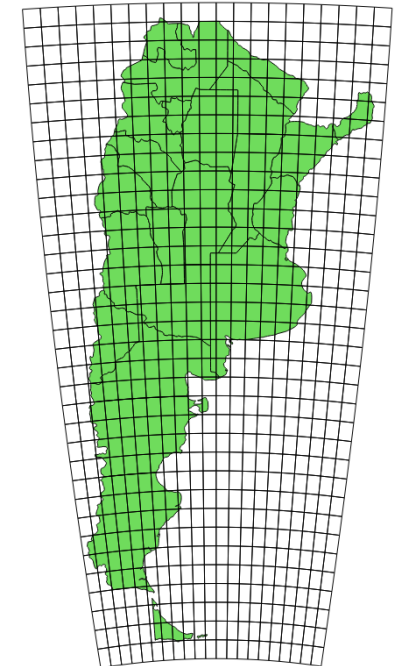


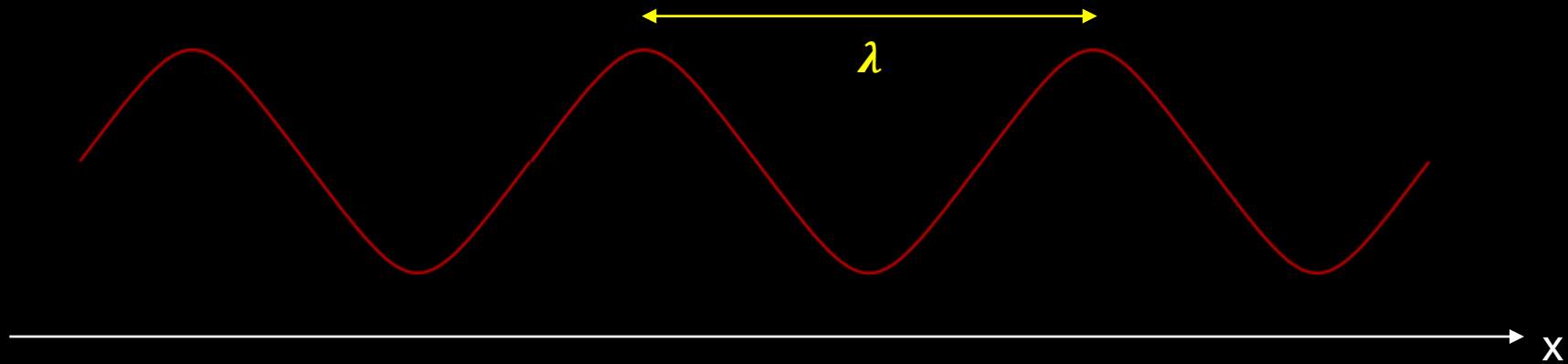


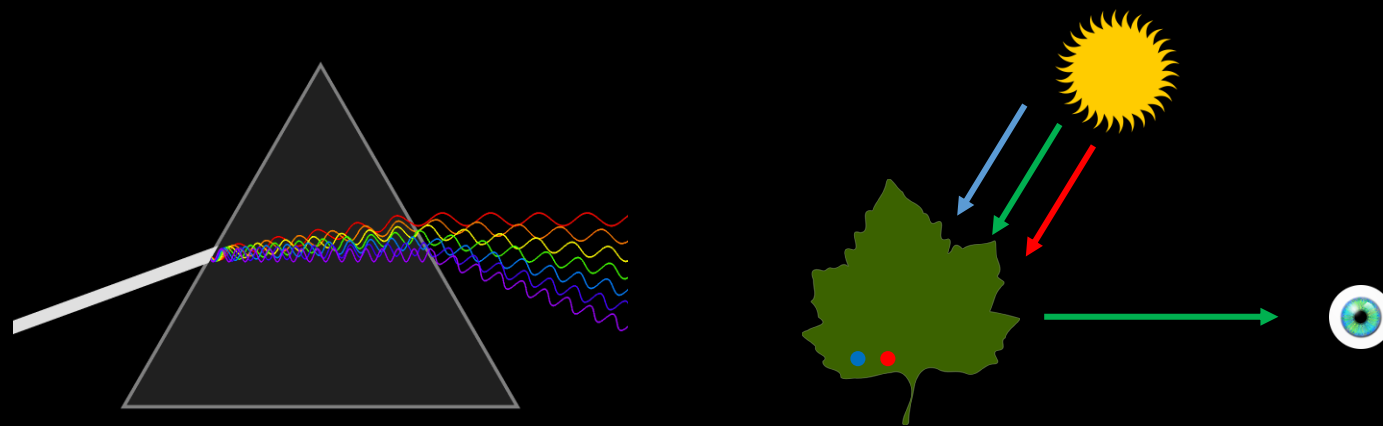
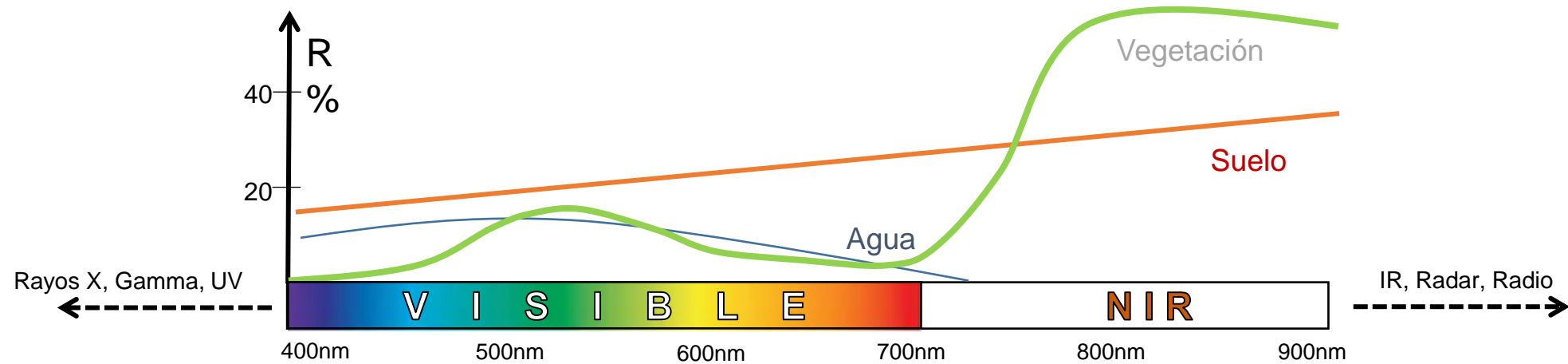
Lat Lon – WGS 84
(EPSG:4326)

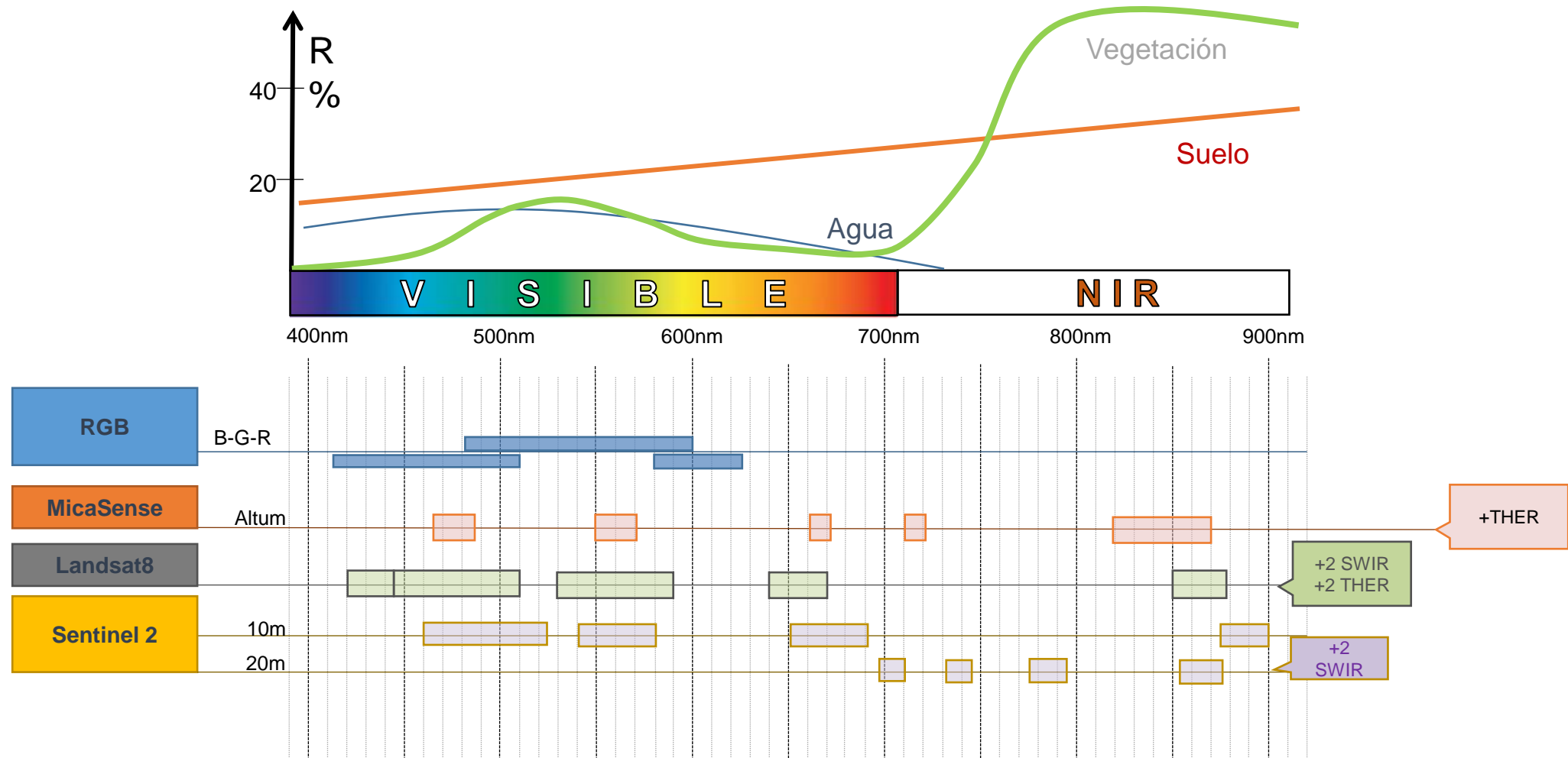


UTM Zona 20S – WGS 84
(EPSG:32720)









B2(Azul)



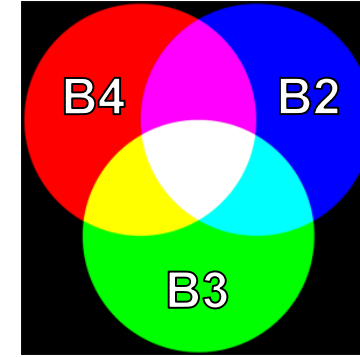
B3(Verde)



B4(Rojo)



B8(NIR)



B2(Azul)



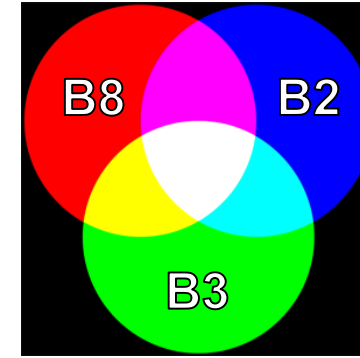
B3(Verde)



B4(Rojo)



B8(NIR)



B2(Azul)



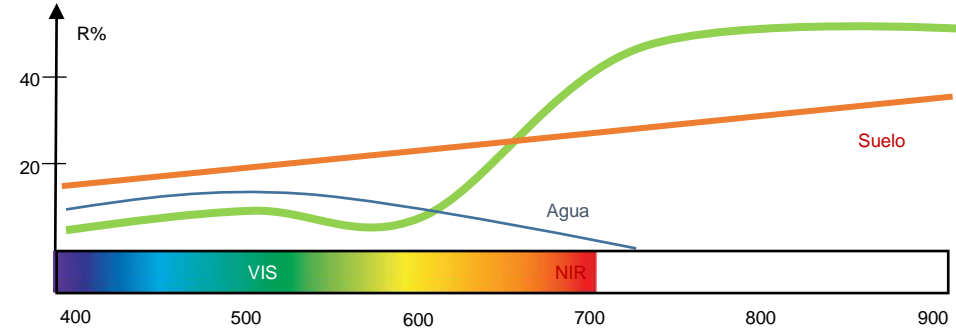
B3(Verde)



B4(Rojo)



B8(NIR)



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



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

$$EVI = \frac{2.5 (NIR - R)}{NIR + 6R - 7.5B + 1}$$

$$SAVI = \frac{NIR - R}{NIR + R + L} \cdot (1 + L)$$

$$TDVI = \sqrt{0.5 + \frac{NIR - R}{NIR + R}}$$

$$GNDVI = \frac{NIR - G}{NIR + G}$$

$$LAI = 3.618 \cdot EVI - 0.118$$

$$SR = \frac{NIR}{R}$$

$$OSAVI = \frac{1.5 (NIR - R)}{NIR + R + 0.16}$$

$$GSAVI = \frac{NIR - G}{NIR + G + L} \cdot (1 + L)$$

$$VARI = \frac{G - R}{G + R - B}$$

$$GRVI = \frac{NIR}{G}$$

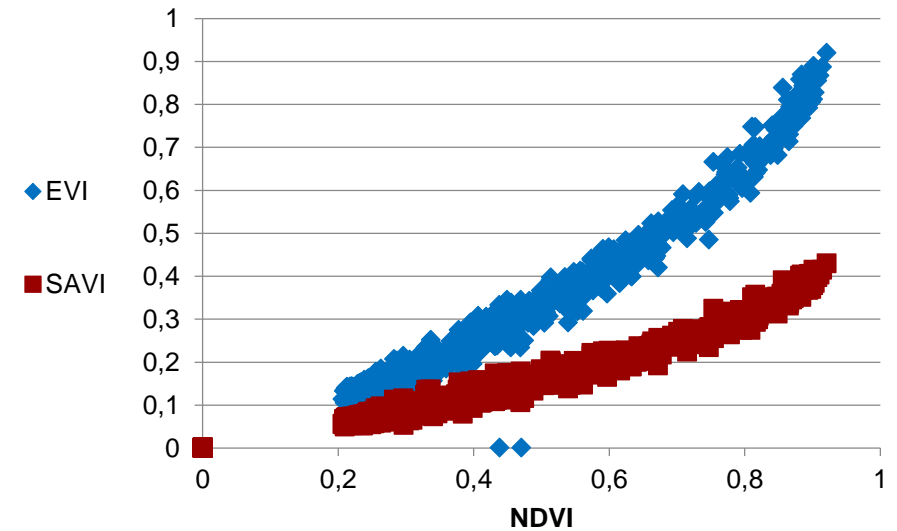
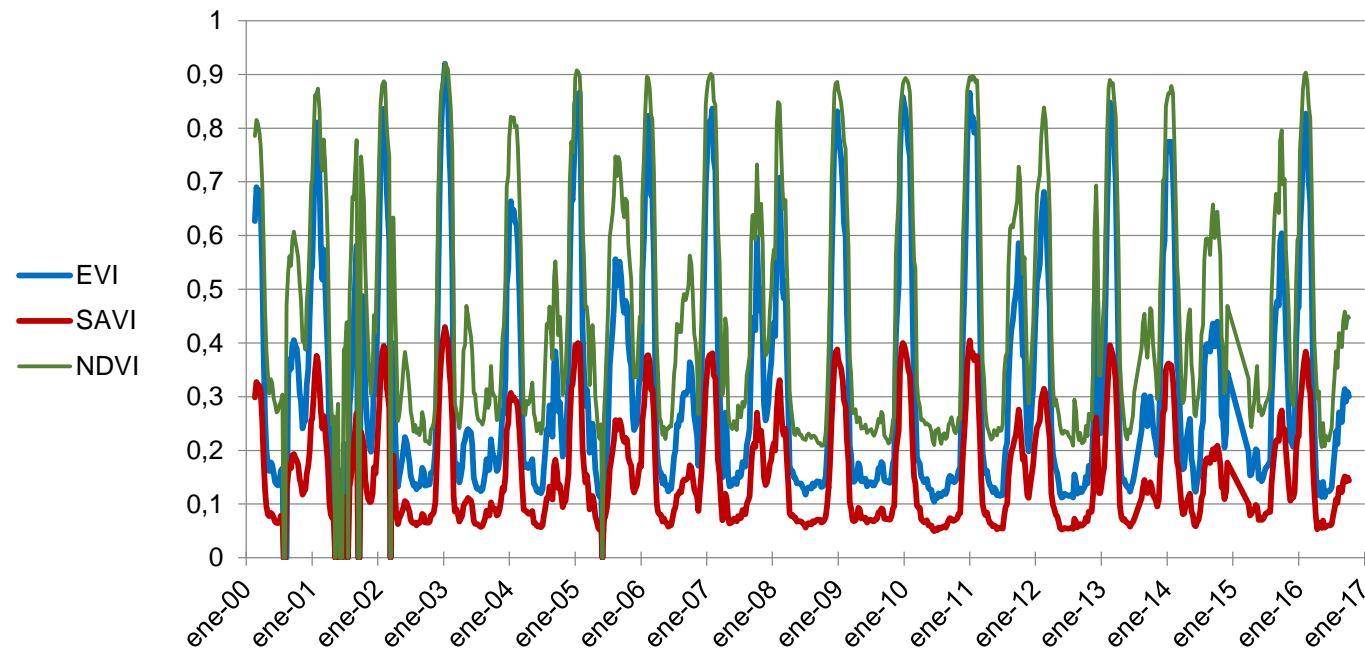
$$DVI = NIR - R$$

$$ENDVI = \frac{NIR + G - 2B}{NIR + G + 2B}$$

$$GDVI = NIR - G$$

$$IPVI = \frac{NIR}{NIR + R}$$

$$NGRDI = \frac{G - R}{G + R}$$



Landsat 8

- píxel: 30m
- img c/16 días
- 11 bandas
- \$=0*



Sentinel II

- Pixel 10m
- img c/10 días
- 13 bandas
- \$=0*



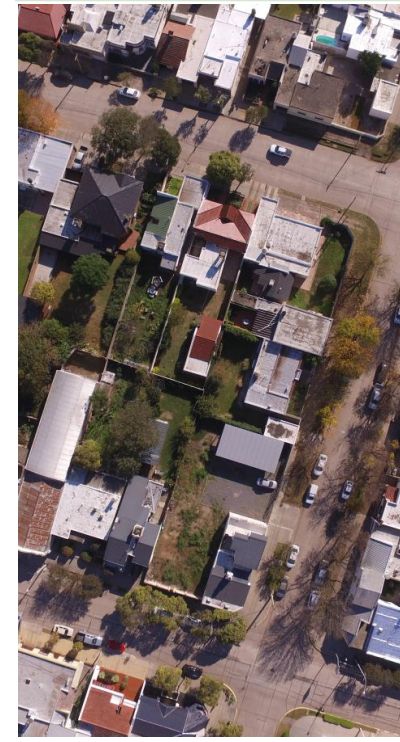
Planet

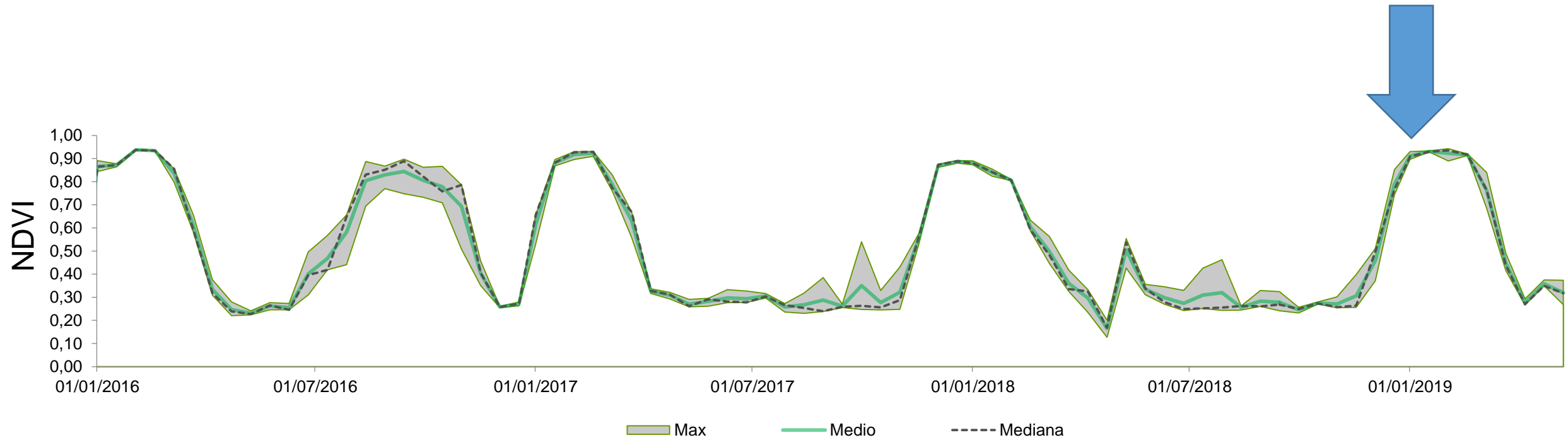
- Pixel 3m
- img c/1 día
- 4 bandas
- \$ > 0



Drone

- Pixel < 5cm
- img c/visita
- 3-5 bandas
- \$↑





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

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



Google Earth Engine



SENTINEL Hub
by SINERGISE



Hacer una clasificación en Gral. López:

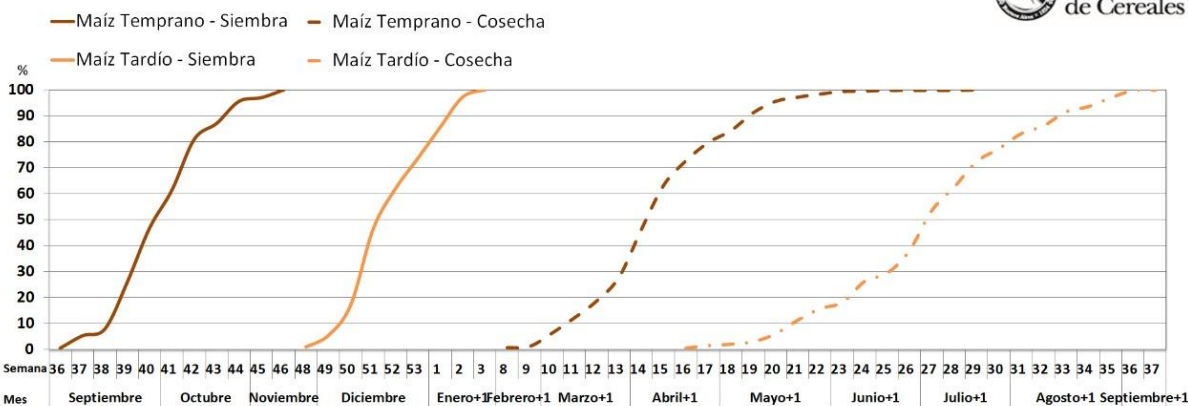
- Identificar lo cultivado para ciertas ubicaciones.
- Hacer un código que trabaje de forma automática.

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

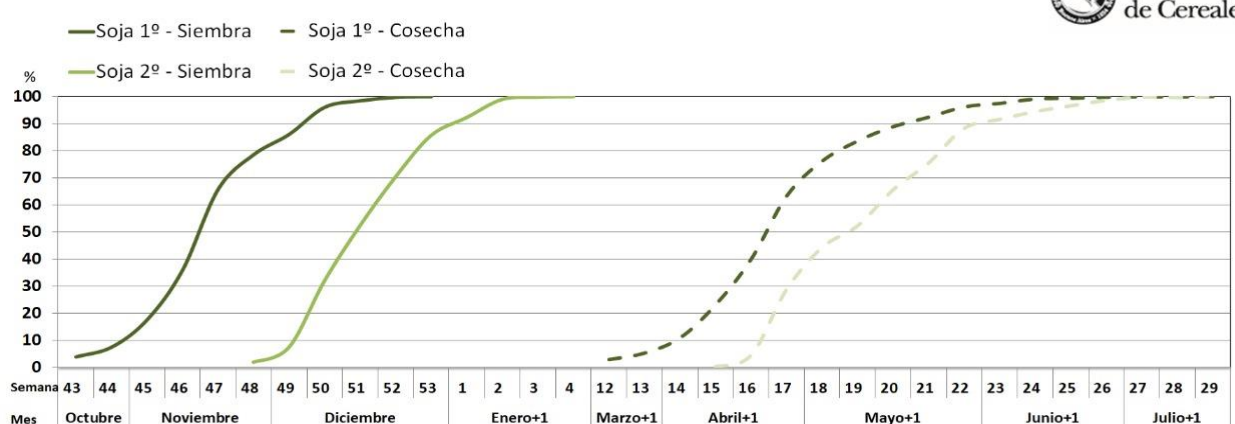
Datasets: Train, Test, Etiquetas

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

Siembra y cosecha de Maíz en Gral. Lopez (S.Fe)



Siembra y cosecha de Soja en Gral. Lopez (S.Fe)



EJEMPLOS CON



DISCLAIMER: 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 gdal.
- Vamos a abrir una imagen Sentinel 2.
- Abriremos unos puntos de entrenamiento y obtendré sus datos asociados.
- Visualizaremos los valores asociados a cada uno de los puntos.

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)

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 ee (Google Earth Engine).
- 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)

GRACIAS!

<https://github.com/camposalfredo/>

Consultas?  **slack**