

INFORME DE ESTANCIA

CONVOCATORIA BECAS CONACYT-GOBIERNO DEL ESTADO DE YUCATÁN

CRONOGRAMA DE ACTIVIDADES

SEMANA

(Descripción detallada de las actividades desarrolladas a lo largo de la estancia)

1

- Como primer paso, se llevó a cabo la instalación y configuración de la plataforma en línea Dataiku en el servidor designado para llevar a cabo el desarrollo del proyecto. Posteriormente, una base de datos de 233 imágenes multiespectrales (Tamaño: 960 x 1280 x 4) de parcelas agrícolas fue transferida a dicho servidor. Un ejemplo representativo de las imágenes transferidas se muestra en la **Figura 1**.



Figura 1. Ejemplo representativo en formato JPG de una imagen multiespectral de cultivos de mamey. La imagen es de tamaño 960 x 1280 y contiene 4 bandas espectrales: Green, Red, Red-Edge (REG), y Near-Infrared (NIR).

- Una vez instalada y configurada la plataforma Dataiku, se llevó a cabo la revisión de tutoriales para llevar a cabo un flujo básico de trabajo en dicha plataforma. La pantalla de inicio con el nuevo proyecto **ParrotSequoia** configurado se muestra en la **Figura 2**.

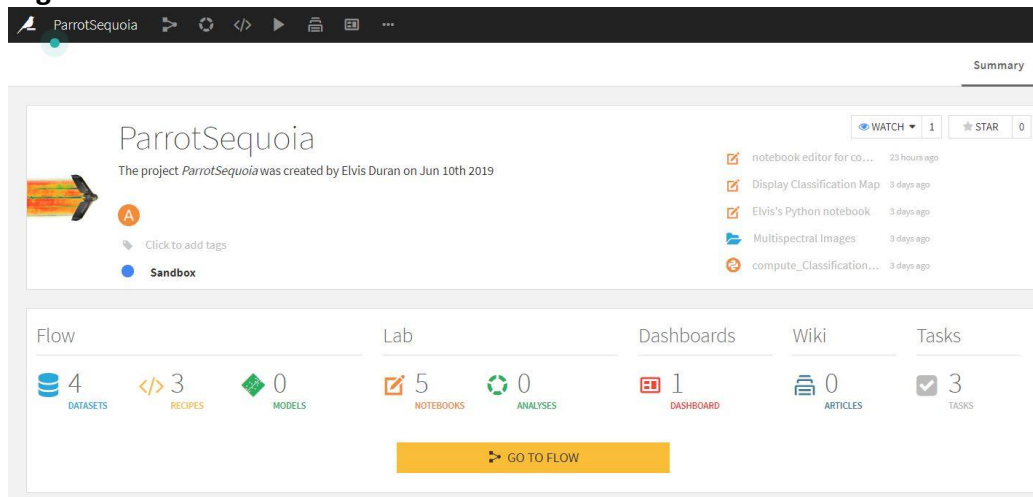


Figura 2. Pantalla de inicio del servidor Dataiku con el proyecto ParrotSequoia.

- Se llevó a cabo el desarrollo del tutorial: *101: Basic (Tutorial) import, explore, and prepare your first dataset*, con el fin de entender el flujo básico de trabajo dentro de Dataiku para posteriormente aplicarlo a la base de datos de imágenes.

INFORME DE ESTANCIA

CONVOCATORIA BECAS CONACYT-GOBIERNO DEL ESTADO DE YUCATÁN

2

- Se desarrolló código utilizando el lenguaje *Python* para llevar a cabo la extracción de información espectral utilizando las cuatro bandas (*Green*, *Red*, *Red-Edge*, y *NIR*) contenidas en las imágenes multispectrales. Dicho código fue importado como herramienta de procesamiento dentro del flujo de trabajo en Dataiku como se muestra en la **Figura 3**. El diagrama muestra en forma circular el módulo desarrollado en Python, el cual toma como entrada la carpeta de imágenes multispectrales *Multispectral Images* y produce como resultado el archivo *Spectral_Features* en formato CSV con las variables espectrales calculadas por píxel. La información espectral extraída incluye las intensidades de cada canal y los valores de los índices de vegetación: *NDVI*, *NDVI_REG*, *MSR*, *MSR_REG*, *CI_GRE* y *CI_REG*. De modo que un total de **10** variables espectrales fueron calculadas por píxel (ver **Figura 4**).

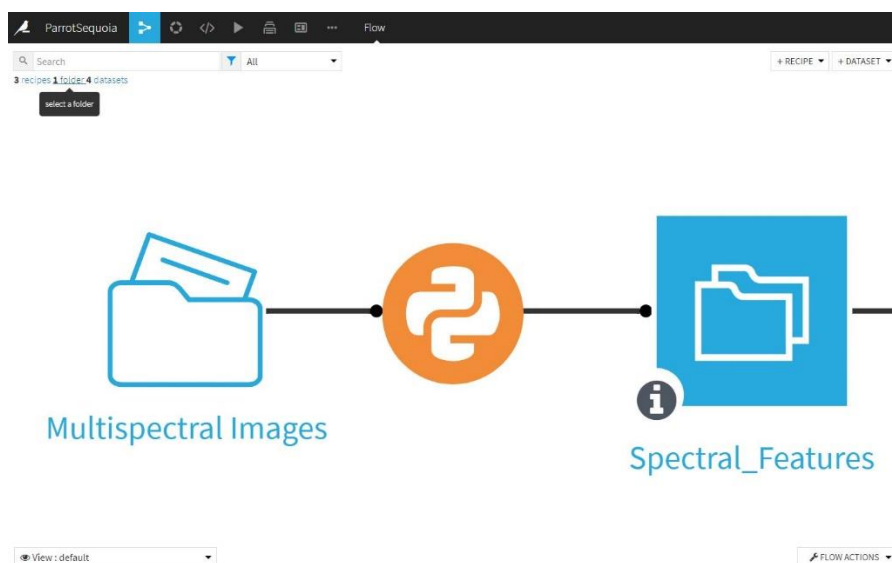


Figura 3. Diagrama de flujo en Dataiku que muestra los siguientes elementos de izquierda a derecha: a) Carpeta de imágenes multispectrales, b) Código en Python para extracción de información espectral y c) Archivo con las variables espectrales por píxel en formato CSV

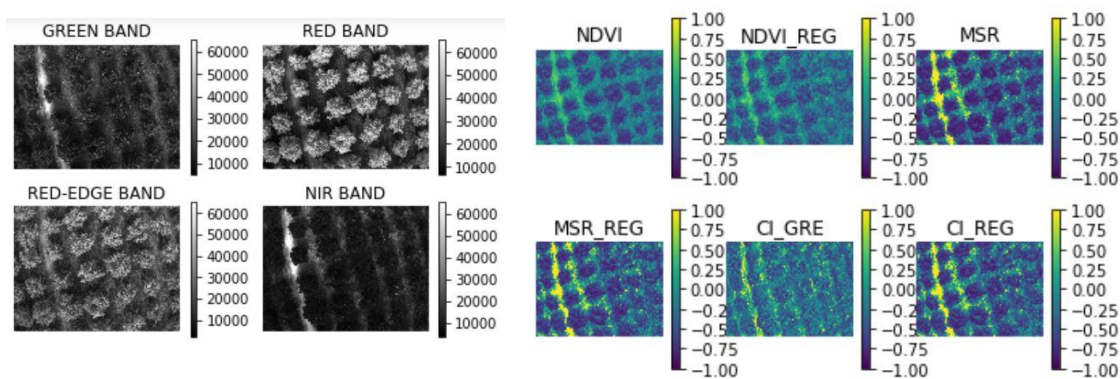


Figura 4. Intensidad de las cuatro bandas espectrales de la imagen (izquierda) y valores de seis mapas de índices de vegetación (derecha)

INFORME DE ESTANCIA

CONVOCATORIA BECAS CONACYT-GOBIERNO DEL ESTADO DE YUCATÁN

- Una vez generado el archivo con las variables espectrales, se procedió a normalizar dichas variables para colocarlas en una misma escala y posteriormente entrenar un algoritmo de Machine Learning. Para ello, se desarrolló un módulo de Python que fue introducido dentro del flujo de trabajo como se muestra en la **Figura 5**. Este módulo calcula el *puntaje Z* o *puntaje estandarizado* ($Z = \frac{X - \mu}{\sigma}$) de cada píxel en la imagen para cada una de las 10 variables, de manera que cada variable tenga una distribución con media cero y desviación estándar unitaria. Este archivo se genera en formato CSV (*Spectral_Features_Normalized*). Adicionalmente, el módulo descarta pixeles con valores tipo NaN y genera una máscara binaria (*Pixel_Mask*), en donde "0" representa el píxel descartado y "1" el píxel utilizado.

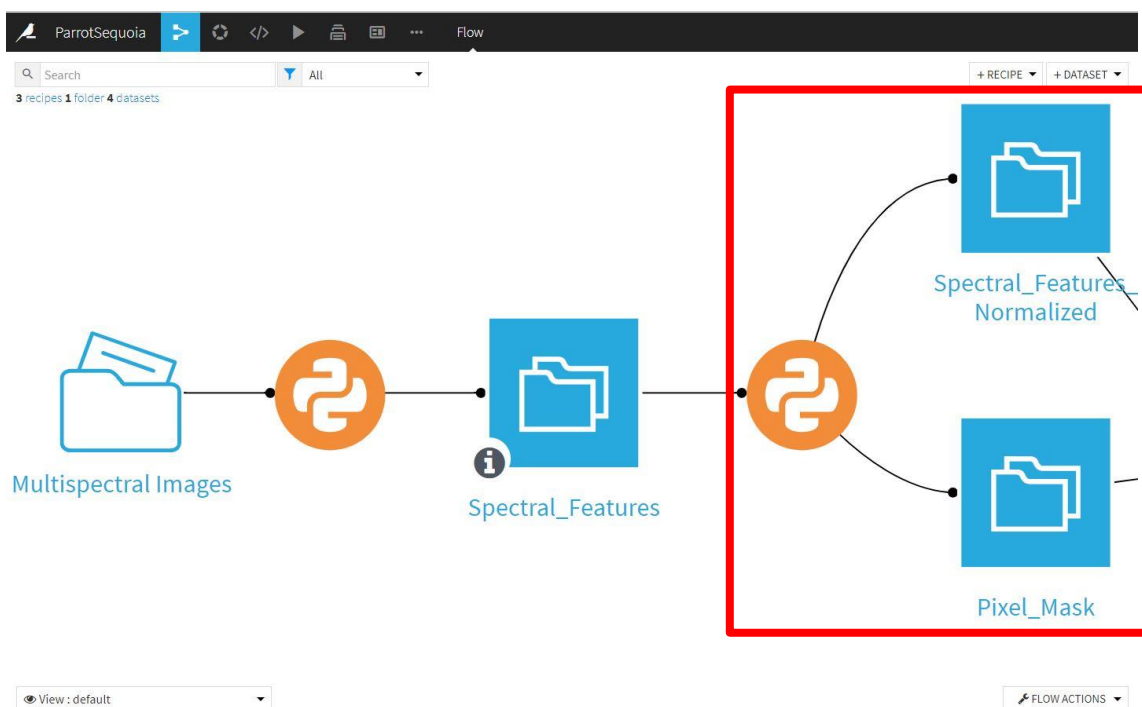


Figura 5. Diagrama de flujo en Dataiku donde se muestra con un recuadro rojo el módulo de Python para normalización de las variables espectrales mediante el cómputo del puntaje Z.

INFORME DE ESTANCIA

CONVOCATORIA BECAS CONACYT-GOBIERNO DEL ESTADO DE YUCATÁN

- Finalmente, se desarrolló un módulo en Python para llevar a cabo la clasificación de imágenes multiespectrales utilizando el algoritmo de agrupamiento **KMeans** (ver **Figura 6**). Este algoritmo establece automáticamente las clases u objetos que se encuentran en la imagen, agrupándolos en base a la similitud de sus características espectrales. Este módulo genera como resultado un mapa de clasificación (*Classification_Map*), en el que cada color representa un conjunto de elementos que comparten las mismas características espectrales. Para nuestra aplicación en el área agrícola, el mapa de clasificación distingue principalmente entre diversos tipos de suelo y vegetación presentes en las imágenes. Ejemplos representativos de los resultados de clasificación se muestran en la **Figura 7**. En estos ejemplos se compara la imagen original tomada con el Drone y el mapa de clasificación generado con KMeans. Es posible apreciar una gran correlación entre los cultivos y el suelo de la imagen original con las clases representadas con tres colores diferentes en el mapa de clasificación. Esto demuestra el potencial del algoritmo KMeans para agrupación automática de cultivos y suelos en imágenes multiespectrales, con el cual es posible desarrollar aplicaciones de mayor complejidad para la detección de zonas agrícolas en deterioro que requieran atención inmediata.

4

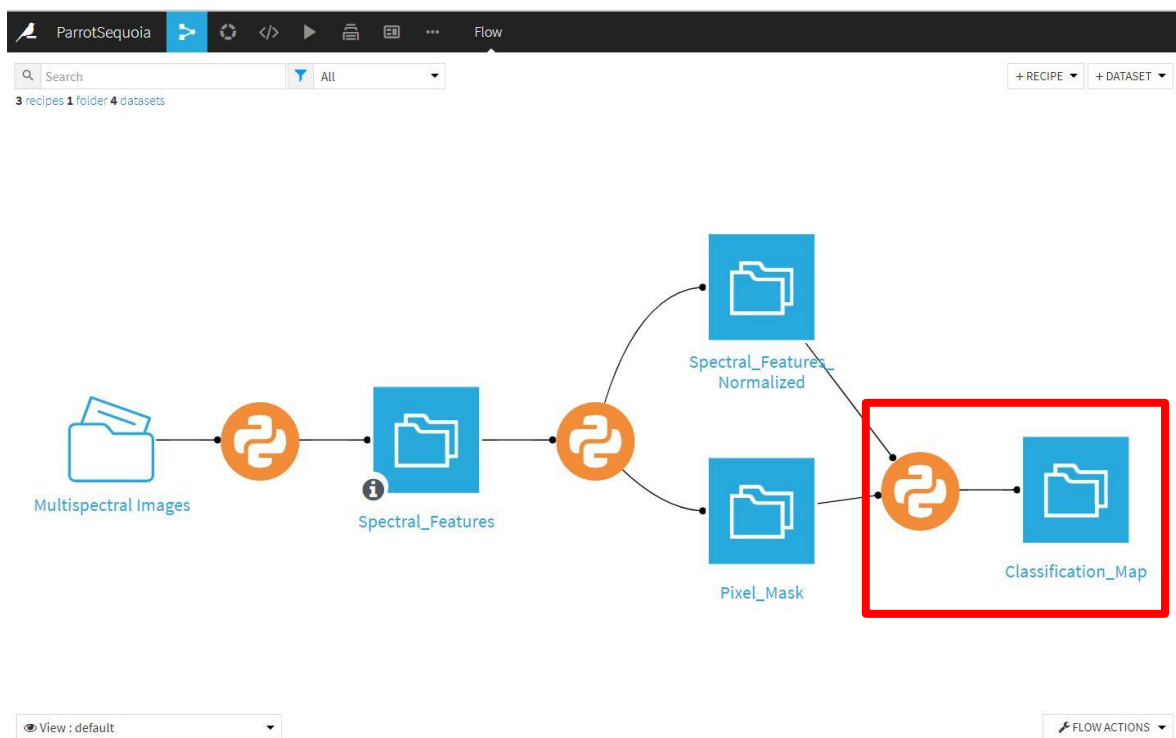


Figura 6. Diagrama de flujo final en Dataiku donde se muestra con un recuadro rojo el módulo de Python para clasificación automática de imágenes multiespectrales usando el algoritmo KMeans.

INFORME DE ESTANCIA

CONVOCATORIA BECAS CONACYT-GOBIERNO DEL ESTADO DE YUCATÁN

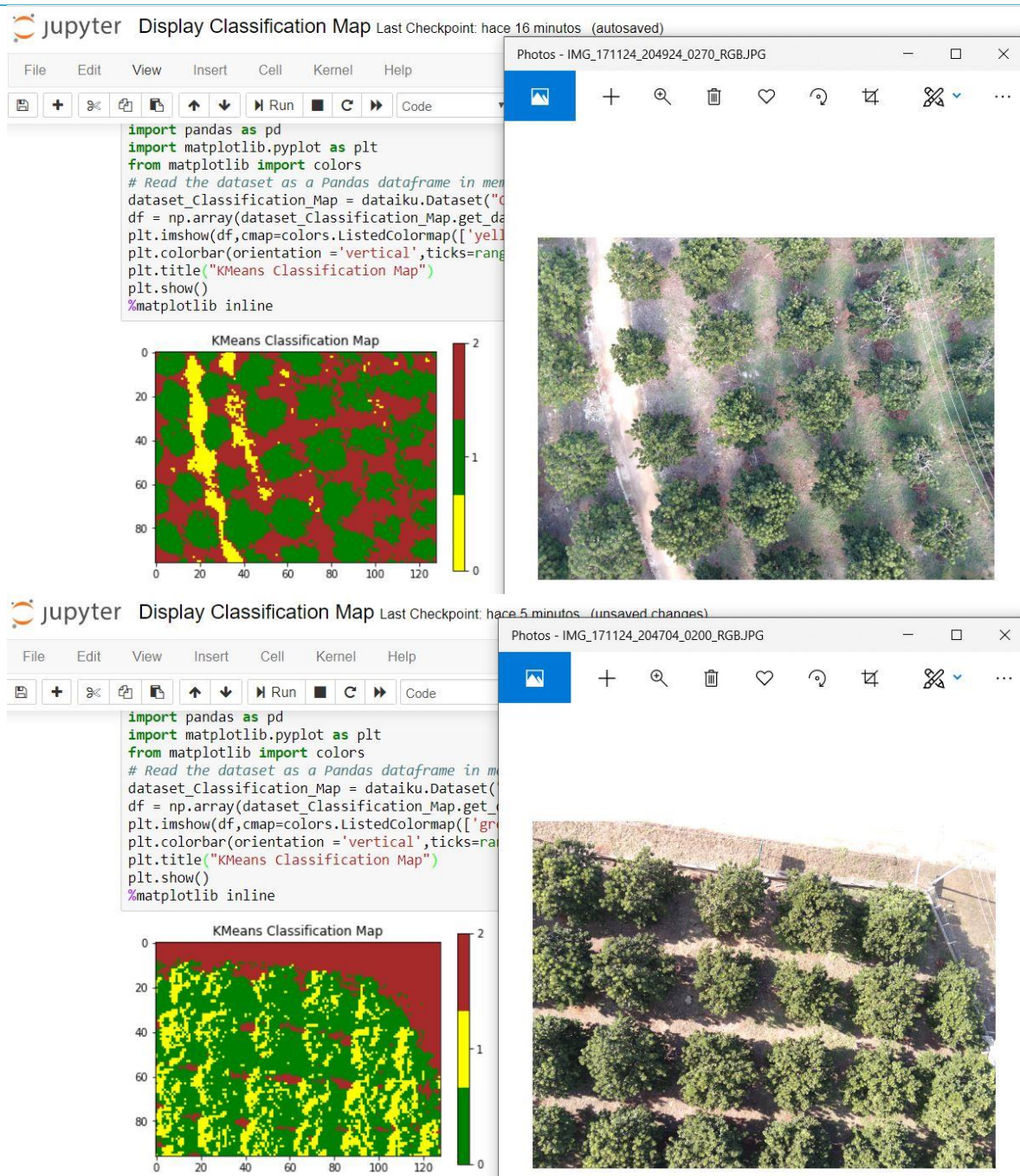


Figura 7. Ejemplos representativos de imágenes aéreas multispectrales de zonas agrícolas (derecha) con sus respectivos mapas de clasificación generados con el algoritmo KMeans (izquierda)