MANUAL TÉCNICO Y DE USUARIO

Estimación del Rendimiento de Papa (t/ha) con UAV Multtespectral y Aprendizaje Automatico

Versión 1.0

2025

Autor: John Guerra Flores

1. Introducción

Este manual describe el uso y la operación de la aplicación desarrollada para estimar automáticamente el rendimiento de papa (t/ha) a nivel de parcela (plot), a partir de ortomosaicos multiespectrales UAV de 5 bandas (B, G, R, RE, NIR) adquiridos en tres fechas (20251002, 20251103, 20251210).
El sistema implementa un flujo completo de: preprocesamiento geoespacial, segmentación automática (Otsu), desmezcla lineal (LUM), extracción de variables espectrales por parcela, unión multitemporal, integración con rendimiento real y entrenamiento/evaluación de modelos (baseline PLSR y modelo propuesto RandomForest con evaluación robusta).

2. Alcance de la aplicación

La aplicación permite:

1. Construir **stacks multibanda** por fecha (GeoTIFF de 5 bandas).
2. Recortar automáticamente cada parcela definida en shapefile (plot).
3. Separar vegetación–fondo mediante **umbral de Otsu**.
4. Generar productos **LUM** (abundancias veg/suelo y reflectancia reconstruida).
5. Calcular variables por parcela: **VF, NDVI_mean, SR_mean, CHLGR_mean, MARI_mean, OSAVI_mean, SAVI2_mean**.
6. Unir variables multitemporales (3 fechas) y crear el dataset final **X/y**.
7. Preparar rendimiento (yield) y unirlo por parcela.
8. Entrenar y evaluar modelos (PLSR y RandomForest).
9. Exportar resultados: dataset final, predicciones, métricas, importancia de variables y modelo entrenado.

Limitaciones:

- El pipeline está diseñado para procesamiento por parcelas; procesar ortomosaicos completos en memoria puede generar errores de RAM.
- Requiere geometrías válidas y correcta correspondencia `plot_id` ↔ `obsUnitId` ↔ rendimiento de campo.

3. Descripción general del flujo (pipeline)

La aplicación está implementada en **13 notebooks** (1…13), que se ejecutan en orden:

1. **Apilar imágenes multiespectrales** (por fecha)
2. **Cargar archivo de forma** (parcelas)
3. **Recorte por parcela** (plot)
4. **Umbral de Otsu en datos crudos**
5. **Selección de endmembers** (veg/suelo)
6. **Desmezcla lineal (LUM)**
7. **Otsu sobre mapas LUM**
8. **Extracción de características** (por parcela y fecha)

9. **Unir y organizar las 3 fechas (temporal)**
10. **Preparar YIELD** (plot_id y t/ha)
11. **Unión final (X/y)**
12. **PLSR (baseline)**
13. **Modelo propuesto (RandomForest)**

PARTE A: MANUAL DE USUARIO

4. Requisitos del sistema (usuario)

Hardware recomendado

- RAM: **16 GB** mínimo (ideal 32 GB si los ortomosaicos son muy grandes).
- CPU: 4 núcleos o más.
- Almacenamiento: espacio suficiente (los recortes y salidas pueden ser grandes).

Software

- Windows 10/11 (o Linux).
- Anaconda / Miniconda.
- Jupyter Lab.

---

5. Instalación y configuración (pasos del usuario)

5.1 Crear entorno conda

1. Abrir **Anaconda Prompt**.
2. Crear el entorno (ejemplo):

```
conda create -n uav_papa python=3.10 -y
conda activate uav_papa
```

5.2 Instalar librerías necesarias

conda install -c conda-forge rasterio geopandas shapely fiona pyproj scikit-image matplotlib pandas numpy scikit-learn openpyxl joblib -y

5.3 Ejecutar Jupyter Lab

jupyter lab

6. Estructura de carpetas (recomendada)

Se recomienda mantener una estructura consistente como:

- `drone_data/FECHA/BANDAS/` → GeoTIFF por banda (B,G,R,RE,NIR)
- `shapes/` → shapefile/geojson de parcelas
- `measurements/` → archivos de rendimiento (yield) y otros datos
- `resultado/` → salidas del pipeline

Ejemplo:

- `.../drone_data/F03112025/BANDAS/20251103_B.tif`
- `.../resultado/PASO_11_dataset_final/dataset_Xy_3dates_yield2025.csv`

---

7. Preparación de datos de entrada (usuario)

7.1 Orto mosaicos por fecha

Por cada fecha deben existir 5 archivos:

- `*_B.tif`
- `*_G.tif`
- `*_R.tif`
- `*_RE.tif`
- `*_NIR.tif`

Condición importante: todas las bandas deben tener el mismo CRS, tamaño y alineación.

7.2 Parcelas (shapefile)

El shapefile debe contener:

- `plot_id` (1…202 en tu caso)
- `obsUnitId` (código experimental, por ejemplo `2025_F1P1`)

7.3 Rendimiento (yield)

Debe existir un archivo de campo que permita calcular `yield_t_ha`. En tu caso se usó:

- `obsUnitId`
- variable de rendimiento (por ejemplo `tubwght_total_kgm-2`)

---

8. Ejecución paso a paso (orden recomendado)

Regla práctica: ejecutar en orden 1 → 13.

Paso 1 (Notebook 1): Apilamiento por fecha

- Entrada: 5 GeoTIFF (bandas)
- Salida: `*_stack.tif` (5 bandas)

Paso 2: Cargar shapefile

- Entrada: shapefile con parcelas
- Salida: parcelas validadas (CRS correcto)

Paso 3: Recorte por parcela

- Entrada: stack por fecha + parcelas
- Salida: `plot_*_stack.tif` por parcela

Paso 4: Otsu en crudo

- Entrada: plot stacks
- Salida: máscara veg/fondo + conteos válidos

Paso 5: Endmembers

- Entrada: parcela + máscara
- Salida: `veg_end` y `soil_end` por parcela

Paso 6: LUM

- Entrada: stacks por parcela + endmembers
- Salida: `plot_*_LUM_abund.tif`, `plot_*_LUM_RL.tif` + PNG abundancias

Paso 7: Otsu en LUM

- Entrada: mapas LUM
- Salida: máscara refinada

Paso 8: Extracción de features

- Salida por fecha: CSV con variables:
    - VF, NDVI_mean, SR_mean, CHLGR_mean, MARI_mean, OSAVI_mean, SAVI2_mean
    - valid_pixels, total_pixels

Paso 9: Unión temporal 3 fechas

- Entrada: 3 CSV de features
- Salida: tabla temporal unida

Paso 10: Preparación yield

- Entrada: yield raw + obsUnitId
- Salida: `yield_by_plot_2025.csv` con `plot_id`, `yield_kgm2`, `yield_t_ha`

Paso 11: Dataset final X/y

- Entrada: tabla temporal + yield por plot
- Salida:
  - `dataset_Xy_3dates_yield2025.csv`
  - `dataset_Xy_3dates_yield2025.xlsx`

Paso 12: PLSR (baseline)

- Salida: predicciones y coeficientes del baseline

Paso 13: Modelo propuesto (RandomForest)

- Salida:
  - predicciones CV capP99
  - importancia de variables
  - resumen final
  - (opcional) modelo `.joblib`

---

9. Resultados generados (archivos principales)

En tu proyecto se generaron, entre otros:

- `dataset_Xy_3dates_yield2025.csv` (dataset final)
- `plsr_test_predictions_yield_t_ha.csv` (baseline)
- `RandomForest_CV_predictions_with_plotid_capP99.csv` (predicciones del modelo propuesto)
- `RandomForest_feature_importance_yield_t_ha_capP99.csv` (importancia de variables)
- `final_model_summary.csv` (resumen comparativo)

---

10. Uso del modelo para predecir un nuevo lote (usuario)

Para predecir en una nueva campaña o en nuevas parcelas:

1. Repetir pasos 1 a 9 para obtener el dataset multitemporal de features (sin yield).
2. Asegurar que las columnas tengan el mismo nombre/formato que el entrenamiento (mismas variables y sufijos de fecha).
3. Cargar el modelo entrenado (si existe `.joblib`) y ejecutar predicción:

```
import pandas as pd

import joblib


Xnew = pd.read_csv("TU_DATASET_MULTITEMPORAL_SIN_YIELD.csv")


# seleccionar columnas predictoras (las mismas 21 usadas en entrenamiento)

features = [c for c in Xnew.columns if any(k in c for k in ["VF_", "NDVI_mean_", "SR_mean_", "CHLGR_mean_", "MARI_mean_", "OSAVI_mean_", "SAVI2_mean_"])]

X = Xnew[features]


model = joblib.load("RandomForest_model_yield_t_ha_capP99.joblib")  # si lo guardaste

Xnew["yield_pred_t_ha"] = model.predict(X)


Xnew.to_csv("predicciones_yield.csv", index=False)
```

11. Solución de problemas frecuentes (usuario)

11.1 Error: MemoryError / bad allocation

Causa: procesamiento de arrays muy grandes o kernel con memoria fragmentada.
Solución práctica:

- Reiniciar el kernel de Jupyter (“Kernel → Restart Kernel”).
- Procesar por lotes (por ejemplo plots 1–50, 51–100, etc.).
- Cerrar otras aplicaciones pesadas.
- Evitar apilar todo el ortomosaico completo en un solo array cuando no sea necesario.

11.2 Error: “No such file or directory”

Causa: ruta incorrecta o nombre de archivo diferente.
Solución: verificar rutas y nombres exactos. Preferir `r"..."` para rutas en Windows.

11.3 Error TIFF (TileWidth / StripOffsets)

Causa: parámetros inválidos de tiling o escritura incompleta.
Solución: usar bloques estándar (ej. 256×256) y evitar tiles enormes.

PARTE B: MANUAL TÉCNICO

12. Arquitectura técnica de la aplicación

La aplicación está compuesta por:

1. **Capa geoespacial (Raster/Vector)**
   - Raster: stacks multibanda y productos LUM (GeoTIFF).
   - Vector: parcelas (shapefile) con `plot_id` y `obsUnitId`.
2. **Capa de análisis espectral**
   - Cálculo de índices (NDVI, SR, OSAVI, SAVI2, CHLGR, MARI).
   - Métricas por parcela (medias sobre píxeles válidos).
3. **Capa de modelado**
   - Baseline: PLSR.
   - Propuesto: RandomForest + validación cruzada + capP99.
4. **Capa de salidas**
   - CSV/XLSX, GeoTIFF intermedios, predicciones, importancias, resumen final.

---

13. Especificación de entradas y salidas

13.1 Entradas

- Raster por fecha: 5 bandas (B, G, R, RE, NIR) GeoTIFF.
- Vector parcelas: shapefile con geometrías y campos `plot_id`, `obsUnitId`.
- Yield de campo: CSV con `obsUnitId` y variable de rendimiento base.

13.2 Salidas

- `*_stack.tif` (5 bandas por fecha).
- `plot_*_stack.tif` (recortes por parcela).
- máscaras (Otsu crudo y Otsu en LUM).
- `plot_*_LUM_abund.tif` (2 bandas) y `plot_*_LUM_RL.tif` (5 bandas).
- CSV de features por fecha.
- Dataset final X/y (CSV y XLSX).
- Predicciones y métricas de modelos.
- Importancia de variables (RandomForest).

---

14. Descripción técnica de módulos clave

14.1 Segmentación Otsu

- Se aplica umbral automático para separar vegetación/fondo.

- Genera máscara binaria.
- Permite computar VF y definir píxeles válidos para extracción de índices.

14.2 Desmezcla lineal (LUM)

- Modelo espectral con dos endmembers (vegetación y suelo).
- Estima abundancias por píxel:
  - A_veg, A_soil
- Reconstruye reflectancia:
  - RL = A × E
- Beneficio: reduce influencia de mezcla suelo–vegetación.

14.3 Extracción de features

- Se filtran píxeles válidos (máscara).
- Se calculan medias por parcela y fecha.
- Variables generadas:
  - VF, NDVI_mean, SR_mean, CHLGR_mean, MARI_mean, OSAVI_mean, SAVI2_mean.

14.4 Estrategia robusta capP99

- Se recorta `yield_t_ha` al percentil 99 para evaluación robusta.
- Reduce sensibilidad del RMSE a valores extremos.
- Mantiene todas las observaciones (no elimina filas).

---

15. Validación y métricas

Se emplean métricas estándar de regresión:

- **RMSE** (error cuadrático medio)
- **MAE** (error absoluto medio)
- **R^2** (coeficiente de determinación)

La evaluación se realiza principalmente mediante **validación cruzada k-fold**.

---

16. Seguridad, integridad y buenas prácticas

- Mantener respaldos del directorio `resultado/`.
- Congelar versiones de librerías (exportar `environment.yml`).
- No modificar manualmente `plot_id` sin documentar.
- Verificar unión `plot_id–obsUnitId` antes de entrenar.
- Registrar fecha/hora de ejecución y parámetros de vuelos si se reportan.

---

17. Mantenimiento y actualización

Para adaptar a nuevas campañas:

- Mantener mismas variables e índices.
- Ajustar fechas y rutas.
- Reentrenar modelo con yield de la nueva campaña (recomendado).
- Validar si los outliers se repiten y recalcular percentiles.

ANEXO A. Glosario rápido

- **UAV:** vehículo aéreo no tripulado.
- **Ortomosaico:** mosaico georreferenciado corregido geométricamente.
- **LUM:** Linear Unmixing Model (desmezcla lineal).
- **Endmember:** firma espectral “pura” (vegetación/suelo).
- **VF:** fracción de vegetación.
- **capP99:** recorte del target al percentil 99.