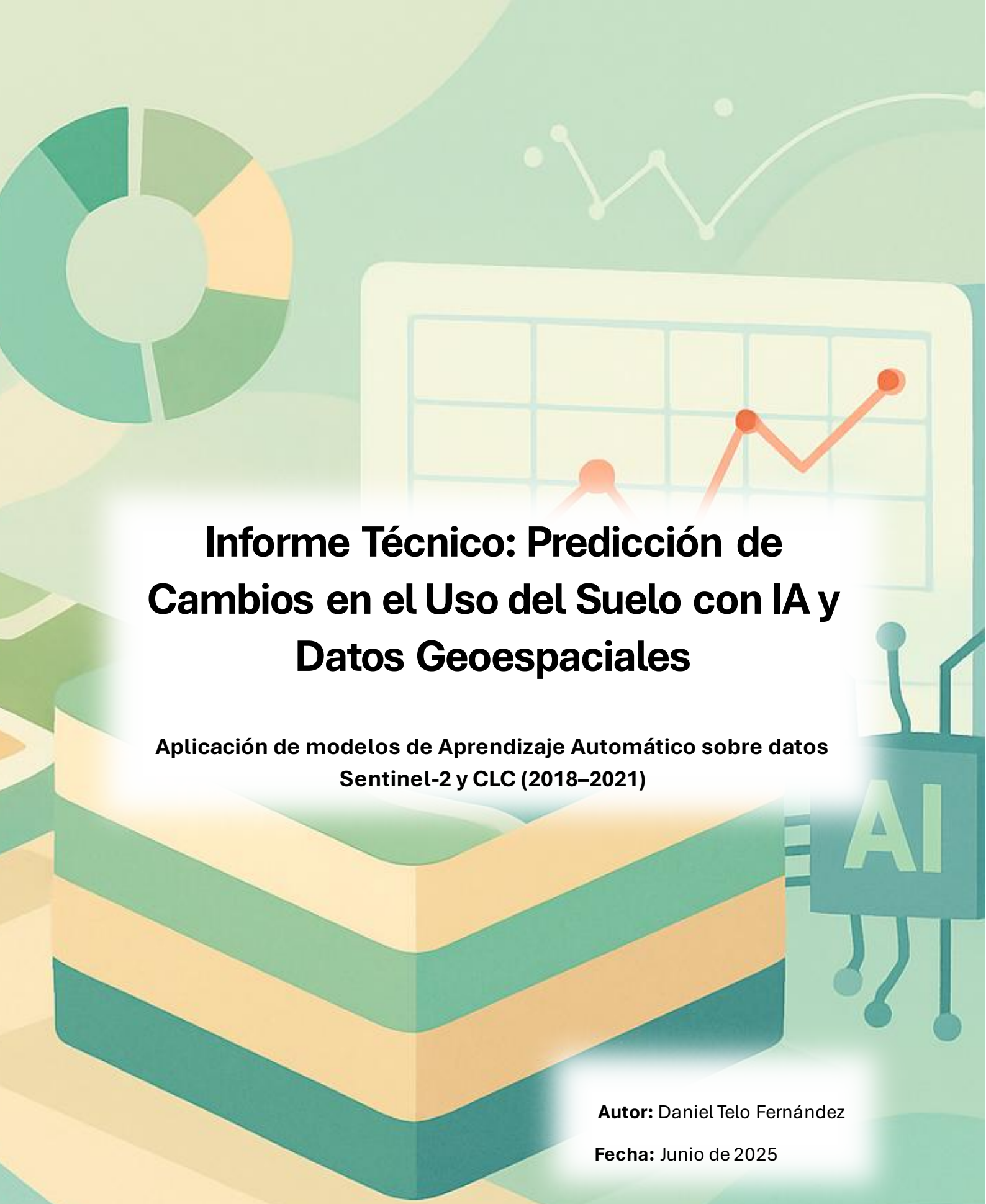


Informe Técnico: Predicción de Cambios en el Uso del Suelo con IA y Datos Geoespaciales

Aplicación de modelos de Aprendizaje Automático sobre datos Sentinel-2 y CLC (2018–2021)

Autor: Daniel Telo Fernández

Fecha: Junio de 2025

Contenido

1. Introducción

Este proyecto tiene como objetivo detectar cambios en el uso del suelo a lo largo del tiempo, en una zona específica, utilizando técnicas de inteligencia artificial. Para ello, se analiza la evolución del terreno en distintos años, identificando patrones de cambio mediante el procesamiento de datos satelitales.

La inteligencia artificial se encarga de reconocer estos patrones a partir de las **bandas espectrales** descargadas del satélite **Sentinel-2**, junto con algunos **índices** derivados de dichas bandas. A lo largo del documento, se presentará de manera detallada todo el flujo de trabajo implementado: desde la recopilación y preprocesamiento de los datos, hasta la generación de los insumos necesarios para el entrenamiento del modelo.

Además, se explicará el proceso de entrenamiento de la inteligencia artificial, así como la evaluación del modelo y los resultados obtenidos. Todo el desarrollo estará acompañado de imágenes ilustrativas y recursos visuales que facilitarán la comprensión de cada etapa.

2. Objetivos del proyecto

Familiarizarme con el uso de la inteligencia artificial como herramienta aplicada al análisis de datos espaciales estructurados, aprendiendo a interpretar conjuntos de datos numéricos que, de forma aislada, carecen de significado evidente. El objetivo práctico fue generar un modelo de IA entrenado con capas de cambio de uso del suelo (CLC), aplicarlo para evaluar su capacidad de generalización y medir su rendimiento mediante métricas cuantitativas. A través de este proceso, se buscó no solo validar la predicción de cambios territoriales, sino también comprender cómo optimizar el entrenamiento y mejorar la capacidad del modelo para extraer patrones útiles en escenarios geográficos reales.

3. Datos y fuentes utilizadas

3.1. Bandas Sentinel-2

Los datos independientes, refiriéndonos a datos que predicen, se obtienen a partir de las bandas del sentinel-2.
Fueron obtenidos a partir del "Copernicus Browser", concretamente se cogieron los datos del Sentinel-2 L2A, debido a que estas ya incluyen **corrección atmosférica**, lo cual garantiza que los valores de reflectancia correspondan a la superficie terrestre. Gracias a esto evitamos distorsiones causadas por condiciones atmosféricas variables.

Se seleccionaron las bandas **B02, B03, B04, B08, B11 y B12** de Sentinel-2 por su relevancia en la detección y caracterización de cambios en el uso del suelo, de la misma zona en años distintos, en nuestro caso 2018 y 2021.

Banda	Nombre	Resolución	Rango espectral (nm)
B02	Azul	10 m	490
B03	Verde	10 m	560
B04	Rojo	10 m	665
B08	Infrarrojo cercano (NIR)	10 m	842
B11	Infrarrojo de onda corta (SWIR 1)	20 m	1610
B12	Infrarrojo de onda corta (SWIR 2)	20 m	2190

Adicionalmente, la plataforma **Copernicus Browser** permite definir un área de interés y exportarla en formato **GeoJSON**, lo cual resulta especialmente útil para el preprocesamiento de las capas espaciales involucradas en el flujo de trabajo.

3.2. CLC (Corine Land Cover)

Los **datos dependientes**, es decir, aquellos que el modelo intenta predecir, provienen del **Corine Land Cover (CLC)**, parte del programa **Copernicus Land Monitoring Service**. Se utilizaron los mapas de cobertura del suelo correspondientes a los años **2018** y **2021**, en formato **raster**, con resolución adecuada para análisis territoriales a gran escala.

Estos datos representan **categorías de uso y cobertura del suelo** en toda Europa, y fueron empleados como etiquetas (label) durante el entrenamiento del modelo de clasificación supervisada.

Objetivo del modelo: Predecir estas categorías a partir de datos espectrales e índices derivados de imágenes de satélite Sentinel-2.

4. Preprocesamiento

4.1. Descarga y formatos de entrada

Datos Sentinel-2

Los datos del satélite **Sentinel-2** se descargan en formato **raster .JP2 (JPEG2000)** desde la plataforma Copernicus. Cabe destacar que la descarga no se limita al área de interés especificada por el usuario, sino que corresponde a una **tesela completa de 100 km x 100 km**, según el sistema de teselado utilizado por Sentinel-2.

El sistema de referencia espacial es **WGS 84 / UTM zona 30N (EPSG: 32630)**, aunque varía según la ubicación geográfica del área analizada. Las resoluciones específicas de las bandas utilizadas se detallan en el apartado **3.1**.

Datos CLC

Los datos del **Corine Land Cover (CLC)**, descargados desde el **Copernicus Land Monitoring Service (CLMS)**, se presentan en formato **raster GeoTIFF (.tif)** con una **resolución espacial de 10 metros por píxel**. El sistema de referencia utilizado es **ETRS89 / LAEA Europe (EPSG: 3035)**, adecuado para análisis continentales.

Estos archivos abarcan la totalidad del territorio europeo, lo que implica un **tamaño de archivo considerable** y, por tanto, un mayor requerimiento de recursos para su procesamiento y análisis.

4.2. Conversión de JP2 a TIFF 32-bit float

Las imágenes descargadas de Sentinel-2 vienen en formato **.JP2 (JPEG 2000)**. Aunque este formato es eficiente en términos de compresión y calidad, **muchas librerías de procesamiento geoespacial en Python (como rasterio, GDAL, xarray o numpy) presentan limitaciones o incompatibilidades al trabajar directamente con archivos JP2**, especialmente al realizar operaciones como recortes, reproyecciones o lecturas en bloque.

Para facilitar el flujo de trabajo y garantizar una compatibilidad total, las imágenes fueron convertidas a formato **GeoTIFF**. Además, se especificó que el tipo de datos fuera **32-bit float (coma flotante)**, lo cual es fundamental para evitar pérdida de información en los futuros procesamientos.

Código 4.2, 4.3, 4.4 y 4.5 (ver código ressample_recorte_jp2-tif.py)

4.3. Justificación del cambio de resolución.

Las bandas **B11** y **B12** del satélite Sentinel-2, que corresponden al infrarrojo de onda corta (SWIR), tienen una resolución espacial nativa de **20 metros por píxel**, a diferencia de otras bandas como B02, B03, B04 y B08, que poseen una resolución de **10 metros**.

Para poder integrar estas bandas con el resto de los datos y realizar análisis espectrales coherentes, fue necesario realizar un **cambio de resolución (resampleado)** de **20 m a 10 m**. Este paso permite alinear espacialmente todas las bandas y evitar inconsistencias durante el apilado de capas o el cálculo de índices espectrales multibanda, como el NDVI, NDBI o NDWI.

Además, al mantener una resolución uniforme entre todas las bandas (10 m), se mejora la compatibilidad con los modelos de inteligencia artificial, que requieren **matrices con dimensiones consistentes** como entrada. También se facilita la visualización, el preprocesamiento y el análisis geoespacial conjunto.

4.4. Reproyección de datos

Los datos CLC se encontraban originalmente en el sistema de referencia espacial **ETRS89 / LAEA Europe (EPSG: 3035)**, mientras que los datos Sentinel-2 utilizados en este proyecto están en **WGS 84 / UTM zona 30N (EPSG: 32630)**.

Para garantizar la correcta superposición y alineación de ambos conjuntos de datos, fue necesario realizar una **reproyección del raster CLC al sistema EPSG: 32630**. Este paso es fundamental para asegurar la integridad espacial del análisis, especialmente en tareas como el etiquetado de píxeles, análisis zonal y entrenamiento del modelo de inteligencia artificial.

4.5. Recorte geográfico con GeoJSON

Para preparar los datos de entrada del modelo de inteligencia artificial, fue necesario **recortar todas las capas geoespaciales** utilizando un archivo **GeoJSON** que define con precisión el área de estudio.

Este recorte tiene como principal objetivo asegurar que **todas las bandas y capas de información compartan exactamente la misma extensión espacial**, el **mismo número de filas y columnas (píxeles)** y estén **perfectamente alineadas**. Esta homogeneidad es fundamental para construir matrices multibanda que puedan ser utilizadas por los modelos de aprendizaje automático, donde cada píxel debe representar la misma ubicación y contener valores consistentes en todas las capas.

Además, este proceso evita el procesamiento de zonas irrelevantes fuera del área de estudio, reduciendo el peso de los datos y optimizando el rendimiento computacional.

4.6. Generación de índices espectrales (NDVI, SAVI, etc.)

Como parte del preprocesamiento, se calcularon varios **índices espectrales** a partir de las bandas de Sentinel-2, con el objetivo de **resaltar características específicas del terreno** como la vegetación o el contenido de humedad. Estos índices se derivan de combinaciones matemáticas entre bandas específicas, y resultan fundamentales para enriquecer los datos de entrada del modelo de inteligencia artificial.

La generación de estos índices se realizó mediante **Python**, utilizando librerías como numpy y rasterio, que permiten el manejo eficiente de datos ráster.

A continuación se detallan los principales índices calculados:

Índice	Fórmula	Uso principal
NDVI	$\frac{(B08 - B04)}{(B08 + B04)}$	Vegetación sana
SAVI	$\left(\frac{(B08 - B04)}{(B08 + B04 + 0.5)}\right) \times 1.5$	Vegetación con corrección de suelo
MSAVI	$\frac{\left(2 \times B08 + 1 - \sqrt{(2 \times B08 + 1)^2 - 8 \times (B08 - B04)}\right)}{2}$	Vegetación en zonas áridas
NDWI	$\frac{(B03 - B08)}{(B03 + B08)}$	Humedad y cuerpos de agua
NDBI	$\frac{(B11 - B08)}{(B11 + B08)}$	Zonas urbanas y construidas
BSI	$\frac{((B11 + B04) - (B08 + B02))}{((B11 + B04) + (B08 + B02))}$	Suelo desnudo
GCI	$\left(\frac{B09}{B03}\right) - 1$	Contenido de clorofila

El uso de estos índices no solo mejora la capacidad del modelo para distinguir entre diferentes tipos de cobertura del suelo, sino que también **reduce la dependencia**

exclusiva de las bandas espectrales crudas, aportando **información derivada y más interpretativa** para la clasificación de uso del suelo.

Código 4.6 (ver código calculo_indices.py)

5. Preparación de datos para entrenamiento

5.1. Generación del mapa de cambio (CLC2018*100 + CLC2021)

Se generó un **mapa de cambio de uso del suelo** a partir de los datos CLC de 2018 y 2021 utilizando la fórmula **CLC2018 × 100 + CLC2021**, que permite codificar en un único valor numérico la transición entre clases.

Este mapa fue posteriormente utilizado para crear la columna **label**, que actúa como **etiqueta de clase** en los archivos de entrenamiento. De esta forma, el modelo no solo aprende a clasificar una cobertura del suelo, sino a detectar y predecir **cambios** entre dos periodos temporales.

CLC 2018	CLC 2021	Etiqueta (Label)
2	2	202
3	5	305
10	10	1010

Código 5.1 (ver código calculo_mapa_cambio.py)

5.2. Generación de archivos Parquet

Para el entrenamiento y posterior predicción mediante modelos de inteligencia artificial, los datos geoespaciales procesados fueron convertidos a archivos en formato **Parquet**, un tipo de archivo columnar optimizado para **almacenamiento eficiente y procesamiento rápido**, especialmente útil en flujos de trabajo con grandes volúmenes de datos (casi 15 millones solo para el entrenamiento).

Estructura de los archivos Parquet

Cada archivo Parquet contiene una **tabla estructurada** en la que:

- **Cada fila representa un píxel** del área de estudio.
- **Cada columna corresponde a una variable o característica** (banda espectral, índice espectral o **etiqueta de clase**, según el caso).

Tipos de Parquet

- **Los archivos destinados al entrenamiento** contienen:
  - Bandas espectrales y/o índices como **variables de entrada**.
  - Una columna adicional con la **etiqueta o clase objetivo (label)**, extraída de los datos CLC reproyectados y alineados.
  - Solo se incluyen píxeles dentro del área de interés y con datos válidos (sin nulos).
- **Los archivos generados para predicción** contienen:
  - Las mismas columnas de variables que los archivos de entrenamiento (bandas e índices), pero **sin la columna de etiqueta**.
  - **Su objetivo es ser utilizados como input del modelo ya entrenado, que devolverá una predicción de clase para cada píxel**.

Estos archivos permiten que el modelo aprenda a asociar características espectrales con una clase específica de cobertura del suelo.

Código 5.2 (ver código parquet_entrenadores.py)

(ver código parquet_prediccion.py)

6. Entrenamiento del modelo

6.1. Selección del modelo (XGBoost)

Inicialmente se empleó un modelo **Random Forest**, pero presentó limitaciones importantes al escalar con grandes volúmenes de datos, millones de píxeles, tanto en consumo de memoria que se hacía inviable, como en tiempo de entrenamiento. Además, su rendimiento frente a clases desbalanceadas fue subóptimo, no está diseñada para manejar datasets muy grandes.
Por estas razones, se optó por utilizar **XGBoost**, una implementación altamente optimizada del algoritmo de **gradient boosting**, que ha demostrado ser especialmente eficaz en tareas de clasificación supervisada sobre grandes volúmenes de datos.
XGBoost ofrece ventajas significativas frente a otros modelos, tanto en términos de rendimiento como de flexibilidad. Su arquitectura permite un control detallado sobre el proceso de entrenamiento, incluyendo aspectos como la **profundidad máxima de los árboles**, la **tasa de aprendizaje**, los **parámetros de regularización** y la **ponderación de clases desbalanceadas**. Esta capacidad de ajuste fino fue esencial para adaptar el modelo a las particularidades del conjunto de datos geoespaciales utilizados, donde las clases de cambio en el uso del suelo no solo eran múltiples, sino también desbalanceadas en frecuencia.
Además, su compatibilidad con estructuras de datos eficientes como los archivos

Parquet, y su integración fluida con librerías como pandas, dask y numpy, lo convierten en una herramienta especialmente adecuada para el tratamiento de datasets extensos como el de este proyecto.

6.2. Parámetros utilizados

El modelo XGBoost fue configurado con el objetivo de realizar **clasificación multiclase** sobre los datos de cambio de uso del suelo. Se utilizaron parámetros estándar, seleccionados por su robustez y compatibilidad con conjuntos de datos grandes.

En particular, se empleó el modo 'multi:softprob', que permite obtener una distribución de probabilidad por clase para cada píxel, lo cual facilita análisis adicionales como mapas de incertidumbre. El parámetro tree_method='hist' fue clave para acelerar el entrenamiento sobre millones de muestras, al reducir el consumo de memoria mediante histogramas.

La correcta codificación de las clases mediante **LabelEncoder** aseguró la compatibilidad con el modelo, y la división del conjunto en entrenamiento y validación permitió evaluar el rendimiento inicial de forma objetiva.

Durante la codificación de las etiquetas con LabelEncoder, surgieron problemas debido a que los valores generados con la fórmula CLC2018 × 100 + CLC2021 no eran secuenciales ni uniformes, lo que causó **"huecos"** en la numeración. Se intentó aplicar un orden lógico personalizado (por ejemplo, que 101 fuera 1, 102 fuera 2, etc.), pero esto complicó la gestión y generó inconsistencias. Finalmente, se optó por utilizar LabelEncoder en su forma estándar, permitiendo que asignara identificadores enteros consecutivos automáticamente, garantizando la compatibilidad con XGBoost.

6.3. Script de entrenamiento y comentarios

El entrenamiento del modelo se realizó mediante un script en Python, el cual integró todas las fases necesarias: carga de datos, codificación de etiquetas, división del dataset y entrenamiento del modelo XGBoost. A continuación, se describen las etapas principales del script, junto con observaciones relevantes:

- Carga del dataset:
  Se utilizó un archivo .parquet con millones de muestras ya preprocesadas (bandas, índices y etiquetas). Este formato permitió una lectura eficiente en memoria con pandas.
  `df = pd.read_parquet("…/combined_donana_rivas_filtrado.parquet")`
- Codificación de clases:

Se usó LabelEncoder para transformar las etiquetas de cambio de uso del suelo (generadas con la fórmula CLC2018 × 100 + CLC2021) en índices enteros.

```
encoder = LabelEncoder()
df["label"] = encoder.fit_transform(df["label"])
```

- División del dataset:
  Se dividió el conjunto en **80 % para entrenamiento** y **20 % para validación**, utilizando train_test_split de scikit-learn.

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
```

- Entrenamiento del modelo:
  Se instanció un clasificador XGBClassifier con configuración para clasificación multiclase (multi:softprob) y método de crecimiento de árboles optimizado (hist), ideal para grandes volúmenes de datos.

```
model = xgb.XGBClassifier(
  objective='multi:softprob',
  eval_metric='mlogloss',
  tree_method='hist',
  verbosity=1,
  num_class=n_classes,
  use_label_encoder=False
)
model.fit(X_train, y_train)
```

- Guardado del modelo y del codificador:
  El modelo entrenado y el codificador de etiquetas se guardaron en formato .pkl para su uso posterior en predicción.

```
pickle.dump(model, open("xgboost_model.pkl", "wb"))
pickle.dump(encoder, open("label_encoder.pkl", "wb"))
```

Código 6.3 (ver código Entrenamiento.py)

7. Predicción

Una vez entrenado el modelo XGBoost, se aplicó sobre una **zona geográfica diferente** (Ribarrojas, Valencia) a la utilizada durante el entrenamiento. El objetivo fue evaluar la capacidad del modelo para generalizar sobre nuevos datos, simulando un escenario real de detección de cambios de uso del suelo en zonas no etiquetadas previamente.

7.1. Proceso de predicción sobre zona nueva

Se utilizó un archivo .parquet con la misma estructura de columnas que los utilizados para entrenamiento (bandas, índices y coordenadas), pero **sin la columna de etiquetas**.

- **Carga del archivo de predicción** y del **modelo entrenado** (xgboost_model.pkl) junto al **codificador de etiquetas** (label_encoder.pkl).
- Se aplicó el modelo para predecir la clase (código de cambio CLC) de cada píxel de la nueva zona.
- Las predicciones numéricas (clases codificadas) se decodificaron con LabelEncoder para obtener los valores originales tipo CLC2018*100 + CLC2021.

Este proceso permitió generar un vector de predicciones espaciales listo para reconstruir en forma de imagen raster.

Código 7.1 (ver código predicción.py)

7.2. Conversión de predicción a TIFF

Para facilitar la visualización geográfica de los resultados, se procedió a **convertir el vector de predicción a un archivo raster GeoTIFF**.

Este paso se realizó utilizando las coordenadas x e y de cada píxel, reconstruyendo la malla espacial original en el sistema de referencia EPSG:32630. Cada valor de píxel en el TIFF corresponde a una clase de cambio de uso del suelo predicha por el modelo.

La generación del GeoTIFF se realizó utilizando librerías como rasterio o xarray, permitiendo conservar la **resolución espacial de 10 metros** y la alineación geográfica precisa para su comparación con capas reales o visualización en SIG.

7.3. Aplicación de suavizado espacial

Dado que los modelos de clasificación basados en píxeles pueden generar resultados con **ruido local** (píxeles aislados con clases distintas a su entorno), se aplicó un **suavizado espacial post-procesado** sobre la predicción.

El objetivo fue mejorar la coherencia visual y temática del mapa resultante, reduciendo errores puntuales sin alterar significativamente las estructuras reales.

se empleó un **filtro de mayoría con una vecindad de 3x3 píxeles**, es decir, cada píxel fue reclasificado según la clase más frecuente entre sus 8 vecinos

Códifo 7.3 (ver código Suavizado.py)

8. Evaluación del modelo

Una vez finalizado el proceso de predicción y postprocesamiento (incluyendo el suavizado espacial), se evaluó la calidad del modelo comparando los resultados generados con los datos reales de cambio obtenidos del **CLC** mediante la fórmula CLC2018 × 100 + CLC2021.

8.1. Precisión global

La comparación directa entre la predicción del modelo (ya suavizada) y la capa real de cambios CLC mostró una **precisión global del 36 %**. Es decir, el **36 % de los píxeles** clasificados por la inteligencia artificial coincidieron exactamente con la clase esperada de cambio de uso del suelo.

Esta cifra, aunque moderada, refleja la **dificultad inherente al problema**, ya que:

- Existen **decenas de clases posibles** (transiciones entre pares de usos del suelo).
- El dataset presenta un alto **desbalance**, con ciertas clases mucho más frecuentes que otras.
- Las diferencias espaciales y espectrales entre zonas (entrenamiento vs. predicción) reducen la capacidad de generalización.

8.2. Otros métodos

Además de la precisión global, se utilizaron otras métricas más específicas para evaluar el rendimiento del modelo:

- Matriz de confusión:
  Se generaron matrices de confusión tanto para la predicción original como para la suavizada. Estas matrices revelan una **alta concentración en unas pocas clases** y múltiples errores de clasificación, especialmente en transiciones menos representadas. Esto evidencia que el modelo **generaliza con dificultad en clases poco frecuentes**.
- Métricas por clase:
  A partir del archivo metricas_ia_suavizada.csv, se calcularon métricas clásicas para cada clase:
  - **Precisión (Precision)**: proporción de aciertos sobre las predicciones hechas para una clase.
  - **Exhaustividad (Recall)**: proporción de aciertos sobre todos los casos reales de una clase.
  - **F1-score**: media armónica entre precisión y recall.

- **Support**: número de muestras reales de esa clase.

La evaluación muestra que, si bien el modelo es capaz de detectar algunos patrones generales, su rendimiento aún es limitado, especialmente en contextos con muchas clases, datos desbalanceados y transiciones complejas. Estas observaciones son valiosas para orientar **mejoras futuras.**

Código 8 (ver código prueba_modelo.py)

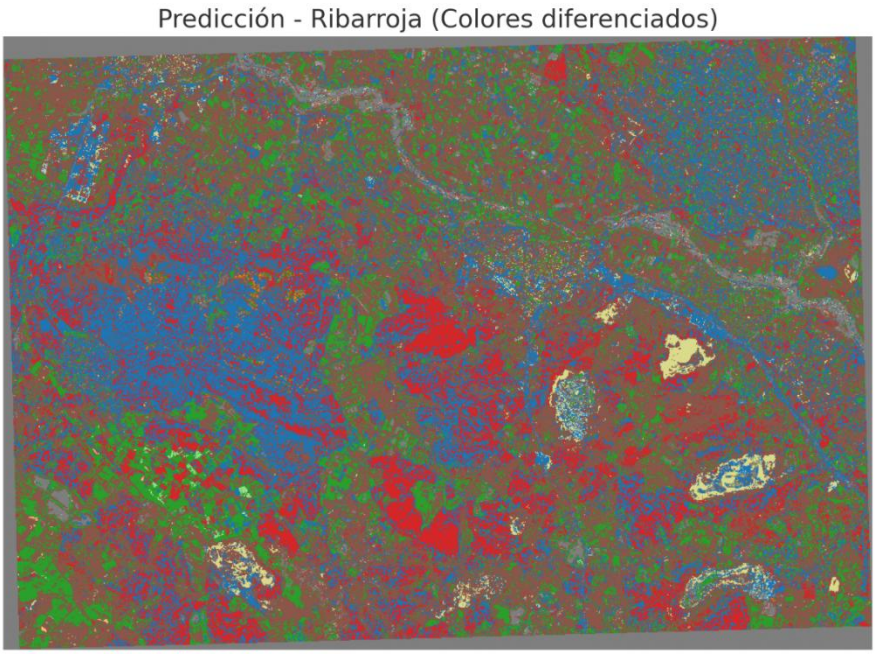
9. Resultados

Una vez entrenado el modelo y aplicada la predicción sobre una zona nueva (Ribarroja), se generaron dos salidas principales:

- Una **predicción original**, obtenida directamente del modelo.
- Una **predicción suavizada**, a la que se aplicó un filtro espacial para reducir ruido y mejorar la coherencia visual.

9.1. Predicción original

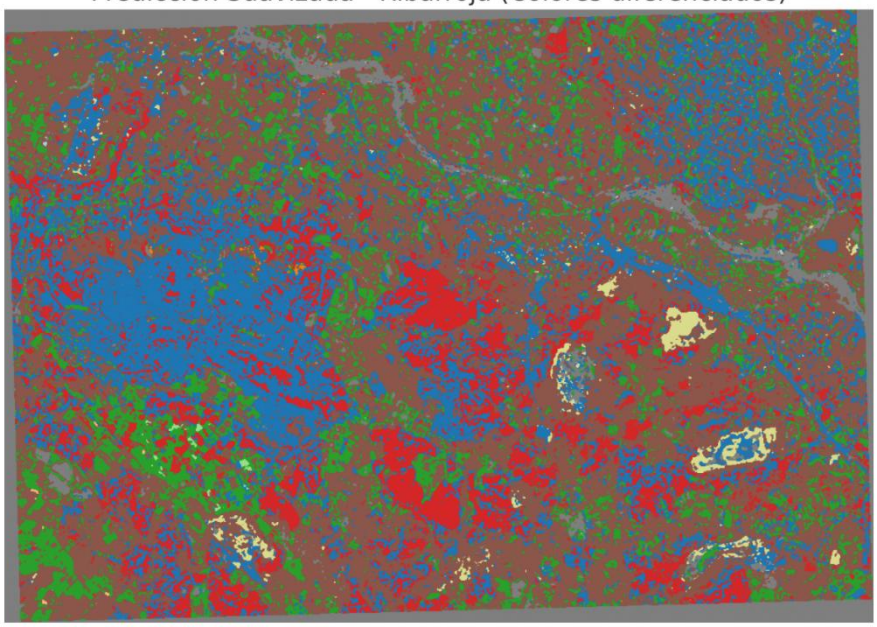
La predicción original corresponde a la clasificación píxel a píxel realizada por el modelo XGBoost sobre la zona de Ribarroja. Esta versión conserva todos los detalles y la granularidad del modelo.



9.2. Predicción suavizada

Para mejorar la legibilidad e interpretación de los resultados, se aplicó un **suavizado espacial mediante un filtro de mayoría 3x3**, el cual reclasifica cada píxel en función de sus vecinos más cercanos.

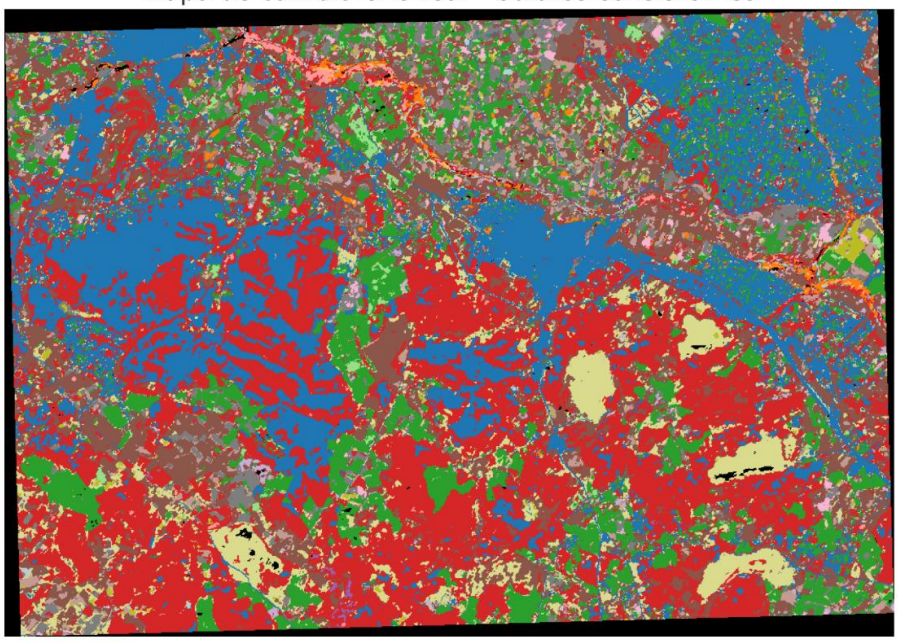
Predicción Suavizada - Ribarroja (Colores diferenciados)



9.3 Datos reales de cambio (CLC 2018–2021)

Como referencia para evaluar la calidad de las predicciones generadas por el modelo, se utilizó el **mapa de cambio real** obtenido a partir de los datos **Corine Land Cover (CLC)** correspondientes a los años **2018 y 2021**.

Mapa de cambio CLC real - Colores consistentes



10. Conclusión

La realización de este proyecto ha supuesto un **aprendizaje significativo** en todas las etapas del flujo de trabajo aplicado a datos geoespaciales e inteligencia artificial. Desde el **tratamiento y preprocesamiento de los datos**, la generación de scripts para automatizar tareas, la estructuración de datos en formato **Parquet** y el **entrenamiento de modelos**, hasta la evaluación cuantitativa y visual de los resultados, cada fase ha contribuido a una comprensión más profunda y aplicada de la integración entre IA y SIG.

En cuanto al modelo desarrollado, si bien se alcanzó una **precisión global del 36 %**, el análisis posterior ha puesto de manifiesto **diversos aspectos mejorables**, tanto a nivel técnico como metodológico. Uno de los principales puntos a revisar ha sido la **selección de las zonas de entrenamiento y predicción**. Las zonas utilizadas (Doñana y Rivas) presentan características muy distintas a la zona objetivo (Ribarroja), lo que ha dificultado la capacidad del modelo para **generalizar correctamente**. A posteriori, habría sido más adecuado:

- Entrenar el modelo con datos más representativos de la zona de predicción.

- O bien, seleccionar una zona de predicción que compartiera condiciones similares con las zonas de entrenamiento.

A pesar de estos errores y decisiones subóptimas —propias de cualquier proceso de aprendizaje práctico— el resultado final ha despertado en mí un **interés aún mayor por el uso de la inteligencia artificial en el ámbito de los SIG**. Este proyecto ha sembrado la base para futuros desarrollos en los que buscaré construir modelos **más robustos, eficaces y eficientes**, aplicables a problemáticas reales de análisis territorial.

Más allá del rendimiento numérico, me quedo con la **satisfacción de haber consolidado conocimientos esenciales** y con la certeza de que los errores cometidos en este proyecto son los que permitirán **mejorar y avanzar** en los próximos.