

# **Análisis comparativo de algoritmos de clasificación de imágenes en el software R para la identificación de cobertura boscosa en el municipio El Dorado (Meta)**

**Carol Tatiana Chicuzaque Gutiérrez**

Trabajo de grado presentado como requisito parcial para optar al título de Especialista en  
Sistemas de Información Geográfica

Universidad de Manizales

Facultad de Ciencias e Ingeniería

Especialización en Sistemas de Información Geográfica

Manizales, 2024

## Resumen

El objetivo de este estudio es determinar el algoritmo de clasificación supervisada con mejores métricas en la identificación de la cobertura boscosa, para el municipio de El Dorado (Meta), actualmente se tienen a disposición diferentes desarrollos tecnológicos enfocados en el Machine Learning para las aplicaciones propias de la teledetección; en el caso de la clasificación supervisada se cuenta con una amplia variedad de algoritmos, sin embargo, estas posibilidades tienden a pasar desapercibidas por el usuario, ya que en la mayoría de softwares de procesamiento esta metodología viene configurada con un algoritmo en particular. Para definir cuál es algoritmo de clasificación supervisada con el que se obtiene un mejor resultado se propone hacer uso de una imagen satelital de Sentinel-2 y del software R para el procesamiento y análisis estadístico; se realiza una selección de cuatro algoritmos de clasificación supervisada, obteniendo resultados de precisión, resultados visuales y tiempos de procesamiento para cada uno de ellos; dando como resultado que es el algoritmo de Red Neuronal Artificial (RNA) con un kappa de 0.96 y un índice de precisión de 0.97 el que mejores métricas tiene y por ende el óptimo para la generación de productos intermedios y finales de caracterización del territorio.

**Palabras clave:** Algoritmos de clasificación, R, Comparación, Clasificación supervisada, Cobertura terrestre, Presición, Sentinel - 2

## Abstract

The objective of this study is to determine the supervised classification algorithm with the best metrics in the identification of forest cover for the municipality of El Dorado (Meta), currently there are different technological developments available focused on Machine Learning for remote sensing applications; in the case of supervised classification there is a wide variety of algorithms, however, these possibilities tend to go unnoticed by the user, since in most processing software this methodology is configured with a particular algorithm. To define which is the supervised classification algorithm with which the best result is obtained, it is proposed to use a Sentinel-2 satellite image and R software for processing and statistical analysis; a selection of four supervised classification algorithms is made, obtaining results of accuracy, visual results and processing times for each of them; resulting in the Artificial Neural Network (ANN) algorithm with a kappa of 0.96 and a precision index of 0.97 the one with the best metrics and therefore the optimal one for the generation of intermediate and final products for the characterization of the territory.

**Keywords: Classification Algorithms, R, Comparison, Supervised Classification, Land Cover, Accuracy, Sentinel - 2**

# Contenido

	<b>Pág.</b>
<b>1. INTRODUCCIÓN .....</b>	<b>9</b>
1.1 PROBLEMA DE INVESTIGACIÓN .....	11
1.2 OBJETIVOS.....	13
1.2.1 <i>Objetivo General</i> .....	13
1.2.2 <i>Objetivos Específicos</i> .....	13
1.3 JUSTIFICACIÓN .....	15
<b>2. MARCO TEÓRICO .....</b>	<b>16</b>
2.1 TELEDETECCIÓN .....	16
2.2 IMAGEN SATELITAL .....	16
2.3 MACHINE LEARNING.....	16
2.4 SENTINEL – 2 .....	17
2.5 CLASIFICACIÓN NO SUPERVISADA.....	17
2.6 CLASIFICACIÓN SUPERVISADA.....	17
2.7 RANDOM FOREST.....	18
2.8 K-NEAREST NEIGHBORS (K-NN).....	18
2.9 RED NEURONAL ARTIFICIAL (RNA).....	19
2.10 MÁQUINA DE SOPORTE VECTORIAL (MSV).....	20
2.11 MATRIZ DE CONFUSIÓN .....	21
2.12 ÍNDICE KAPPA.....	22
<b>3. ESTADO DEL ARTE .....</b>	<b>23</b>
<b>4. MARCO NORMATIVO Y LEGAL.....</b>	<b>26</b>
<b>5. SOFTWARE .....</b>	<b>27</b>
5.1 ARCGIS PRO ® .....	27
5.2 R ®.....	27
<b>6. ZONA DE ESTUDIO .....</b>	<b>28</b>
<b>7. METODOLOGÍA.....</b>	<b>29</b>
7.1 ENFOQUE METODOLÓGICO .....	29
7.2 TIPO DE ESTUDIO.....	30
7.3 DESARROLLO DE LA METODOLOGÍA.....	30
7.3.1 <i>Fase I: Alistamiento de la información</i> .....	30
7.3.2 <i>Fase II: Selección de los algoritmos de clasificación supervisada</i> .....	31
7.3.3 <i>Fase III: Procesamiento</i> .....	32
7.3.4 <i>Fase IV: Interpretación de resultados y generación de productos</i> .....	33
<b>8. RESULTADOS Y GENERACIÓN DE PRODUCTOS.....</b>	<b>35</b>
8.1 COMPARACIÓN DE LAS IMÁGENES ADQUIRIDAS .....	35

V                      Análisis comparativo de algoritmos de clasificación de imágenes en el software R  
para la identificación de cobertura boscosa en el municipio El Dorado (Meta)

---

8.2	DEFINICIÓN DE LAS CLASES O COBERTURAS .....	37
8.3	CLASIFICACIÓN SUPERVISADA CON EL ALGORITMO RANDOM FOREST .....	38
8.4	CLASIFICACIÓN SUPERVISADA CON EL ALGORITMO K-NEAREST NEIGHBORS.....	40
8.5	CLASIFICACIÓN SUPERVISADA CON EL ALGORITMO RED NEURONAL ARTIFICIAL .....	41
8.6	CLASIFICACIÓN SUPERVISADA CON EL ALGORITMO MÁQUINA DE SOPORTE VECTORIAL.....	43
<b>9.</b>	<b>ANÁLISIS DE RESULTADOS .....</b>	<b>45</b>
<b>10.</b>	<b>CONCLUSIONES .....</b>	<b>50</b>
<b>11.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>53</b>

---

## Lista de figuras

	<b>Pág.</b>
<b>Figura 1</b> Funcionamiento KNN .....	19
<b>Figura 2</b> Ubicación zona de estudio.....	28
<b>Figura 3</b> Metodología del proyecto.....	29
<b>Figura 4</b> Comparación visual de las imágenes.....	36
<b>Figura 5</b> Área de entrenamiento.....	38
<b>Figura 6</b> Resultado de clasificación con RF .....	39
<b>Figura 7</b> Resultado de clasificación con K-NN .....	41
<b>Figura 8</b> Esquema de la red neuronal modelada .....	42
<b>Figura 9</b> Resultado de clasificación con RNA.....	43
<b>Figura 10</b> Resultado de clasificación con MSV.....	44
<b>Figura 11</b> Comparación visual zona de bosque No. 1 .....	46
<b>Figura 12</b> Comparación visual zona de bosque No. 2 .....	47

## Lista de tablas

<b>Tabla 1</b>	Características de las escenas adquiridas .....	35
<b>Tabla 2</b>	Coberturas a clasificar.....	37
<b>Tabla 3</b>	Comparación de estadísticas de las 8 repeticiones en RF.....	38
<b>Tabla 4</b>	Matriz de confusión RF .....	39
<b>Tabla 5</b>	Comparación de estadísticas de las combinaciones en K-NN .....	40
<b>Tabla 6</b>	Matriz de confusión K-NN .....	40
<b>Tabla 7</b>	Comparación de estadísticas de las combinaciones en RNA.....	41
<b>Tabla 8</b>	Matriz de confusión RNA.....	42
<b>Tabla 9</b>	Comparación de estadísticas de las combinaciones en MSV .....	43
<b>Tabla 10</b>	Matriz de confusión MSV.....	44
<b>Tabla 11</b>	Estadísticas de los algoritmos .....	45
<b>Tabla 12</b>	Comparación de resultados de la matriz de confusión para la clase 3.....	45
<b>Tabla 13</b>	Comparación de tiempos de ejecución .....	48

---

# Lista de símbolos y abreviaturas

## Abreviaturas

### Abreviatura Término

---

*SIG* Sistema de Información Geográfica

*IGAC* Instituto Geográfico Agustín Codazzi

*GDB* Geodatabase

*RF* Random Forest

*K-NN* K-Nearest Neighbors

*RNA* Red Neuronal Artificial

*MSV* Máquina de Soporte Vectorial



# 1. Introducción

La teledetección, definida desde el campo de las ciencias de la tierra es una técnica cuya finalidad es la captura, el tratamiento y el análisis de las imágenes digitales obtenidas por medio de sensores instalados en plataformas espaciales (Chuvienco, 1995); una de sus aplicaciones con mayor uso es la identificación de cobertura del suelo, datos terrestres que están entre los más importantes y universalmente utilizados al proveer información medioambiental clave en diferentes aplicaciones científicas, siendo cada vez más imperiosa la necesidad de obtener datos exactos, detallados y periódicos (Corts et al., 2010), (Wardlow et al., 2007). La identificación de los cambios de uso y cobertura de la tierra ha sido ampliamente explorada desde las herramientas que conforman la disciplina de la teledetección , al permitir abarcar grandes extensiones en cortos periodos de tiempo y a bajo costo (Nhamo et al., 2018); en la actualidad se tiene a disposición una extensa variedad de imágenes tanto satelitales (Yang et al., 2018) como capturadas con vehículos aéreos no tripulados y diferentes algoritmos de clasificación (Shen et al., 2018) que junto al impulso de la inteligencia artificial permiten la generación de resultados de alto impacto en el análisis de coberturas (Utiltech, 2018).

Sin embargo, la implementación de estos algoritmos de clasificación se encuentra limitada en la mayoría de los software de procesamiento, donde las opciones disponibles son restringidas o, en algunos casos, se reducen a una única opción predefinida; a pesar de esto, encontramos varios autores que a nivel académico han abordado esta temática con el fin de tener mejores rendimientos y precisiones en los procesos de clasificación (Willington et al., 2013), (Ávila-Pérez et al., 2020), (Chucos & Vega, 2022), (Suárez et al., 2017); muchos de estos enfocándose como es el caso del

presente estudio en un tipo de cobertura del suelo en particular. El interés por determinar los beneficios de un algoritmo de clasificación sobre otros, se fundamenta en el hecho de que no todos los algoritmos de clasificación son válidos para abordar cualquier problema o zona, esto debido a la naturaleza de los datos. Los algoritmos pueden ser más o menos efectivos dependiendo el campo de aplicación y así mismo será más o menos complejos; esto no significa que algún algoritmo sea mejor a otro, simplemente que tienen características particulares que se ajustan en mejor medida a los datos a procesar.

El propósito de este documento es la determinación del algoritmo con mejor rendimiento en términos de precisión, calidad y tiempo, para la identificación de la cobertura boscosa en El Dorado, municipio ubicado en los Llanos Orientales de Colombia; para ello se propone una metodología de comparación estadística haciendo uso del software R de cuatro algoritmos de clasificación supervisada, de tipo multiclase, random forest, vecinos más cercanos, red neuronal artificial y máquina de soporte vectorial. Teniendo como insumo una imagen multiespectral del sensor Sentinel – 2 con una resolución espacial de 10 metros; buscando obtener un insumo fiable para la caracterización del territorio y la toma de decisiones en el mismo.

## 1.1 Problema de investigación

La clasificación supervisada es una técnica propia de la teledetección que cuenta con diversas aplicaciones en campos de investigación asociados al conocimiento del territorio por el tipo de información que brinda; es aplicada, por ejemplo, en la cartografía, el monitoreo de recursos naturales, el análisis de cambios en biodiversidad y el monitoreo de cultivos en búsqueda de infecciones (Suárez et al., 2017). En la literatura se encuentra que esta técnica tiene su inicio en los primeros años de la década de los 70 (Alshari & Gawali, 2021) y que a partir de la fecha se han venido incorporando avances en su ejecución principalmente con el desarrollo de diferentes algoritmos utilizados en el proceso de clasificación.

Actualmente, se tiene un escenario de gran disponibilidad de datos y tecnología; se cuenta con una amplia gama de satélites y sensores ópticos que generan imágenes de la tierra con diversas características a nivel de calidad de la imagen, cantidad de área cubierta y temporalidad. Esto combinado con el amplio avance tecnológico y de conocimiento, donde, el desarrollo de software especializado en el procesamiento digital de imágenes y la ejecución de diferentes algoritmos de clasificación; permiten un análisis de territorio más profundo y desde diferentes áreas de interés. Teniendo estas opciones se debe tener presente la importancia de proporcionar información fiable y que cumpla con los diferentes estándares de calidad; sin embargo, esta fiabilidad puede variar tanto por la imagen como por el algoritmo utilizados (Ningthoujam et al., 2016).

Hoy, uno de los temas con mayor interés por parte de los investigadores y la sociedad en general es la aplicación de la inteligencia artificial (AI, por sus siglas en inglés) en la ejecución de diferentes tareas, entre ellas, la generación de cartografía básica. Colombia, está en proceso de actualización catastral con fines multipropósito y uno de sus insumos primordiales es la cartografía básica particularmente, las bases de datos vectoriales; que en su mayoría son generadas de manera manual ya sea mediante captura o restitución. Uno de los objetos de esta base de datos es el bosque, que corresponde a las zonas boscosas del municipio que se esté cartografiando; automatizar el proceso de captura de este elemento puede significar disminución en tiempos de producción, pero para ello es necesario que el proceso de clasificación supervisada se realice con la mayor precisión posible.

Lo expuesto anteriormente, lleva a plantear el interrogante, ¿Cuál es el algoritmo de clasificación más adecuado para la identificación de la cobertura boscosa del municipio de El Dorado (Meta) usando como insumo una imagen satelital de Sentinel 2?

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Determinar el algoritmo de clasificación supervisada más adecuado en el software R para la identificación de la cobertura boscosa en El Dorado–Meta.

### **1.2.2 Objetivos Específicos**

Identificar los diferentes algoritmos de clasificación supervisada aplicados en imágenes aéreas implementables en el software R.

Entrenar de manera óptima los algoritmos de clasificación supervisada seleccionados para el ejercicio de comparación.

Evaluar el rendimiento de los diferentes algoritmos de clasificación utilizados, tanto a nivel del resultado proporcionado como a nivel del gasto computacional.

### 1.3 Justificación

La teledetección, tiene diferentes métodos y herramientas que permiten que el análisis del territorio se pueda abordar desde la estadística y la cuantificación de aquellos atributos que son de carácter cualitativo, este es el caso de la clasificación supervisada, donde a partir de una imagen ya sea satelital u ortoimagen se realiza un entrenamiento de clases definidas por el usuario tanto en cantidad como en características visuales y el software replica la clasificación en la totalidad de la zona de estudio. Hoy se tienen diferentes softwares de escritorio o herramientas en la web con un entorno para generar clasificaciones supervisadas, pero en muchos casos el algoritmo con el que se realiza este proceso ya se define por defecto y ya que usuarios se omiten las demás posibilidades y se desconocen sus beneficios o aportes en el resultado final.

El éxito de un proceso de clasificación supervisada depende de la capacidad del usuario de realizar una correcta definición de clases, y del insumo para realizar el ejercicio; este último depende de la zona de estudio, donde zonas con alto grado de fragmentación o pendientes muy pronunciadas, tendrán mayor presencia de píxeles de incertidumbre principalmente en coberturas y la boscosa. Por esto, se plantea determinar el algoritmo óptimo para clasificar zonas boscosas y se determina usar el software R para el análisis por su gran potencial estadístico.

## **2. Marco Teórico**

### **2.1 Teledetección**

La teledetección (también denominada percepción remota o remote sensing en inglés) es la ciencia y el arte de obtener información sobre un objeto, área o fenómeno a través del análisis de los datos adquiridos por un dispositivo que no está en contacto con el objeto, área o fenómeno que se está investigando (Lillesand et al., 2015). Usando varios sensores, se recopilan datos de forma remota de los cuales muchos pueden ser analizados para obtener información sobre los objetos, áreas o fenómenos que se están investigando (Lillesand et al., 2015).

### **2.2 Imagen satelital**

Es la representación visual de la información capturada por un sensor incorporado en un satélite artificial, estos sensores están encargados de recoger la información que es reflejada por la superficie de la tierra, para luego ser procesada de la manera más conveniente (UCH, 2018).

### **2.3 Machine Learning**

Es una técnica que se asocia a la detección automática de patrones dentro de un conjunto de datos. Se ha convertido en una herramienta común en todas las tareas que requieren extraer información con muchos datos. Existen diferentes tipos de aprendizaje: supervisado, no supervisado, reforzado y profundo, donde cada uno tiene unas características y algoritmos determinados, que se usan en uno u otro dependiendo del problema (Shalev-Shwartz & Ben-David, 2014).

## **2.4 Sentinel – 2**

Los Sentinel son una flota de satélites de la Comisión Europea y hacen parte de un programa de vigilancia medioambiental que tiene como finalidad proporcionar información de utilidad para las prácticas agrícolas y forestales; Sentinel -2, cuenta con una cámara multispectral de alta resolución, con 13 bandas espectrales y un campo de visión que abarca 290 kilómetros de anchura, ofreciendo no solo continuidad sino también una amplia capacidad de generación de datos de la cobertura de la tierra. Sentinel - 2 fue lanzado en su versión 2A en 2015 y en su versión 2B en 2017 (ESA, 2015).

## **2.5 Clasificación no supervisada**

La técnica de clasificación no supervisada hace uso de mecanismos de agrupación, “clustering”, para agrupar píxeles de imágenes de satélite sin asignarles ninguna etiqueta. Posteriormente, el analista asigna las etiquetas a los clusters y genera el producto de la clasificación (Abburu & Golla, 2015).

## **2.6 Clasificación supervisada**

La técnica de clasificación supervisada requiere del análisis previo del usuario, en este caso el analista aporta lo que se conoce como conjunto de entrenamiento, estas muestras de entrenamiento son el factor más importante en la clasificación supervisada de imágenes de satélite. La precisión de los productos generados depende en gran medida de las muestras tomadas para el entrenamiento; las muestras de entrenamiento son de dos tipos, el primero se toma para la clasificación y el segundo para supervisar la precisión de esta (Abburu & Golla, 2015).



## 2.7 Random Forest

Los árboles de decisión conocidos por su nombre en inglés, es de los métodos de clasificación con mayor uso (Breiman et al., 1984). Es un método no paramétrico, robusto y fácil interpretación; su funcionamiento se basa en una serie de particiones sucesivas en el espacio de variables en búsqueda de la variable y el valor de umbral que maximicen la homogeneidad de las particiones que se tengan como resultado. Esta homogeneidad se mide generalmente con el índice de Gini:

$$G = \sum_{k=1}^K f_k * (1 - f_k) \quad i$$

El principal problema que tienen los árboles de decisión es que son muy sensibles a las pequeñas variaciones en los datos de entrada. El algoritmo de Random Forest utiliza dos parámetros: la cantidad de árboles y el número de predictores a usar, sin embargo, una de las grandes ventajas de este algoritmo es su baja sensibilidad al cambio de estos parámetros, por lo cual los valores que generalmente se tienen por defecto tienden a generar buenos resultados (Liaw & Wiener, 2002) (Cánovas-García et al., 2016).

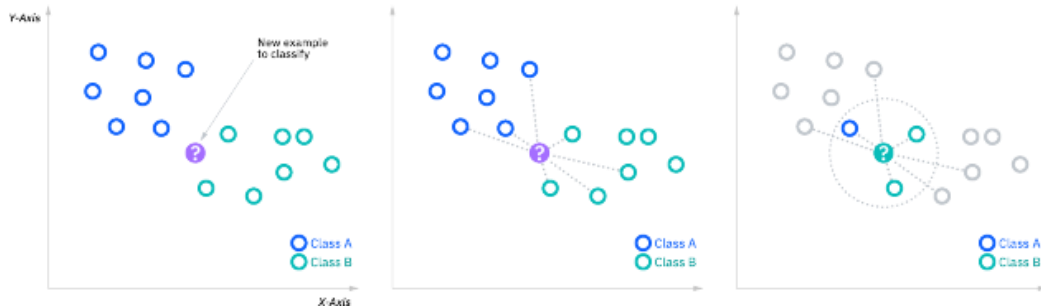
## 2.8 K-Nearest Neighbors (K-NN)

El algoritmo de k vecinos más cercanos, es un clasificador de aprendizaje supervisado no paramétrico, que hace uso de la proximidad para hacer clasificaciones o predicciones sobre la

---

<sup>i</sup> Donde; k es cada una de las clases, K es el total de clases presentes y  $f_k$  es la proporción de los casos en la partición que pertenecen a la clase k.

agrupación de un punto de datos individual, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno de otro (IBM, s.f.).



**Figura 1** Funcionamiento KNN

Fuente: (IBM, s.f.)

El funcionamiento de este algoritmo es dependiente del cálculo de las distancias entre el dato a clasificar y los datos de entrenamiento, seleccionado los  $k$  elementos con menos distancia y dependiendo de la dominancia, se realiza la asignación a la clase o etiqueta correspondiente. Para este cálculo, se utiliza la Distancia Euclidiana para la determinación de la trayectoria más corta entre dos puntos, siguiendo la siguiente ecuación:

$$dx_1x_2 = \sqrt{\Delta x_1^2 + \Delta x_2^2} \quad \text{ii}$$

## 2.9 Red Neuronal Artificial (RNA)

Las redes neuronales artificiales, conocidas también como, sistemas conexionistas; son sistemas implementados para el procesamiento de información con una estructura y funcionamiento inspirados en las redes neuronales biológicas. Basado en un conjunto de elementos simples de procesamiento, denominados nodos o neuronas que se encuentran

---

ii Donde  $\Delta x_1$  es un vector de los datos de entrenamiento y  $\Delta x_2$  es un vector de las etiquetas de las clases.

conectadas entre sí; donde estas conexiones tienen un valor numérico modificado llamado peso.

La actividad de una neurona artificial en un sistema es simple, que suele sumar los valores recibidos de otras neuronas como datos de entrada, comparar la cantidad con el valor umbral y, si se iguala o supera, enviar un dato de salida a las neuronas a las que se esté conectada. Tanto de las entradas como salidas enviadas por las neuronas dependen del peso o fuerza de las conexiones por las que se realizan estas operaciones (Montaño, 2002).

## **2.10 Máquina de Soporte Vectorial (MSV)**

Las MSV se definen como un conjunto de algoritmos de aprendizaje supervisado desarrollado por Vapnik & Cortés en 1995 y el equipo AT&T (Murillo R. A., 2020). Es un clasificador con fuertes bases en el campo del entrenamiento estadístico y ha encontrado un amplio rango de aplicaciones en el reconocimiento de patrones, clasificación de imágenes, predicción financiera en series de tiempo, reconocimiento facial, análisis de señales biomédicas, diagnósticos médicos y minería de datos (Hwang & Chiang, 2010). Pertenece a la familia de los clasificadores lineales y su principal objetivo es obtener una superficie que es capaz de separar las diferentes clases en las que se puede agrupar una distribución de datos en el espacio N-dimensional, utilizando un proceso de optimización basado en la obtención de vectores los cuales definen los límites de las clases (Rojas, 2009). El éxito de este sistema de aprendizaje se debe a que posee las siguientes ventajas: i) cuenta con una fundamentación matemática sólida, ii) está basada en el concepto de minimización de riesgo estructural, minimizando la probabilidad de una clasificación errónea en nuevos ejemplos, iii) dispone de potentes herramientas y algoritmos para la solución óptima y eficiente (Riaño et al., 2016).

## 2.11 Matriz de confusión

Es una herramienta que permite evaluar el desempeño de un algoritmo de clasificación supervisada, donde cada columna representa el número de predicciones de cada clase y cada fila representa las instancias en la clase real. Esta matriz permite ver qué tipo de aciertos y errores tiene el modelo en lo relacionado al ajuste con los datos observados. En la matriz de confusión, se tienen cuatro opciones:

Verdadero positivo (VP): Cantidad de positivos que se clasificaron correctamente como positivos en el modelo.

Verdadero negativo (VN): Cantidad de negativos que se clasificaron correctamente como negativos en el modelo.

Falso negativo (FN): Cantidad de positivos que se clasificaron incorrectamente como negativos.

Falso positivo (FP): Cantidad de negativos que se clasificaron incorrectamente como positivos (Godoy, 2021).

En el caso de las imágenes satelitales, esta matriz se construye con N celdas clasificadas en M clases, sobre las columnas se ordenan las clases reales y sobre las filas las unidades cartográficas o de mapa. Los elementos sobre la diagonal indican el número de clasificaciones realizadas correctamente y los que aparecen fuera se entienden como las migraciones o fugas. Existen dos tipos de errores:

Errores de omisión (riesgo del usuario): Elementos que perteneciendo a esa clase no aparecen en ella por estar incluidos de manera errónea en otra. Están por debajo de la diagonal principal de la matriz.

Errores de comisión (riesgo de productor): Elementos que no perteneciendo a una clase aparecen en ella. Están por encima de la diagonal principal de la matriz.

La matriz de confusión facilita la detección de errores, permite realizar un análisis descriptivo y analítico, da una visión general de las asignaciones correctas y las equivocaciones y da la posibilidad de analizar los resultados desde diferentes niveles (Sánchez, 2016).

## 2.12 Índice Kappa

Propuesto por Cohen en 1960, es un índice utilizado para la evaluación de la concordancia de los métodos que tienen un resultado de tipo categórico involucrando dos o más clases; este índice representa la proporción de aciertos observados respecto al máximo de aciertos posibles más allá del azar. Desde la interpretación del índice kappa hay que tener presente que el índice tiene una dependencia con el nivel de acierto observado. La ecuación mediante la cual se describe kappa es la siguiente:

$$K = \frac{P_0 - P_e}{1 - P_e} \quad \text{iii}$$

En 1977 Landis y Koch propusieron la siguiente escala de valoración: < 0.00 sin aciertos, entre 0.00 y 0.20 insignificante, entre 0.21 y 0.40 mediano, entre 0.41 y 0.60 moderado, entre 0.61 y 0.80 sustancial, entre 0.81 y 1.00 casi perfecto (Murillo R. A., 2020).

---

<sup>iii</sup> Donde  $P_0$  es la proporción de aciertos observados y  $P_e$  es la proporción de aciertos esperados por azar.

### 3. Estado del arte

La eficiencia de los algoritmos de clasificación supervisada ha sido objeto de análisis de diferentes autores, estos documentos no solo sirven como antecedente, sino que además han permitido robustecer la cantidad de algoritmos a contemplar en el proceso de análisis comparativo que se plantea desarrollar.

Las primeras investigaciones basadas en la comparación de algoritmos se centraban en la comparativa entre la clasificación no supervisada y la clasificación supervisada; es el caso de (Sá et al., 1996) donde se evaluó de manera cuantitativa la exactitud de ambos procedimientos; el no supervisado haciendo uso de K Media y el supervisado haciendo uso de Maxver, en este caso la exactitud de las clasificaciones se midió con un análisis estadístico multivariante discreto, “kappa” y el resultado indico que el método supervisado es más eficaz para la discriminación de las formaciones presentes en la zona y por consecuencia para la elaboración de la cartografía en escala 1:100.000.

En los últimos años, se encuentran estudios enfocados en una comparación más extensa de los algoritmos; en (Al-Ahmadi & Hames, 2009) donde se tiene como área de estudio las zonas áridas de Arabia Saudí, se hace uso de cuatro técnicas de clasificación diferentes no supervisada (ISODATA), y supervisada (Máxima Verosimilitud, Distancia de Mahalanobis y Distancia Mínima); la conclusión presentada por los autores, es que el método de máxima verosimilitud ofrece los mejores resultados y que tanto la distancia mínima como el método de Mahalanobis son los más eficaces; sin embargo, estos sobrestiman los terrenos agrícolas y las zonas suburbanas.

En estudios como (Del Toro et al., 2015) se comparan dos algoritmos de clasificación supervisada, Random Forest y SMAP (método de clasificación contextual) y se realiza la validación por el método de máxima verosimilitud, como resultado de este análisis se tiene que tanto Random Forest como SMAP obtienen resultados similares, sin embargo, el segundo obtiene mejores porcentajes de acierto. Otros autores que abordaron la comparación de algoritmos fueron Nolasco, Bocco y Scavuzzo (Nolasco et al., 2017) enfocando su estudio en el mapeo de cultivos agrícolas en Córdoba; compararon algoritmos de Máxima Verosimilitud, Support Vector Machine y Random Forest, dando como conclusión la conveniencia del algoritmo de máxima verosimilitud, que presenta excelentes resultados, siendo un algoritmo de ejecución simple y con un bajo costo computacional.

En diferentes análisis comparativos se evidencia el uso de imágenes de los satélites Landsat-8 y Sentinel-2 como insumo del procesamiento; en (Ávila-Pérez et al., 2020) se realiza la comparación de cuatro algoritmos de clasificación y dos tipos de imágenes satelitales en búsqueda de la determinación del tipo de algoritmo e imagen satelital que permite obtener un resultado más fiable en la identificación de las cobertura boscosa en zonas donde se presenta una alta fragmentación, el análisis permite determinar que las imágenes de Sentinel-2 tienen un mejor desempeño y se identifica que la temporada del año en la cual son capturadas las imágenes afecta el resultado; finalmente, se concluye que la mayor fiabilidad en la identificación de la cobertura boscosa se logra mediante el uso de los algoritmos de Máxima Verosimilitud, Máquinas de Vectores Soporte y Redes Neuronales empleando imágenes Sentinel-2 tomadas en la temporada seca.

Siguiendo esta misma línea de comparativos y análisis encontramos estudios como (Chugh et al., 2020), (Serra, 2020) y (Chucos & Vega, 2022) donde en el caso del primero, se tiene una comparación de algoritmos clásicos de aprendizaje automático: K-Nearest Vecino (KNN), Perceptrón Multicapa (MLP) y clasificador Random Forest (RF); como resultado se tiene que MLP tiene la mayor precisión del 89,57%, seguido de RF con una precisión del 89,2% y, por último, un modelo KNN con una precisión del 85,87%. En el segundo estudio, se tiene una comparativa de algoritmos de clasificación supervisada basa en árboles de decisión como lo son: árbol de decisión, Bagging, RF y AdaBoost, obteniendo que mejor modelo es el RF, seguido muy de cerca del Bagging y finalmente, en el tercer estudio, se tiene una comparación de algoritmos ejecutados en diferentes entornos, permitiendo concluir que el algoritmo SVM y el algoritmo Naive Bayes presentaron una precisión de 0.909 en comparación con el algoritmo de árbol de decisión con 0.864.



## 4. Marco normativo y legal

**Ley 1450 de 2011:** *“Por la cual se expide el Plan Nacional de Desarrollo, 2010 - 2014”*

Establece que el Estado colombiano tiene la facultad de regular el uso de imágenes satelitales.

Esta ley establece que las imágenes satelitales deben usarse de manera responsable y respetar la privacidad de las personas.

## 5. Software

### 5.1 ArcGIS Pro ®

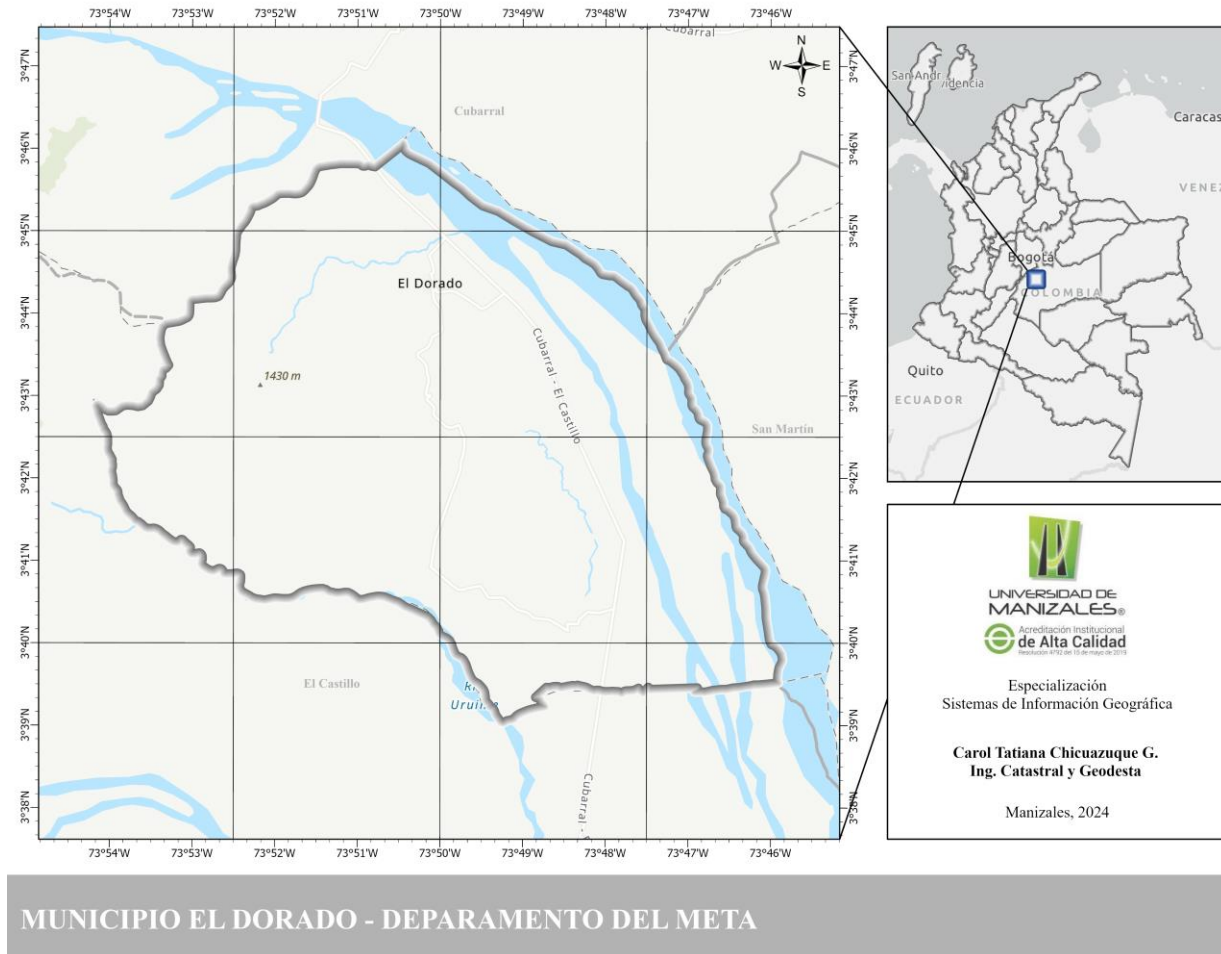
Es un completo sistema que permite recopilar, organizar, administrar, analizar, compartir y distribuir información geográfica; es considerada como la plataforma líder mundial para la creación y uso de sistemas de información geográfica (SIG). Permite la publicación de la información geográfica para que esté accesible para cualquier usuario, está disponible en cualquier lugar por medio de navegadores Web, dispositivos móviles de smartphones y equipos de escritorio (Esri, s.f.). Como este software es privado, para este proyecto se hace uso de la licencia académica de la Universidad de Manizales.

### 5.2 R ®

El paquete estadístico R, un entorno informático estadístico con herramientas de análisis de datos y generación de gráficas, es un software libre que funciona bajo Windows, MAC OS y Linux; es ideal para la docencia, pues permite el uso de todas las herramientas disponibles. Nace como una iniciativa de crear una implementación libre del lenguaje de programación S (utilizado en estadística, bioestadística, matemáticas financieras, entre otros). Dado que R es un lenguaje de programación, el paquete que se descarga sólo incluye las librerías; para manejar esa aplicación se recomienda la instalación de un interfaz amigable (Universidad de Zaragoza, s.f.).

## 6. Zona de estudio

El área de estudio es el municipio de El Dorado, ubicado en el departamento del Meta. El Dorado se encuentra en la cuenca alta del Río Ariari, en el sector noroccidental del departamento del Meta y forma parte del parque nacional de Sumapaz, con 1.143,7 hectáreas de este sistema geográfico. Se selecciona este municipio como área de estudio por sus características a nivel de coberturas terrestres, es una zona con presencia de bosques, cultivos, zona urbana y cuerpos de agua; lo cual es ideal para el análisis de resultados en un proceso de clasificación supervisada.

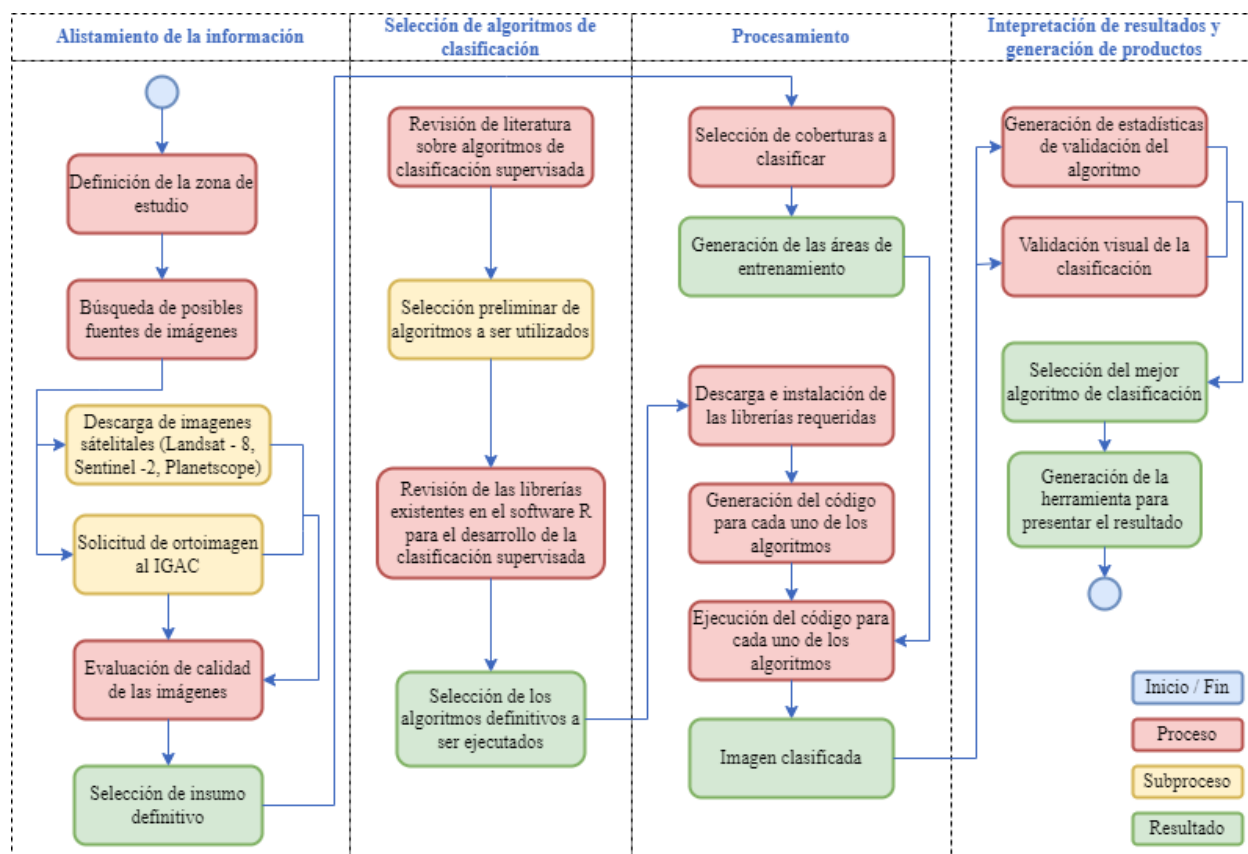


**Figura 2** Ubicación zona de estudio

Fuente: Elaboración propia. Datos IGAC 2024. ArcGIS Pro

## 7. Metodología

La metodología que se define para este proyecto consta de 4 fases desarrollables de manera consecutiva y las cuales agrupan una serie de actividades. En la figura 3, se muestran estas actividades y en el numeral 7.3 se describen de manera más detallada.



**Figura 3** Metodología del proyecto

Fuente: Elaboración propia. Draw.io

### 7.1 Enfoque metodológico

Este proyecto tiene un enfoque cualitativo, en el cual se hace tratamiento de datos medibles desde la estadística y análisis espacial y se tiene un producto final el cual va a tener una evaluación cualitativa que valide su funcionamiento y servicio a los interesados.

## 7.2 Tipo de estudio

El proyecto tiene un estudio de exploratorio y correlacional, donde se propone hacer un tratamiento de diferentes datos espaciales y disponerlos en un solo entorno que permita un análisis simultáneo para la mejor toma de decisiones.

## 7.3 Desarrollo de la metodología

Para determinar el mejor algoritmo de clasificación en el software R para identificar cobertura boscosa en El Dorado–Meta. Se ejecutan las siguientes actividades de manera consecutiva, donde cada actividad realizada da continuidad a las posteriores:

### 7.3.1 Fase I: Alistamiento de la información.

*Actividad 1.1. Definición de la zona de estudio:* La selección de la zona en proyectos de este tipo es importante, pues en la disponibilidad de información a nivel geográfico depende directamente de la zona de interés. El municipio de El Dorado (Meta), cumple con tener presencia de diferentes coberturas y estar ubicado en una zona del país que por sus condiciones atmosféricas permite tener fotografías sin nubosidad en diferentes épocas.

*Actividad 1.2. Búsqueda de posibles fuentes de imágenes:* En la actualidad se cuentan con diferentes fuentes de imágenes del territorio, tanto de satélite como de vehículo aéreo no tripulado. Se tiene disponibilidad de diferentes sensores que ofrecen a su vez productos con especificaciones particulares a nivel de resoluciones principalmente espacial y temporal. En esta fase se realiza la descarga de imágenes de los satélites Landsat – 8, Sentinel – 2 y PlanetScope y

se realiza solicitud de la ortoimagen rural del municipio generada por el Instituto Geográfico Agustín Codazzi – IGAC.

*Actividad 1.3. Evaluación de la calidad de las imágenes:* Una vez descargadas las imágenes se evalúa la resolución espectral y la resolución espacial de cada una, así como los procesos de corrección que han tenido y el peso del archivo; estas características fueron consideradas en la selección del mejor insumo para el procesamiento.

### **7.3.2 Fase II: Selección de los algoritmos de clasificación supervisada.**

*Actividad 2.1. Revisión de literatura sobre algoritmos de clasificación supervisada:* Para determinar los algoritmos existentes para el proceso de clasificación supervisada, se realiza una revisión de la bibliografía existente, que permite tener un listado preliminar de los posibles algoritmos a utilizar.

*Actividad 2.2. Revisión de las librerías existentes en el software R para el desarrollo de clasificación supervisada:* El software R cuenta con una amplia variedad de librerías estadísticas y graficas que permiten ejecutar los diferentes algoritmos de clasificación supervisada y su validación; en esta fase se consideró importante revisar la disponibilidad y requerimientos de estas para determinar el listado definitivo de algoritmos a implementar.

### 7.3.3 Fase III: Procesamiento.

*Actividad 3.1. Selección de coberturas a clasificar:* El primer paso para realizar una clasificación supervisada es la definición de las clases identificadas y en las cuales se pretende clasificar la imagen. Para esta clasificación se tiene en cuenta la resolución espacial del insumo que a su vez define el nivel de interpretación que se tiene.

*Actividad 3.2. Descarga e instalación de las librerías requeridas:* Una vez definidas las librerías necesarias para la ejecución de los algoritmos de clasificación supervisada, se realiza la descarga e instalación de las mismas, este procedimiento se lleva a cabo desde el mismo entorno de R Studio.

*Actividad 3.3. Generación del código para cada uno de los algoritmos:* R es un software que maneja un lenguaje de programación interpretado y que al ser software libre cuenta con diferentes repositorios que facilitan la construcción del código. En esta fase se realiza una búsqueda en los diferentes repositorios de usuarios como RPub y GitHub para diseñar el código de cada uno de los algoritmos de clasificación supervisada.

*Actividad 3.4. Ejecución del código para cada uno de los algoritmos:* Con los códigos construidos y validados en cuanto a funcionamiento, se realiza la ejecución de los mismos; el resultado de cada algoritmo es una imagen clasificada.

### **7.3.4 Fase IV: Interpretación de resultados y generación de productos**

*Actividad 4.1. Generación de estadísticas de validación del algoritmo:* Haciendo uso de los códigos existentes para la construcción de matriz de confusión y cálculo del índice “kappa”, se realiza la validación estadística de cada uno de los algoritmos de clasificación los cuales sirven para la elección del algoritmo más apropiado en el objetivo del proyecto.

*Actividad 4.2. Validación visual de la clasificación:* Una de las maneras de verificar los beneficios de cada uno de los algoritmos de clasificación es de manera visual, permitiendo diferenciar las bondades para cada cobertura. Este análisis se considera pertinente en la comparación planteada y la posterior selección del “mejor” algoritmo de clasificación supervisada.



## 8. Resultados y generación de productos

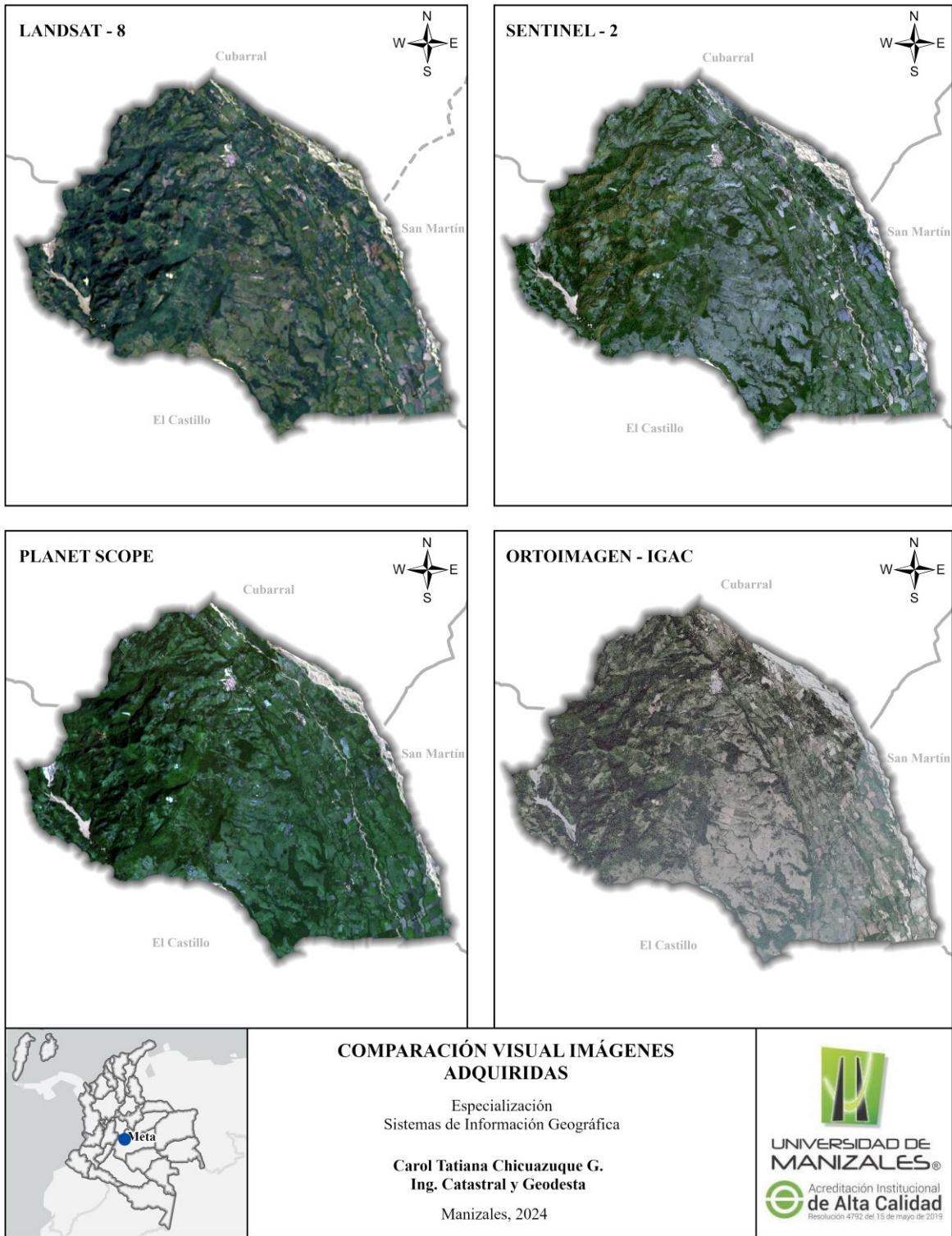
En este capítulo se muestran los resultados obtenidos luego llevar a cabo todas las actividades planteadas en la metodología, se tiene como producto final, cuatro imágenes clasificadas con diferente algoritmo y sus respectivas estadísticas de validación.

### 8.1 Comparación de las imágenes adquiridas

Para la selección del mejor insumo se realiza una comparación de características de cada una de las escenas adquiridas, igual que la capacidad interpretativa que brindan. En la tabla 1 se muestra el resumen de esta comparación y en la figura 4, su despliegue visual. Este análisis permitió determinar que el mejor insumo en este caso es la imagen del sensor Sentinel – 2.

**Tabla 1** Características de las escenas adquiridas

Producto	LC08_L2SP_007057_20240106_...	S2B_MSIL2A_20240123T152649_...	L15 [0604E, 0603E] [1045N, 1044N]	Orto100_50270_20220103
<i>Sensor</i>	Landsat - 8	Sentinel - 2	PlanetScope	World View y Jilin JL-1KF01
<i>Fecha de captura</i>	06 de enero de 2024	23 de enero de 2024	Mosaico enero 2024	03 de enero de 2022
<i>Resolución espacial</i>	30 mts	10 mts	4.7 mts	0.5 mts
<i>Resolución radiométrica</i>	16 bits	16 bits	16 bits	8 bits
<i>Cantidad bandas utilizadas</i>	7	4	5	3
<i>Correcciones</i>	Nivel L2	Nivel L2	Sin información	Radiométrica
<i>Peso del archivo cortado</i>	3.07 MB	15.63 MB	173.90 MB	2.51 GB
<i>Adquisición</i>	Earth Explorer - USGS	Earth Explorer - USGS	NICFI Program - Planet	IGAC



**Figura 4** Comparación visual de las imágenes

Fuente: Elaboración propia. Datos IGAC, USGS, NICFI. ArcGIS Pro.

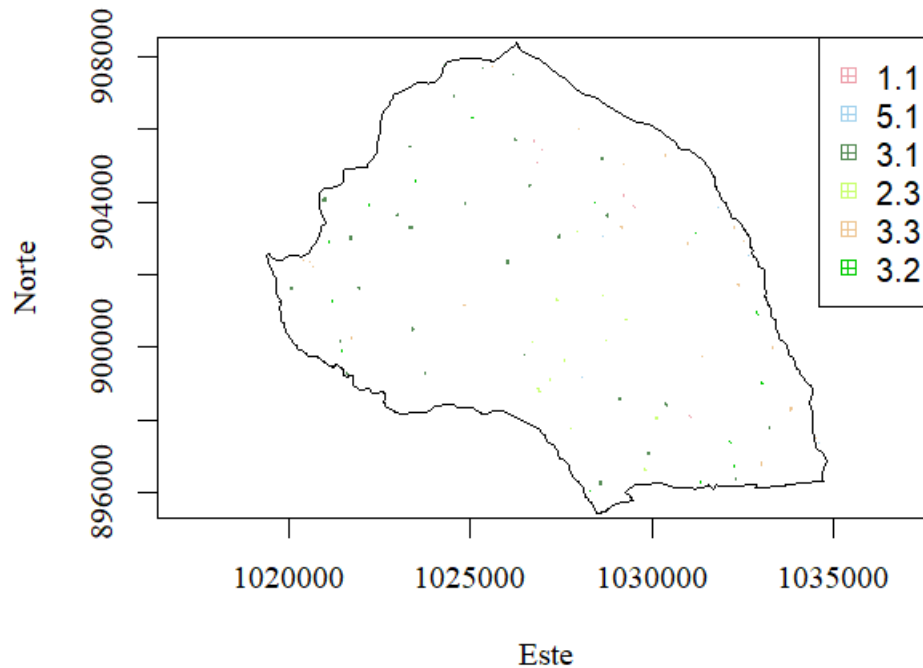
## 8.2 Definición de las clases o coberturas

Uno de los primeros pasos para ejecutar una clasificación supervisada es la definición de las clases o en este caso coberturas que se desean identificar; esta tarea requiere un conocimiento previo del comportamiento de la zona de estudio por parte del analista y depende directamente de la escala de identificabilidad que le permita el insumo, en este caso una imagen con una resolución espacial de diez (10) metros de Sentinel – 2. Para la definición de las coberturas a identificar se tomó como referencia la clasificación Corine Land Cover para Colombia escala 1:100.000 (IDEAM, 20210), y se decidió que por la escala de trabajo se llegaría al segundo nivel. Las coberturas seleccionadas para la clasificación son:

**Tabla 2** Coberturas a clasificar

Identificador	Código	Nivel 1	Nivel 2
1	1.1	Territorios artificializados	Zonas urbanizadas
2	5.1	Superficies de agua	Aguas continentales
3	3.1	Bosques y áreas semi naturales	Bosques
4	2.3	Territorios agrícolas	Pastos
5	3.3	Bosques y áreas semi naturales	Áreas abiertas, sin o con poca vegetación
6	3.2	Bosques y áreas semi naturales	Áreas con vegetación

Con las coberturas plenamente definidas, se realiza la creación de las áreas de entrenamiento, estas se generan en ArcGIS Pro en geometría polígono; en este paso se debe garantizar que las muestras se encuentran distribuidas en la totalidad de la escena. En la figura 5 se muestra la distribución de las áreas de entrenamiento.



**Figura 5** Área de entrenamiento

Fuente: Elaboración propia. RStudio.

### 8.3 Clasificación supervisada con el algoritmo Random Forest

Para la clasificación supervisada con el algoritmo de Random Forest, se hizo la adaptación necesaria de los códigos desarrollados por Lorena Orellana (Orellana, s.f.) y João Gonçalves (Gonçalves, s.f.). La ejecución del algoritmo se realizó con un total de 100 árboles y 8 repeticiones, cada de una de estas repeticiones es evaluada por los siguientes estadísticos:

**Tabla 3** Comparación de estadísticas de las 8 repeticiones en RF

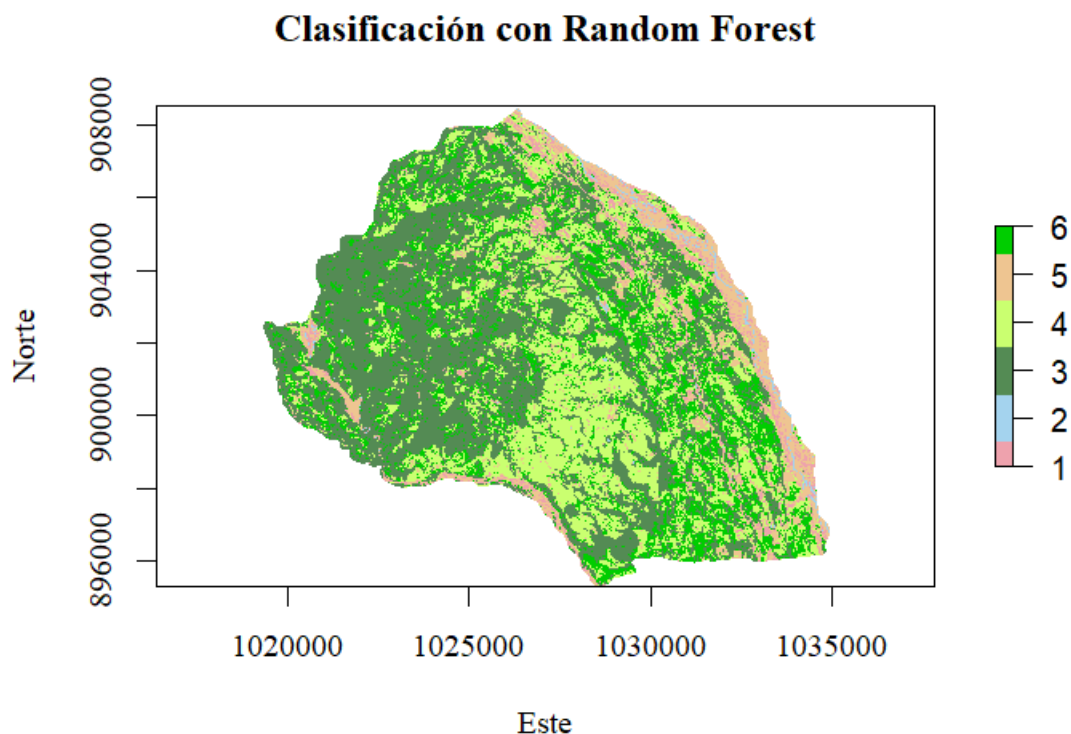
	Precision	Kappa	PSS
R_1	0.863	0.936	0.933
R_2	0.892	0.930	0.925
R_3	0.864	0.937	0.938
R_4	0.862	0.939	0.939
R_5	0.875	0.935	0.933
R_6	0.966	0.945	0.944
R_7	0.657	0.932	0.932
R_8	0.893	0.926	0.923

Con la mejor ronda, que en este caso es R\_6, se genera la predicción de la imagen satelital y se genera la matriz de confusión, teniendo el siguiente resultado:

**Tabla 4** Matriz de confusión RF

	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6
Clase 1	57	0	1	6	21	0
Clase 2	0	59	0	0	0	0
Clase 3	0	0	835	2	0	14
Clase 4	0	0	0	244	0	5
Clase 5	2	0	0	1	293	0
Clase 6	0	0	13	5	0	256

Una vez ejecutado en su totalidad el código (Ver Anexo 1), se tiene la siguiente imagen clasificada con índice kappa de 0.9454 y precisión de 0.9661, con un tiempo de entrenamiento del algoritmo de 1'6''16 y un tiempo de clasificación de 24''55.



**Figura 6** Resultado de clasificación con RF

Fuente: Elaboración propia. RStudio.

## 8.4 Clasificación supervisada con el algoritmo K-Nearest Neighbors

Para la clasificación supervisada con el algoritmo K-Nearest Neighbors, se hizo la adaptación necesaria del código desarrollado por Aliz (Aliz, s.f.). La ejecución del algoritmo se realizó con 3,5,7,9,11 vecinos y valores de distancia de 1 y 2, cada una de estas combinaciones es evaluada por los siguientes estadísticos:

**Tabla 5** Comparación de estadísticas de las combinaciones en K-NN

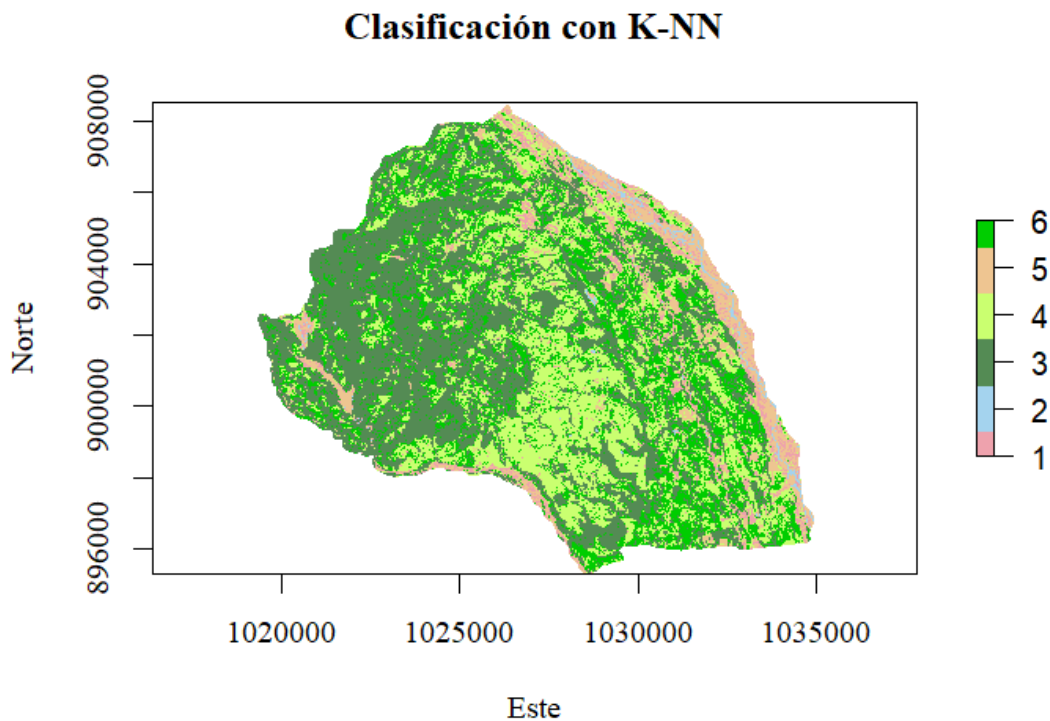
kmax	distance	Accuracy	Kappa
3	1	0.9412701	0.9176391
3	2	0.9456336	0.9237194
5	1	0.9503209	0.9301287
5	2	0.9534939	0.9346025
7	1	0.9503209	0.9301400
7	2	0.9542813	0.9357420
9	1	0.9495304	0.9290266
9	2	0.9542813	0.9357420
11	1	0.9495304	0.9290266
11	2	0.9519004	0.9324215

Con la construcción del anterior modelo, se genera la predicción de la imagen satelital y se genera la matriz de confusión, teniendo el siguiente resultado:

**Tabla 6** Matriz de confusión K-NN

	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6
Clase 1	35	0	0	0	4	0
Clase 2	0	38	0	0	0	0
Clase 3	0	0	484	0	0	8
Clase 4	0	0	1	144	0	2
Clase 5	15	0	0	0	182	0
Clase 6	0	0	18	2	0	158

Una vez ejecutado en su totalidad el código (Ver Anexo 2), se tiene la siguiente imagen clasificada con índice kappa de 0.9359 y precisión de 0.9542, con un tiempo de entrenamiento del algoritmo de 34''39 y un tiempo de clasificación de 45''30.



**Figura 7** Resultado de clasificación con K-NN

Fuente: Elaboración propia. RStudio.

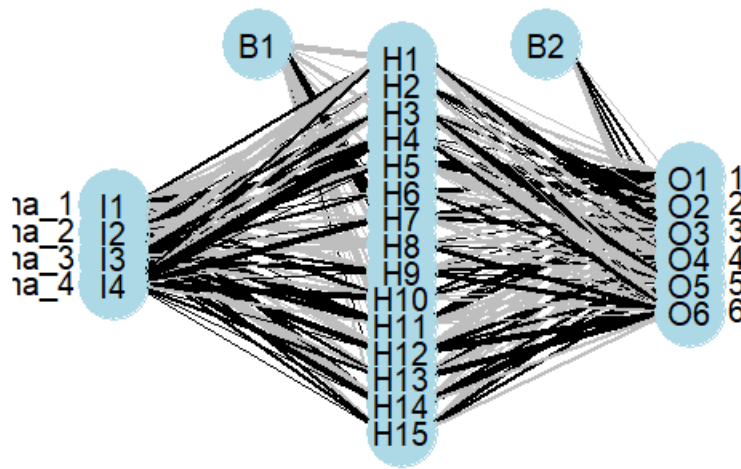
## 8.5 Clasificación supervisada con el algoritmo Red Neuronal Artificial

Para la clasificación supervisada con el algoritmo Red Neuronal Artificial, se hizo la adaptación necesaria del código desarrollado por Aliz (Aliz, s.f.). La ejecución del algoritmo se realizó con los tamaños 5, 10, 15 y valores de decadencia 0.001, 0.01, 0.1, cada una de estas combinaciones es evaluada por los siguientes estadísticos:

**Tabla 7** Comparación de estadísticas de las combinaciones en RNA

size	decay	Accuracy	Kappa
5	0.001	0.9515254	0.9317051
5	0.010	0.9499382	0.9294297
5	0.100	0.9495475	0.9287566
10	0.001	0.9570560	0.9394744
10	0.010	0.9605668	0.9443908
10	0.100	0.9586138	0.9416426
15	0.001	0.9645148	0.9499905
15	0.010	0.9649038	0.9504848
15	0.100	0.9590075	0.9422088

Con estas combinaciones se crea un modelo final de red neuronal, donde se crean las relaciones entre las cuatro bandas que componen la imagen satelital y las seis clases definidas, generando el siguiente esquema:



**Figura 8** Esquema de la red neuronal modelada

Fuente: Elaboración propia. RStudio.

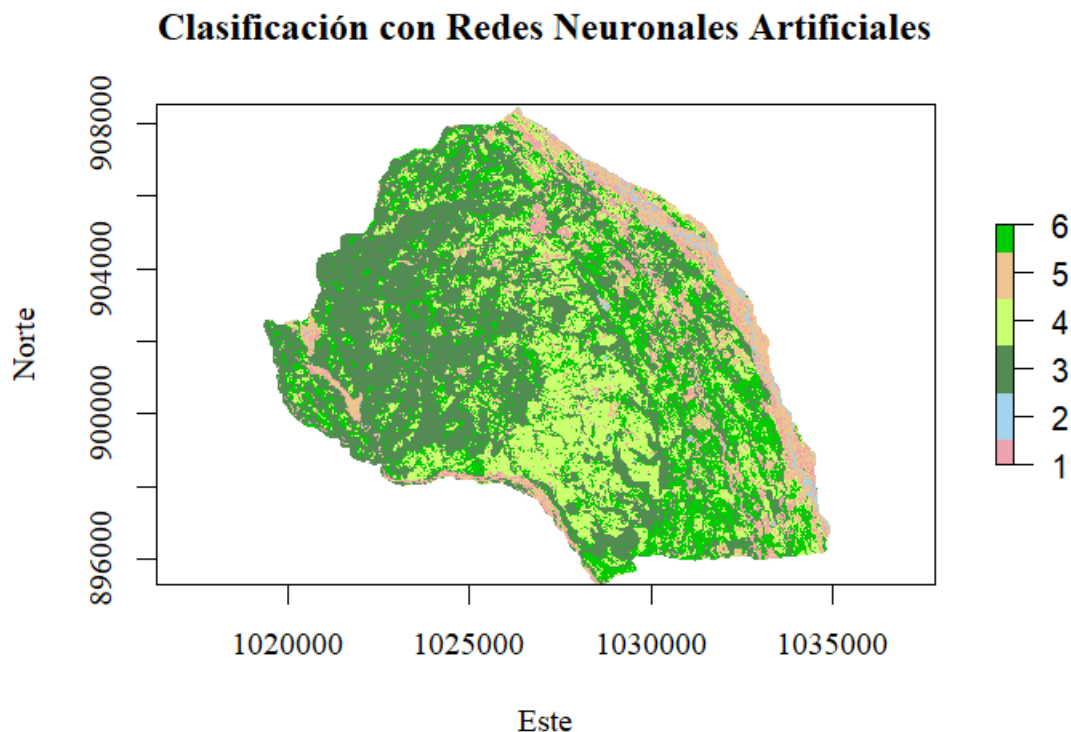
Con la construcción del anterior modelo, se genera la predicción de la imagen satelital y se genera la matriz de confusión, teniendo el siguiente resultado:

**Tabla 8** Matriz de confusión RNA

	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6
Clase 1	43	0	0	0	3	0
Clase 2	0	38	0	0	0	0
Clase 3	0	0	497	0	0	8
Clase 4	0	0	0	145	1	3
Clase 5	7	0	0	0	182	0
Clase 6	0	0	6	1	0	157

Una vez ejecutado en su totalidad el código (Ver Anexo 3), se tiene la siguiente imagen clasificada con índice kappa de 0.9627 y precisión de 0.9734, con un tiempo de entrenamiento del algoritmo de 46''90 y un tiempo de clasificación de 05''91.





**Figura 9** Resultado de clasificación con RNA

Fuente: Elaboración propia. RStudio.

## 8.6 Clasificación supervisada con el algoritmo Máquina de Soporte Vectorial

Para la clasificación supervisada con el algoritmo Máquina de Soporte Vectorial, se hizo la adaptación necesaria del código desarrollado por Aliz (Aliz, s.f.). La ejecución del algoritmo se realizó con método radial que es evaluado por los siguientes estadísticos:

**Tabla 9** Comparación de estadísticas de las combinaciones en MSV

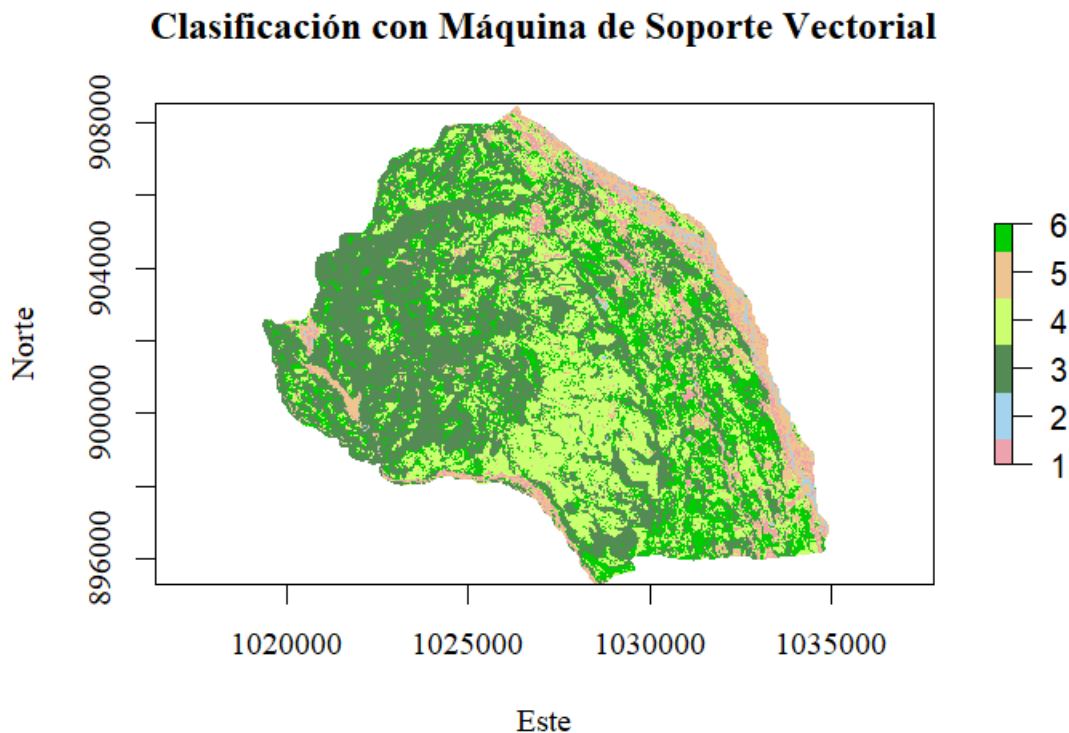
C	Accuracy	Kappa
0.25	0.9373310	0.9116433
0.50	0.9546742	0.9360810
1.00	0.9598017	0.9434119
2.00	0.9629560	0.9479005
4.00	0.9625607	0.9473753
8.00	0.9633466	0.9484254
16.00	0.9641371	0.9495298
32.00	0.9637418	0.9489943
64.00	0.9609875	0.9451113
128.00	0.9601970	0.9439593

Con la construcción del anterior modelo, se genera la predicción de la imagen satelital y se genera la matriz de confusión, teniendo el siguiente resultado:

**Tabla 10** Matriz de confusión MSV

	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6
Clase 1	42	0	0	0	5	0
Clase 2	0	38	0	0	0	0
Clase 3	0	0	495	0	0	8
Clase 4	0	0	0	146	0	4
Clase 5	8	0	0	0	181	0
Clase 6	0	0	8	0	0	156

Una vez ejecutado en su totalidad el código (Ver Anexo 4), se tiene la siguiente imagen clasificada con índice kappa de 0.9576 y precisión de 0.9698, con un tiempo de entrenamiento del algoritmo de 21''22 y un tiempo de clasificación de 58''48.



**Figura 10** Resultado de clasificación con MSV

Fuente: Elaboración propia. RStudio.

## 9. Análisis de resultados

En este capítulo se evalúa y compara los resultados generados con cada algoritmo, para determinar cuál es más adecuado para identificar la cobertura boscosa (clase 3 en la definición de coberturas) en el municipio. En la tabla 11 se muestra el consolidado de las estadísticas a considerar en la comparativa.

**Tabla 11** Estadísticas de los algoritmos

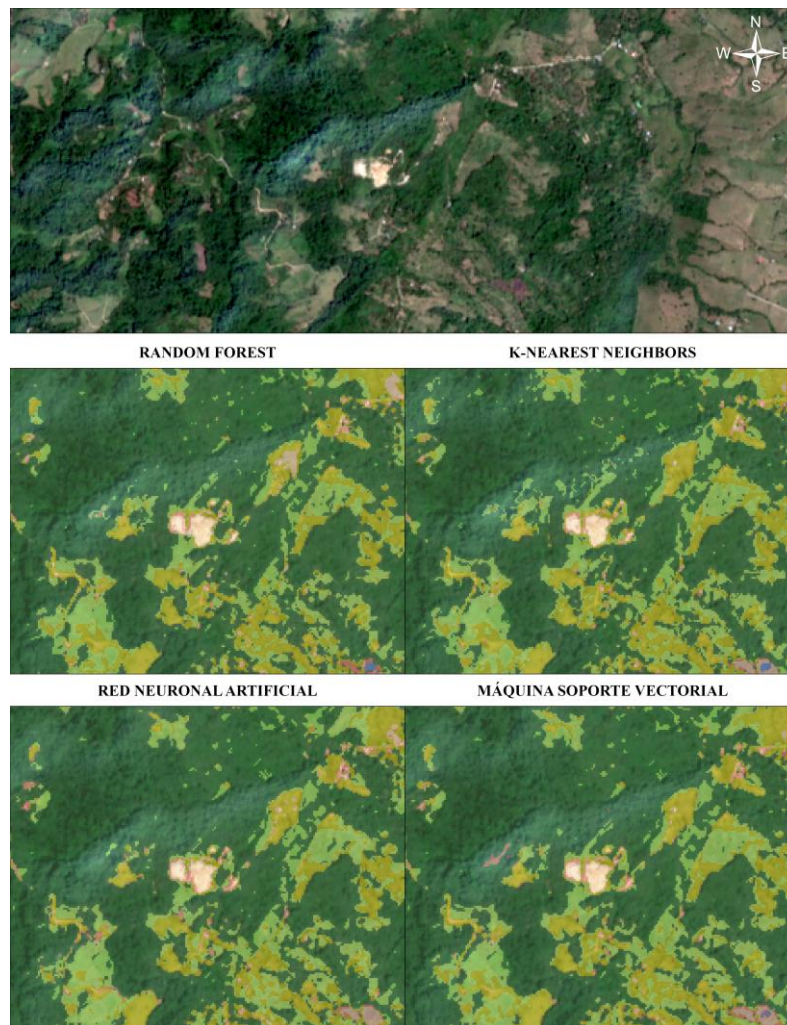
Algoritmo	Kappa	Precisión
RF	0.9454	0.9661
K-NN	0.9359	0.9542
RNA	0.9627	0.9734
MSV	0.9576	0.9698

Los valores registrados tanto en el índice kappa, como en la precisión indican que todos los algoritmos utilizados generan un resultado confiable estadísticamente en la clasificación supervisada de la totalidad de la imagen; siendo el de mejor bondad de ajuste de Red Neuronal Artificial. Sin embargo, al ser el interés puntual de este análisis la cobertura boscosa, se considera necesario evaluar el comportamiento de cada algoritmo en esta cobertura en particular haciendo uso de los resultados de la matriz de confusión donde las columnas son las categorías reales y las filas las categorías predichas, teniendo los resultados que se muestran en la tabla 12.

**Tabla 12** Comparación de resultados de la matriz de confusión para la clase 3

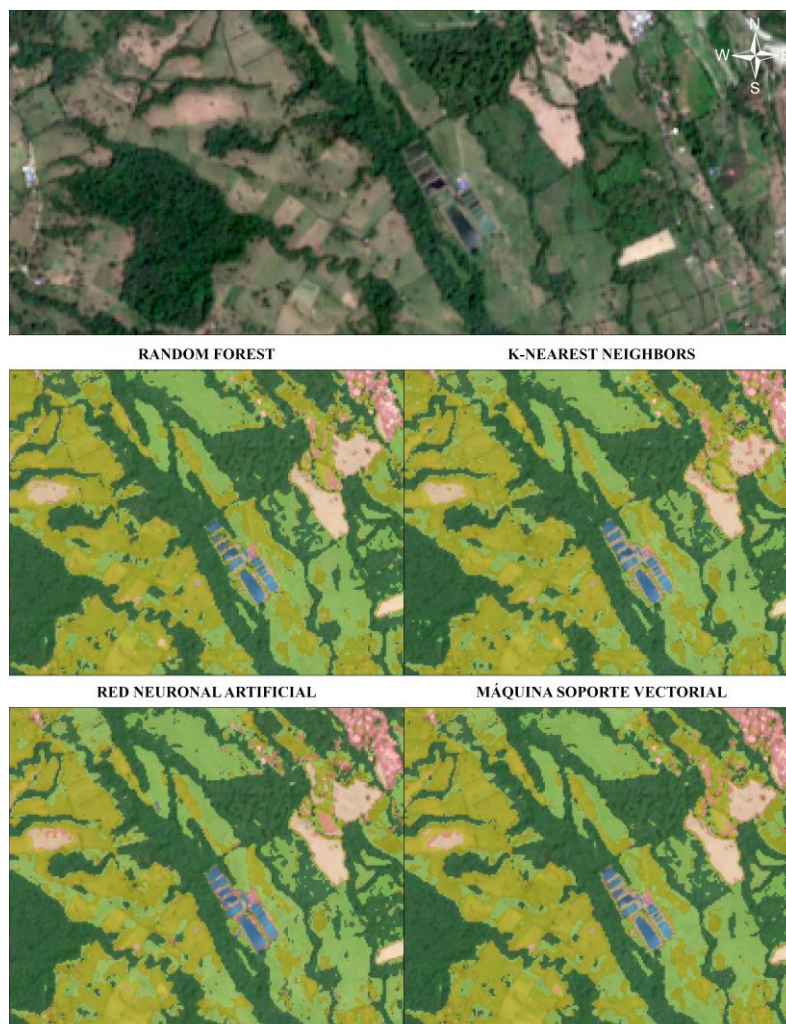
Clase 3: 3.1 Bosques				
Algoritmo	Total, de elementos	Clasificados correctamente	Clasificado incorrectamente	Porcentaje de acierto
RF	849	835	30	98,4%
K-NN	503	484	27	96,2%
RNA	503	497	14	98,8%
MSV	503	495	16	98,4%

En la matriz de confusión, se tomó el 50% de las muestras para el algoritmo RF y el 30% para los demás; esto por la naturaleza de cada uno. Los resultados de cada matriz de confusión ratifican que todos los algoritmos tienen un desempeño óptimo en la clasificación para la cobertura boscosa y que el algoritmo con mayor porcentaje de acierto con un 98.8% es el RNA. Finalmente, se realiza una última validación, en este caso visual y para la cual aplica el filtro mayoritario buscando el mejoramiento del producto de cada clasificación al eliminar llamado efecto sal y pimienta. A continuación, se muestran algunas de las zonas comparadas visualmente.



**Figura 11** Comparación visual zona de bosque No. 1

Fuente: Elaboración propia. ArcGIS Pro.



**Figura 12** Comparación visual zona de bosque No. 2

Fuente: Elaboración propia. ArcGIS Pro.

Visualmente, se tienen resultados similares y con la calidad requerida con los cuatro algoritmos; sin embargo, cuando se va al detalle de los claros en el bosque y las zonas de bosque más angostas que se perciben de manera casi lineal son los algoritmos Random Forest y Red Neuronal Artificial los que mejores resultados muestran. Esta última validación termina de confirmar lo encontrado desde el componente estadístico de la validación, cuando se realizan comparaciones en procedimientos que tienen un gasto computacional, este rendimiento también debe ser analizado; a nivel de memoria no se evidencia ningún dato relevante a mencionar caso

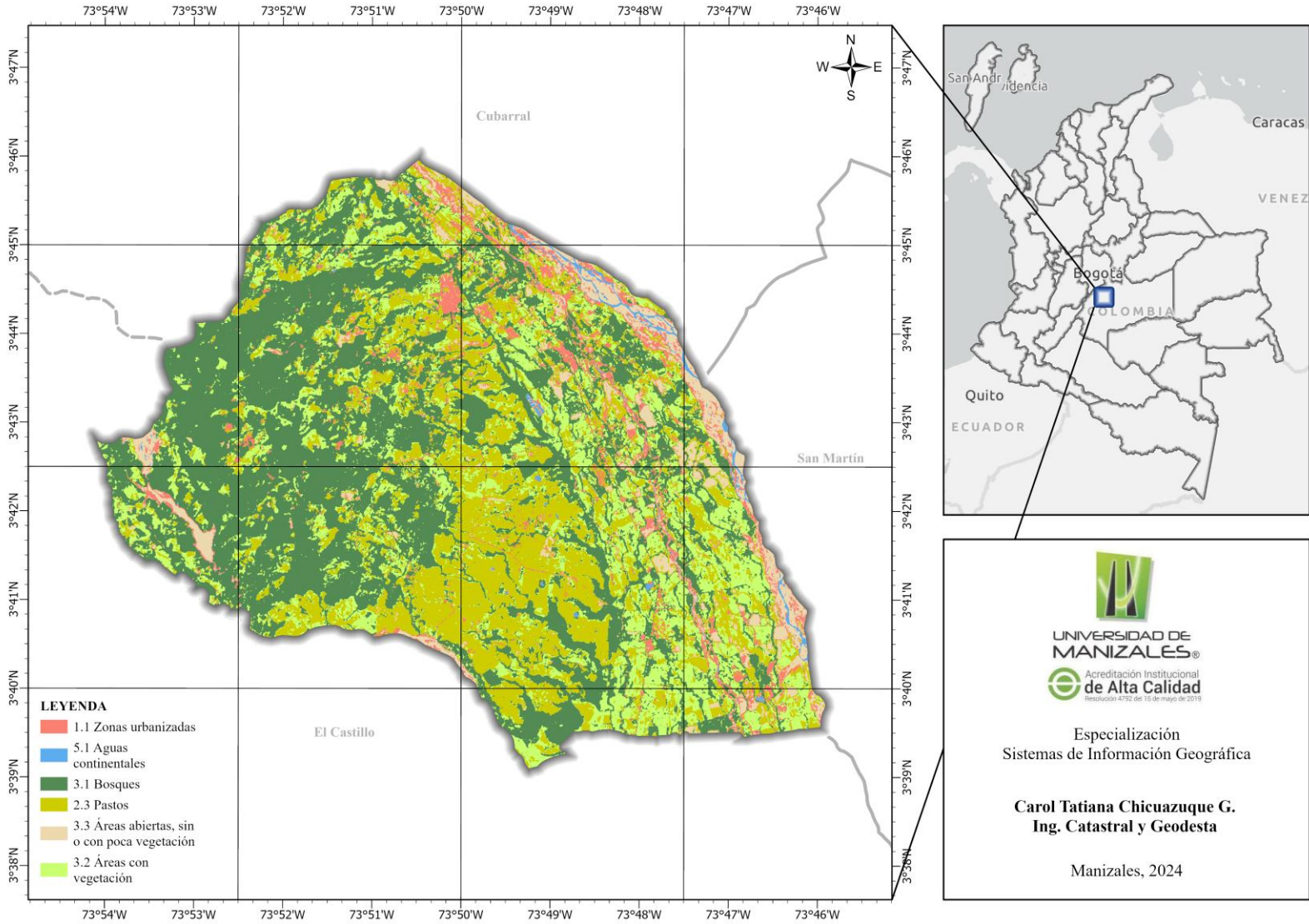
contrario a los tiempos de ejecución donde se presenta una variación con cada algoritmo, en la tabla 13 se muestran los tiempos registrados.

**Tabla 13** Comparación de tiempos de ejecución

Algoritmo	Tiempo de entrenamiento	Tiempo de clasificación	Tiempo total
<b>RF</b>	1' 6.16"	0' 24.55"	1' 30.71"
<b>K-NN</b>	0' 34.39"	0' 45.30"	1' 19.69"
<b>RNA</b>	0' 46.90"	0' 5.91"	0' 52.81"
<b>MSV</b>	0' 21.22"	0' 58.48"	1' 19.70"

En lo referente a tiempos de ejecución se pudo establecer que los algoritmos K-NN y MSV registran tiempos muy similares, casi exactos, que el algoritmo RF es el de mayor tiempo de ejecución y que el RNA es el de menor.

En total se realizaron cuatro validaciones que incluyeron análisis estadísticos, visuales y de rendimiento computacional y aunque en general se puede afirmar que los cuatro algoritmos evaluados generan resultados óptimos y con la calidad requerida, es el algoritmo de Red Neuronal Artificial el que más se destaca en sus resultados. Concluyendo que el algoritmo de clasificación supervisada más adecuado para la identificación de la cobertura boscosa en el municipio El Dorado (Meta) es el de Red Neuronal artificial. Resultado que además tiene concordancia con lo encontrado por otros autores como (Ávila-Pérez et al., 2020) y (Chucos & Vega, 2022) donde el algoritmo RNA fue destacado en sus resultados como uno de los mejores algoritmos en sus respectivos ejercicios. Finalmente, se presenta el mapa de clasificación del municipio con RNA.



UNIVERSIDAD DE MANIZALES®  
Acreditación Institucional de Alta Calidad  
Resolución 4752 del 18 de mayo de 2013

Especialización  
Sistemas de Información Geográfica

**Carol Tatiana Chicuazuque G.**  
Ing. Catastral y Geodesta

Manizales, 2024

RESULTADO CLASIFICACIÓN SUPERVISADA - RNA

## 10. Conclusiones

Analizados los índices de precisión y kappa, la matriz de confusión y los resultados a nivel visual de los cuatro algoritmos de clasificación supervisada es posible afirmar que el mejor algoritmo para la identificación de la cobertura boscosa en el municipio El Dorado es la Red Neuronal Artificial, presentando una precisión de 0.97, un índice kappa de 0.96 y un porcentaje de acierto de acuerdo a la matriz de confusión del 98.8%. A nivel visual es posible evidenciar su bondad en la identificación de los claros de bosque, que le dan una mayor precisión a la identificación de los elementos pertenecientes a esta cobertura.

En general, los resultados obtenidos con los cuatro algoritmos son óptimos y cumplen con las condiciones de calidad requeridas para una buena clasificación de coberturas; esto comprueba que en este tipo de ejercicios de machine learning uno de los factores con mayor incidencia en el resultado es el entrenamiento realizado por el analista, siendo este archivo la base para el procesamiento y en este caso insumo común en la ejecución de los cuatro algoritmos.

Visualmente, el algoritmo de Red Neuronal Artificial tiene un buen desempeño como se mencionó con anterioridad; sin embargo, se evidencia que en aquellas zonas boscosas de menor dimensión donde por la escala de la imagen satelital se tiene una forma más lineal que de área es el algoritmo Random Forest el que mejores resultados muestra. Con este ejercicio de observación también se analizó el comportamiento de las otras coberturas identificadas de acuerdo a los algoritmos y se puede afirmar lo siguiente: para la cobertura de aguas continentales el algoritmo



que mejor resultado visual tienen es el de Red Neuronal Artificial, las zonas urbanizadas aunque se pueden identificar de manera clara en la clasificación presentan una respuesta espectral similar a las áreas abiertas o con poca vegetación, generando que la clase correspondiente a las zonas urbanizadas no tenga una precisión alta respecto a la realidad del territorio en ninguno de los algoritmos, siendo el que mejor se comporta el K-Nearest Neighbors.

En la ejecución de ejercicios de caracterización del territorio por medio de imágenes satelitales, como es el caso del presente, es de gran importancia tener clara la escala de detalle que se tiene de acuerdo con los insumos. En este caso particular, se evidenció la importancia de la escala en la definición de las coberturas a clasificar que tuvo como base la metodología Corine Land Cover para Colombia, siendo el nivel 2 de clasificación el apropiado pues al intentar realizar la clasificación con el nivel 3 que implica una identificación mucho más puntual de cada cobertura el resultado no era óptimo ni correspondiente a la realidad del territorio.

Finalmente, se considera relevante destacar la importancia de la capacidad computacional en los ejercicios de machine learning, pues en muchas ocasiones los algoritmos utilizados o los insumos si se cuenta con imágenes de alta resolución particularmente aquellos capturados con vehículo aéreo no tripulado, requieren un nivel de memoria y disco más allá del proporcionado por los computadores de gama media – alta. En este caso, el requerimiento computacional no fue más allá del esperado y en general se puede afirmar que los cuatro algoritmos presentaron un rendimiento óptimo donde ninguno de ellos superó los 2 minutos en ejecución.

Estos tiempos registrados responden directamente a la imagen utilizada que cuenta con una resolución espacial media, que puede ser utilizada en generación de cartografía temática a nivel rural y la cantidad de clases definidas. Al intentar el procesamiento con una imagen de mejor resolución espacial los tiempos van a aumentar significativamente o en algunos casos van a requerir incluso de una mejora en las características de la máquina para poder llevar a cabo del procesamiento requerido.

A nivel de recomendaciones para futuras aplicaciones de machine learning en imágenes satelitales o aéreas, se tienen tres ítems en los cuales se debe prestar una particular atención; el primero de ellos la calidad el insumo y su correcta georreferenciación, siendo la segunda característica de gran importancia en este tipo de procesamientos pues tener el sistema de referencia adecuado garantiza que los resultados obtenidos cuenten con exactitud la posicional requerida a nivel de calidad. El segundo ítem a tener presente es la escala en la cual se debe trabajar de acuerdo al insumo y así mismo el nivel de detalle con el cual se va a realizar el entrenamiento, pues un nivel de detalle no correspondiente a la escala va a generar una confusión de píxeles que se traduce en una clasificación errónea y finalmente, el tercer ítem y quizá el más relevante es el juicioso entrenamiento del algoritmo a aplicar, pues de un buen entrenamiento depende el éxito de la clasificación y por ende la calidad del producto generado.

## 11. Bibliografía

- Abburu, S., & Golla, S. (2015). Satellite Image Classification Methods and Techniques: A Review. *International Journal of Computer Applications*, 119(8).
- Al-Ahmadi, F., & Hames, A. (2009). Comparison of Four Classification Methods to Extract Land Use and Land Cover from Raw Satellite Images for Some Remote Arid Areas, Kingdom of Saudi Arabia. *JKAU; Earth Sci.*, 1, 167-191.
- Aliz, E. (s.f.). *GitHub*. Obtenido de Clasificacion-imagenes-satelitales-TFG: <https://github.com/elianaaliz/SatelliteImageClassification-TFG>
- Alshari, E., & Gawali, B. (2021). Development of classification system for LULC using remote sensing and GIS. *Global Transitions Proceedings*, 2(1), 8-17.  
doi:<https://doi.org/10.1016/j.gltp.2021.01.002>
- Ávila-Pérez, I., Ortiz-Malavasi, E., Soto-Montoya, C., Vargas-Solano, Y., Aguilar-Arias, H., & Miller-Granados, C. (2020). Evaluación de cuatro algoritmos de clasificación de imágenes satelitales Landsat-8 y Sentinel-2 para la identificación de cobertura boscosa en paisajes altamente fragmentados en Costa Rica. *Revista de Teledetección*(57), 37-49.  
doi:<https://doi.org/10.4995/raet.2020.13340>
- Breiman, L., J.H, F., Olshen, R., & Stone, C. (1984). Classification and Regression Trees. *Chapman and Hall/CRC*.
- Cánovas-García, F., Alonso-Sarría, F., & Gomariz-Castillo, F. (2016). Modificación del algoritmo Random Forest para su empleo en clasificación de imágenes de teledetección. *Aplicaciones de las Tecnologías de la Informaración Geográfica (TIG) para el Desarrollo Económico Sostenible*, 359-368.
- Chucos, N., & Vega, E. (2022). Evaluación de algoritmos de machine learning en la clasificación de imágenes satelitales multiespectrales, caso: Amazonia Peruana. *Ciencia Latina. Revista Multidisciplinaria*, 6(1), 4946-4963.  
doi:[https://doi.org/10.37811/cl\\_rcm.v6i1.1843](https://doi.org/10.37811/cl_rcm.v6i1.1843)
- Chugh, R., Bhatia, V., Khanna, K., & Bhatia, V. (2020). A Comparative Analysis of Classifiers for Image Classification. *IEEE*. doi:10.1109/Confluence47617.2020.9058042
- Chuvieco, E. (1995). Fundamentos de teledetección espacial (Segunda ed.).
- Corts, R., Navarro, M., Pardo, J., & Bisbal, M. (2010). Clasificación Periódica de Coberturas Terrestres a Escala Regional con Imágenes Modis. *Revista Internacional de Ciencia y*, 10, 1-17.
- Del Toro, N., Gomariz-Castillo, F., Cánovas-García, F., & Alonso-Sarría, F. (2015). Comparación de métodos de clasificación de imágenes de satélite en la Cuenca del Río Argos (Región de Murcia). *Boletín de la Asociación de Geógrafos Españoles*(67), 327-347. doi:I.S.S.N.: 0212-9426

- ESA. (17 de Junio de 2015). *The European Space Agency*. Obtenido de Sentinel 2: [https://www.esa.int/Space\\_in\\_Member\\_States/Spain/SENTINEL\\_2](https://www.esa.int/Space_in_Member_States/Spain/SENTINEL_2)
- Espínola, A. (2014). *Clasificación de imágenes de satélite mediante autómatas celulares*. Almería, España: Universidad de Almería.
- Esri. (s.f.). *Introducción a ArcGIS*.
- Godoy, F. (2021). *Métodos clásicos de clasificación: comparación y aplicación*. Universidad Nacional de Córdoba.
- Gonçalves, J. (s.f.). *GitHub*. Obtenido de <https://raw.githubusercontent.com/joaofgoncalves/Evaluation/master/eval.R>
- Hwang, J., & Chiang, H. (2010). The study of high resolution satellite image classification based on support vector machine. *IEEE*, 1-6.
- IBM. (s.f.). *¿Qué es KNN?* Obtenido de <https://www.ibm.com/mx-es/topics/knn>
- IDEAM. (2021). *Leyenda Nacional de Coberturas de la Tierra. Metodología CORINE Land Cover adaptada para Colombia Escala 1:100.000*. Bogotá D.C.: Instituto de Hidrología, Meteorología y Estudios Ambientales.
- Jaramillo, L., & Antunes, A. (2018). Detección de cambios en la cobertura vegetal mediante interpretación de imágenes Landsat por redes neuronales artificiales (RNA). Caso de estudio: Región Amazónica Ecuatoriana. *Revista de Teledetección*, 33. doi:10.4995/raet.2018.8995
- Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, 18-22.
- Lillesand, T., Kiefer, R., & Chipman, J. (2015). *Remote sensing and image interpretation. Séptima ed.*
- Montaño, J. (2002). Redes Neuronales Artificiales aplicadas al Análisis de Datos. *Scielo*, 1(1), 315.
- Murillo, R. (2021). Implementación del método de máquinas de soporte vectorial en bases de datos espaciales para análisis de clasificación supervisada en imágenes de sensores remotos. *Revista Cartográfica 102*. doi:<https://doi.org/10.35424/rcarto.i102.830>
- Murillo, R. (2021). Implementación del método máquinas de soporte vectorial en bases de datos espaciales para análisis de clasificación supervisada en imágenes de sensores remotos. *Revista Cartográfica*. doi:ISSN 2663-3981
- Murillo, R. A. (2020). Implementación del método de máquinas de soporte vectorial en bases de datos espaciales para análisis de clasificación supervisada en imágenes de sensores remotos. *Revista Cartográfica 102*. doi:<https://doi.org/10.35424/rcarto.i102.830>
- Nhamo, L., Van Dijk, R., Magidi, J., Wiberg, D., & Tshikolomo, K. (2018). Improving the accuracy of remotely sensed irrigated areas using post-classification enhancement

- through UAV capability. *Remote Sensing*, 10(5), 712-723.  
doi:<https://doi.org/10.3390/rs10050712>
- Ningthoujam, R., Tansey, K., Balzter, H., Morrison, K., Johnson, S., Gerard, F., . . . Blythe, T. (2016). Mapping forest cover and forest cover change with airborne s-band radar. *Remote Sensing*, 8(7), 577-597. doi:<https://doi.org/10.3390/rs8070577>
- Nolasco, M., Bocco, M., & Scavuzzo, C. (2017). Mapeo de cultivos agrícolas en Córdoba: una comparación del desempeño de tres algoritmos de clasificación. *Congreso Argentino de AgroInformática*, 58-68. doi:ISSN: 2525-0949
- Orellana, L. (s.f.). *Clasificación supervisada*. Obtenido de RPubs by RStudio:  
[https://rpubs.com/marialorena/clasificacion\\_supervisada](https://rpubs.com/marialorena/clasificacion_supervisada)
- Riaño, O., Acosta, C., & Leal, R. (2016). Clasificación de una imagen satelital empleando máquinas de soporte vectorial para cuantificar el área de Pinus Patula en una plantación. *Redes de Ingeniería*. doi:ISSN: 2248 - 762X
- Rojas, M. (2009). *Caracterización de imágenes hiperespectrales utilizando support vector machines y técnicas de extracción de características*. España: Universidad de Extremadura.
- Sá, I., Da Antonio, R., Almorox, J., & De Souza, A. (1996). Comparación de dos métodos de clasificación digital y aplicación al cartografiado de vegetación en el nordeste brasileño. *Revista de Teledetección*(7), 27-33.
- Sánchez, J. (2016). Análisis de Calidad Cartográfica mediante el uso de la Matriz de Confusión. *G.E.I. Pensamiento Matemático MAIC*, VI(2), 09-26. doi:ISSN 2174-0410
- Serra, A. (2020). *Comparación de algoritmos de clasificación supervisada*. Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: from theory to algorithms. *Cambridge University Press*.
- Shen, H., Lin, Y., Tian, Q., Xu, K., & Jiao, J. (2018). A comparison of multiple classifier combinations using different voting-weights for remote sensing image classification. *International Journal of Remote Sensing*, 39(11), 3705-3722.  
doi:<https://doi.org/10.1080/01431161.2018.1446566>
- Suárez, A., Jiménez, A., Castro-Franco, M., & Cruz-Roa, A. (2017). Clasificación y mapeo automático de coberturas del suelo en imágenes satelitales utilizando Redes Neuronales Convolucionales. *Colombia Suplemento*, 21(1), 64-75.
- UCH. (06 de Abril de 2018). *Universidad de Ciencias y Humanidades*. Obtenido de ¿Qué son las imágenes satelitales?: <https://blog.uch.edu.pe/ingenieria-electronica-con-mencion-en-telecomunicaciones/que-son-las-imagenes-satelitales/>
- Universidad de Zaragoza. (s.f.). *R*.

- Utiltech, S. (2018). Teledetección e inteligencia artificial para incrementar los beneficios de tu explotación agrícola siendo más respetuoso con el medio ambiente. *Hispaaviacion*.
- Wardlow, B., Egbert, S., & Kastens, J. (2007). Analysis of time-series MODIS 250 m vegetation index data for crop classification in the U.S. Central Great Plains. *Remote*, 108, 290-310.
- Willington, E., Nolasco, M., & Bocco, M. (2013). Clasificación supervisada de suelos de uso agrícola en la zona central de Córdoba (Argentina): comparación de distintos algoritmos sobre imágenes Landsat . *Congreso Argentino de AgroInformática, CAI, 42 JAIIO*, 207-216. doi:ISSN 1852-4850
- Yang, H., Pan, B., Wu, W., & Tai, J. (2018). Field-based rice classification in Wuhua county through integration of multi-temporal sentinel-1A and landsat-8 OLI data. *International Journal of Applied Earth Observation and Geoinformation*, 69, 226-236. doi:<https://doi.org/10.1016/j.jag.2018.02.019>
- Zamorano, J. (2018). *Comparativa y análisis de algoritmos de aprendizaje automático para la predicción del tipo predominante de cubierta arbórea*. Madrid, España: Universidad Complutense de Madrid.

## Anexos

### Anexo 1: Código en R del algoritmo Random Forest

#### *Código de cargue y preparación del insumo*

##### *#Cargue de las librerías*

```
library(sp)
library(raster)
library(terra)
library(sf)
library(ggplot2)
library(reshape2)
library(caret)
library(kableExtra)
library(randomForest)
```

##### *#Cargue de la imagen*

```
Escena = brick("C:/Especializacion_TrabajoGrado/Magna/EIDorado_Magna.tif")
plot(Escena)
plotRGB(Escena, r=3,g=2,b=1, stretch="lin")
```

##### *#Proceso de clasificación*

```
#Cargue de archivo shapefile con las áreas de entrenamiento
AE = shapefile("C:/Especializacion_TrabajoGrado/Magna/AreasEntrenamiento.shp")
plot(AE)
```

##### *#Cargue shapefile del municipio*

```
Mpio = shapefile("C:/Especializacion_TrabajoGrado/Magna/Municipio.shp")
```

##### *#Conversión de las áreas de entrenamiento a raster*

```
AOI = rasterize(AE, Escena, field='Id')
```

##### *#Grafica de las Areas de entrenamiento*

```
colores = c("#EEA2AD", "#A4D3EE", "#548B54", "#CAFF70", "#EEC591", "#00CD00")
leyenda = c("1.1", "5.1", "3.1", "2.3", "3.3", "3.2")
plot(AOI, col=colores, xlab="Este", ylab="Norte", cex.lab=1, family="serif")
plot(Mpio, add=TRUE)
legend("topright", legend = leyenda, col = colores, pch = 12, cex = 1)
```

##### *# El número de muestras de entrenamiento por categoría*

```
tab = table(values(AOI))
print(tab)
```

##### *#Remoción de 0 de fondo*

```
AOI[AOI==0] = NA
```

```

#-----
#RANDOM FOREST

#Conversión a discreto
Entrenamiento = ratify(AOI)
# Eliminación de NAs
Entrenamiento.df = na.omit(values(stack(Entrenamiento, Escena)))
Entrenamiento.df[, "layer"] = as.factor(as.character(Entrenamiento.df[, "layer"]))

#Establecimiento de parámetros
NRepeticiones = 8
PEntrenamiento = 0.5

# Creación de matriz que permite almacenar información del procedimiento y validación de la clasificación
n = nrow(Entrenamiento.df)
nClases = length(unique(Entrenamiento.df[, "layer"]))

confMats = array(NA, dim= c(nClases,nClases,NRepeticiones))

EvalMatriz = matrix(NA, nrow = NRepeticiones, ncol = 3,
                    dimnames = list(paste("R_", 1:NRepeticiones, sep=""),
                                    c("Precision", "Kappa", "PSS")))

pb = txtProgressBar(1, NRepeticiones, style = 3)

#Calibración y evaluación del clasificador
#Función de evaluación
Evaluate = function(actual=NULL, predicted=NULL, cm=NULL){

  if(is.null(cm)) {
    naVals = union(which(is.na(actual)), which(is.na(predicted)))
    if(length(naVals) > 0) {
      actual = actual[-naVals]
      predicted = predicted[-naVals]
    }
    f = factor(union(unique(actual), unique(predicted)))
    actual = factor(actual, levels = levels(f))
    predicted = factor(predicted, levels = levels(f))
    cm = as.matrix(table(Actual=actual, Predicted=predicted))
  }

  n = sum(cm) # numero de instancias
  nc = nrow(cm) # numero de clases
  diag = diag(cm) # número de instancias correctamente clasificadas por clase
  rowsums = apply(cm, 1, sum) # numero de instancias por calses
  colsums = apply(cm, 2, sum) # numero de predicciones por clases
  p = rowsums / n # distribución de instancias sobre las clases
  q = colsums / n # distribución de instancias sobre las clases predecidas

```



```

# Exactitud
accuracy = sum(diag) / n
recall = diag / rowsums
precision = diag / colsums
f1 = 2 * precision * recall / (precision + recall)

# Macro prf
macroPrecision = mean(precision)
macroRecall = mean(recall)
macroF1 = mean(f1)

# 1-vs-toda la matriz
oneVsAll = lapply(1 : nc,
  function(i){
    v = c(cm[i,i],
      rowsums[i] - cm[i,i],
      colsums[i] - cm[i,i],
      n-rowsums[i] - colsums[i] + cm[i,i]);
    return(matrix(v, nrow = 2, byrow = T))})

s = matrix(0, nrow=2, ncol=2)

for(i in 1:nc){
  s = s + oneVsAll[[i]]
}

# Promedio exactitud
avgAccuracy = sum(diag(s))/sum(s)

# Micro prf
microPrf = (diag(s) / apply(s,1, sum))[1];

# Clase mayoritaria
mcIndex = which(rowsums==max(rowsums))[1] # majority-class index
mcAccuracy = as.numeric(p[mcIndex])

mcRecall = 0*p
mcRecall[mcIndex] = 1

mcPrecision = 0*p
mcPrecision[mcIndex] = p[mcIndex]

mcF1 = 0*p
mcF1[mcIndex] = 2 * mcPrecision[mcIndex] / (mcPrecision[mcIndex] + 1)

# Random /exactitud esperada
expAccuracy = sum(p*q)

# Kappa
kappa = (accuracy - expAccuracy) / (1 - expAccuracy)

```

`# Puntaje de Peirce`

```
a = sum(colsums * rowsums) / n^2
b = sum(rowsums^2) / n^2
pss <- (accuracy - a) / (1 - b)
```

`# Random guess`

```
rgAccuracy = 1 / nc
rgPrecision = p
rgRecall = 0*p + 1 / nc
rgF1 = 2 * p / (nc * p + 1)
```

`# Random guess peso`

```
rwgAccuracy = sum(p^2)
rwgPrecision = p
rwgRecall = p
rwgF1 = p
```

```
classNames = names(diag)
if(is.null(classNames)) classNames = paste("C",(1:nc),sep="")
```

```
metrics = rbind(
  Accuracy = accuracy,
  Precision = precision,
  Recall = recall,
  F1 = f1,
  MacroAvgPrecision = macroPrecision,
  MacroAvgRecall = macroRecall,
  MacroAvgF1 = macroF1,
  AvgAccuracy = avgAccuracy,
  MicroAvgPrecision = microPrf,
  MicroAvgRecall = microPrf,
  MicroAvgF1 = microPrf,
  MajorityClassAccuracy = mcAccuracy,
  MajorityClassPrecision = mcPrecision,
  MajorityClassRecall = mcRecall,
  MajorityClassF1 = mcF1,
  Kappa = kappa,
  PSS = pss,
  RandomGuessAccuracy = rgAccuracy,
  RandomGuessPrecision = rgPrecision,
  RandomGuessRecall = rgRecall,
  RandomGuessF1 = rgF1,
  RandomWeightedGuessAccuracy = rwgAccuracy,
  RandomWeightedGuessPrecision = rwgPrecision,
  RandomWeightedGuessRecall = rwgRecall,
  RandomWeightedGuessF1 = rwgF1)
```

```
colnames(metrics) = classNames
```

```
return(list(ConfusionMatrix = cm, Metrics = metrics))
```

```

#Ejecución del clasificador, usando 100 árboles
for(i in 1:NRepeticiones){

  # Crear índice aleatorio para la selección de filas en cada ronda
  sampIdx <- sample(1:n, size = round(n*PEntrenamiento))

  # Calibrar el clasificador random forest
  rf <- randomForest(y = as.factor(Entrenamiento.df[sampIdx, "layer"]),
                    x = Entrenamiento.df[sampIdx, -1],
                    ntree = 100)

  # Predecir las clases con el st de prueba
  testSetPred <- predict(rf, newdata = Entrenamiento.df[-sampIdx,], type = "response")

  testSetObs <- Entrenamiento.df[-sampIdx,"layer"]

  # Evaluar
  evalData <- Evaluate(testSetObs, testSetPred)

  EvalMatriz[i,] <- c(evalData$Metrics["Precision",1],
                    evalData$Metrics["Kappa",1],
                    evalData$Metrics["PSS",1])

  #Matriz de confusión por ronda evaluada
  confMats[,i] <- evalData$ConfusionMatrix

  # Clasificar toda la imagen
  rstPredClassTMP <- predict(Escena, model = rf,
                            factors = levels(Entrenamiento.df["layer"]))

  if(i==1){
    # Raster predicho
    rstPredClass <- rstPredClassTMP

    # obtener la precision para cada clase
    Precision <- evalData$Metrics["Precision",,drop=FALSE]
    Recall <- evalData$Metrics["Recall",,drop=FALSE]

  }else{
    # Apilar los rasters predichos
    rstPredClass <- stack(rstPredClass, rstPredClassTMP)

    # Precisión y recuperación para cada clase
    Precision <- rbind(Precision,evalData$Metrics["Precision",,drop=FALSE])
    Recall <- rbind(Recall,evalData$Metrics["Recall",,drop=FALSE])
  }

  setTxtProgressBar(pb,i)
}

```

### #Tabla de medidas de precisión

```
TablaPrecision = knitr::kable(EvalMatriz, digits = 3)
```

### #Cálculo del promedio de las 20 rondas para cada una de las medidas y la desviación estándar

#### #Matriz de confusión con la mejor ronda

```
EvalMatriz[which.max(EvalMatriz[, "Kappa"]), , drop=FALSE]
```

#### # Mostrar la matriz de confusión con el mejor kappa

```
cm <- as.data.frame(confMats[,which.max(EvalMatriz[, "Kappa"])])
```

#### # Cambio nombres de la tabla

```
colnames(cm) <- rownames(cm) <- paste("N",levels(as.factor(Entrenamiento.df[, "layer"])),sep="_")
```

```
knitr::kable(cm)
```

#### #Cálculo de clase modelo

```
rstModalClass <- modal(rstPredClass)
```

```
writeRaster(rstModalClass, filename = "RandomForest_Marzo.tif")
```

```
rstModalClassFreq <- modal(rstPredClass, freq=TRUE)
```

```
medFreq <- zonal(rstModalClassFreq, AOI, fun=median)
```

#### #Mapas RandomForest

```
par(mfrow=c(1,2), cex.main=0.8, cex.axis=0.8)
```

```
cols <- c("#EEA2AD", "#A4D3EE", "#548B54", "#CAFF70", "#EEC591", "#00CD00")
```

```
plot(rstModalClass,col=cols,main = "Clasificación con Random Forest", legend=TRUE,xlab="Este",ylab="Norte",cex.lab=1, family="serif")
```

```
plot(rstModalClassFreq, main = "Frecuencia Modal")
```

## Anexo 2: Código en R del algoritmo K-Nearest Neighbors

*Código de cargue y preparación del insumo (Ver Anexo 1)*

```

#-----
#K-NEAREST NEIGHBOR
# Se divide las muestras en 70 y 30
trainx = list(0)
evalx = list(0)
for (i in 1:17){
# Se itera en todas las posibles categorías
  cls = Entrenamiento.df[Entrenamiento.df["layer"] == i,]
  smpl <- floor(0.70 * nrow(cls))
  tt <- sample(seq_len(nrow(cls)), size = smpl)
  trainx[[i]] <- cls[tt,]
  evalx[[i]] <- cls[-tt,]
}

# Se combina en dataframe referenciados, entrenamiento y evaluación
trn = do.call(rbind, trainx)
eva = do.call(rbind, evalx)

# Definiendo el control del entrenamiento
fitControl2 <- trainControl(method = "cv", # cross-validation of the training data
                             number = 10)

kkn.grid <- expand.grid(kmax = c(3, 5, 7, 9, 11), distance = c(1, 2),
                       kernel = "optimal")
# Entrenamiento con Knn
knn_model <- caret::train(x = trn[,2:ncol(trn)], y = as.factor(trn["layer"]),
                          method = "knn",
                          metric="Accuracy",
                          tuneGrid = kkn.grid,
                          preProcess = c("center", "scale"),
                          trControl = fitControl2)

#Información modelo KNN
print(knn_model)

#Evaluación del modelo
pred_knn_model <- predict(knn_model, newdata = eva)
cm <- confusionMatrix(data = pred_knn_model, as.factor(eva["layer"]))

#Predicción y generación del mapa
classi_knn_model <- raster::predict(Escena, model=knn_model)
cols <- c("#EEA2AD", "#A4D3EE", "#548B54", "#CAFF70", "#EEC591", "#00CD00")
plot(classi_knn_model, col=cols, main="Clasificación con K-NN", legend=
TRUE, xlab="Este", ylab="Norte", cex.lab=1, family="serif")
writeRaster(classi_knn_model, filename = "KNN_marzo.tif")

```

**Anexo 3: Código en R del algoritmo Red Neuronal Artificial***Código de cargue y preparación del insumo (Ver Anexo 1)*

```

#-----
#RED NEURONAL ARTIFICIAL
#Cargue de las librerías
library(NeuralNetTools)

# Definiendo el control del entrenamiento
fitControl3 <- trainControl(method = "cv", number = 10)

# Entrenamiento con Redes Neuronales
nnet_grid <- expand.grid(size = c(5, 10, 15),
                        decay = c(0.001, 0.01, 0.1))

nnet_model <- caret::train(x = trn[, (2:ncol(trn))], y = as.factor(trn[, "layer"]),
                          method = "nnet", metric = "Accuracy",
                          maxit = 100,
                          tuneGrid = nnet_grid,
                          preProcess = c("center", "scale"),
                          trControl = fitControl3,
                          trace = FALSE)

#Estructura de la red
plotnet(nnet_model$finalModel)

#Información modelo NNET
print(nnet_model)

#Evaluación del modelo
pred_nnet_model <- predict(nnet_model, newdata = eva)

cm <- confusionMatrix(data = pred_nnet_model, as.factor(eva[, "layer"]))

#Predicción del modelo y generación del mapa
classi_nnet_model <- raster::predict(Escena, model = nnet_model)
cols <- c("#EEA2AD", "#A4D3EE", "#548B54", "#CAFF70", "#EEC591", "#00CD00")
plot(classi_nnet_model, col = cols, main = "Clasificación con Redes Neuronales Artificiales", legend =
TRUE, xlab = "Este", ylab = "Norte", cex.lab = 1, family = "serif")
writeRaster(classi_nnet_model, filename = "RNN_marzo.tif")

```

## Anexo 4: Código en R del algoritmo Máquina de Soporte Vectorial

*Código de cargue y preparación del insumo (Ver Anexo 1)*

```
#-----  
#MAQUINA DE SOPORTE VECTORIAL  
# Definiendo el control del entrenamiento  
fitControl4 <- trainControl(method = "cv", # repeated cross-validation of the training data  
                           number = 10)  
  
# Entrenamiento con MSV  
svm_model<- caret::train(x = trn[, (2:ncol(trn))], y = as.factor(trn[, "layer"]),  
                        method = "svmRadial", metric="Accuracy",  
                        trControl = fitControl4,  
                        preProcess = c("center", "scale"),  
                        tuneLength = 10)  
  
#Información modelo MSV  
print(svm_model)  
  
#Evaluación del modelo  
pred_svm_model<- predict(svm_model,newdata = eva[,-1])  
  
cm <- confusionMatrix(data = pred_svm_model, as.factor(eva[, "layer"]))  
cm  
  
#Predicción y visualización  
classi_svm_model <- raster::predict(Escena, model=svm_model)  
plot(classi_svm_model, col=cols, main="Clasificación con Máquina de Soporte Vectorial", legend=  
TRUE,xlab="Este",ylab="Norte",cex.lab=1, family="serif")  
writeRaster(classi_svm_model, filename = "SVM_marzo.tif")
```