



DOBLE GRADO EN
MATEMÁTICAS Y ESTADÍSTICA

— TRABAJO FIN DE GRADO —

*MODELOS DE
PREDICCIÓN DE
INCENDIOS FORESTALES
EN ANDALUCÍA*

Juan Baeza Ruiz-Henestrosa

Sevilla, Mayo 2024

Índice general

Prólogo	V
Resumen	VII
Abstract	VIII
Índice de Figuras	X
Índice de Tablas	XI
1. Introducción	1
1.1. Objetivos	1
1.2. Hipótesis	1
1.3. Revisión bibliográfica	1
2. Preliminares	3
2.1. Datos georreferenciados	3
2.1.1. Datos Vectoriales	3
2.1.2. Datos Ráster	4
2.1.3. Sistemas de Referencia de Coordenadas	4
2.2. Análisis Exploratorio de Datos	6
2.2.1. Depuración de los datos	6
2.2.2. Análisis de Componentes Principales	7
2.3. Modelos	7
2.3.1. Regresión logística (con penalización)	7
2.3.2. Support Vector Machine	9
2.3.2.1. SVM lineal	9
2.3.2.2. SVM no lineal	10
2.3.3. Decision Trees	10
2.3.4. Random Forest	12
2.3.5. K-Nearest Neighbors	13
2.4. Validación del ajuste	13

2.5. Evaluación de los modelos	13
2.5.1. Clasificación binaria	13
2.6. Herramientas	14
3. Construcción del conjunto de datos	17
3.1. Determinación del marco del estudio	18
3.1.1. Incendios forestales	18
3.1.2. Variables predictoras	19
3.2. Fuentes de datos	20
3.3. Procesamiento de los datos	21
3.3.1. Generación de una muestra balanceada de casos positivos y negativos.	22
3.3.2. Asignación de las variables descriptivas a cada observación	24
3.3.3. Depuración de la muestra	25
4. Cuerpo	29
4.1. Análisis exploratorio de datos	29
4.1.1. Distribución de la variable objetivo	30
4.1.2. Análisis univariantes variables numéricas	32
4.1.3. Análisis multivariantes de las variables numéricas	37
4.1.4. Análisis de las variables categóricas	37
4.2. Modelización	39
4.2.1. Regresión logística con penalización	41
4.2.2. Regresión logística con penalización + PCA	42
4.2.3. Árboles de decisión	42
4.2.4. Bosques aleatorios	43
4.2.5. KNN	43
4.2.6. SVM lineal	44
4.2.7. SVM radial	44
4.3. Comparación	44
5. Aplicación de los modelos	49
5.1. Visión general del desempeño de los modelos	49
5.2. Caso estudio	51

6. Conclusión y Trabajo futuro	55
6.1. Aportaciones	55
6.2. Conclusiones	55
6.3. Trabajo futuro	55
A. Apéndice: Salidas	57
A.1. Gráficos espaciales EDA	57
A.1.1. Variables meteorológicas	57
A.1.2. Variables demográficas	61
A.1.3. Variable de vegetación	62
A.1.4. Variables topográficas	62
A.1.5. Variables antropológicas	64
A.2. Salidas de los modelos	67
A.2.1. Coeficientes regresión logística con penalización	67
A.2.2. VIP Bosque Aleatorio	67
B. Apéndice: Código	71
B.1. Generación de la muestra	71
B.2. Asignación de variables a localizaciones	75
B.3. Depuración de la muestra	85
B.4. Análisis exploratorio	88
B.4.1. Variable objetivo	88
B.4.2. Análisis univariantes	90
B.4.3. Análisis multivariantes	99
B.5. Modelos	101
B.5.1. Partición temporal entrenamiento / validación / test	102
B.5.2. Regresión logística con penalización	103
B.5.3. Regresión logística con penalización + PCA	104
B.5.4. Árboles de decisión	105
B.5.5. Bosques aleatorios	107
B.5.6. k-Nearest Neighbours	110
B.5.7. SVM lineal	111
B.5.8. SVM radial	112
B.5.9. Comparación	113
B.5.10. Test	115

B.6. Aplicación de los modelos	117
B.6.1. Visión general del desempeño del modelo	117
B.7. Caso de interés	121
Bibliografía	125

Prólogo

Escrito colocado al comienzo de una obra en el que se hacen comentarios sobre la obra o su autor, o se introduce en su lectura; a menudo está realizado por una persona distinta del autor.

También se podrían incluir aquí los agradecimientos.

Resumen

Resumen...

Abstract

Abstract...

Índice de figuras

2.1. Cadena de valor estadística. <i>Fuente: van der Loo y de Jonge [22]</i>	6
2.2. Flujo de trabajo en un proyecto de extitdata science. <i>Fuente: Wickham et al. [24]</i>	15
3.1. Áreas recorridas por el fuego entre 2002 y 2022 en incendios mayores de 100 hectáreas en Andalucía	19
3.2. Número total de incendios por mes durante el periodo de estudio	23
3.3. Observaciones para los que no está disponible alguna de las variables topográficas	26
4.1. Resumen numérico del conjunto de datos depurados	30
4.2. Distribución temporal de la variable objetivo	31
4.3. Distribución espacial de la variable objetivo	32
4.4. Boxplot de cada variable numérica en función de la variable objetivo	33
4.5. Media mensual de <i>T2M</i> en función de <i>fire</i>	33
4.6. Media mensual de <i>RH2M</i> en función de <i>fire</i>	34
4.7. Media mensual de <i>PRECTOTCORR</i> en función de <i>fire</i>	35
4.8. Media mensual de <i>GWETTOP</i> en función de <i>fire</i>	35
4.9. Media mensual de <i>WS10M</i> en función de <i>fire</i>	36
4.10. Media mensual de <i>NDVI</i> en función de <i>fire</i>	36
4.11. Correlaciones entre variables numéricas	38
4.12. Gráfico de coordenadas paralelas de las variables numéricas tipificadas . .	39
4.13. PCA sobre la matriz de correlaciones de las variables numéricas	39
4.14. Histogramas de las variables categóricas en función de <i>fire</i>	40
4.15. Métricas de rendimiento de los modelos de regresión logística con penalización	41
4.16. Métricas de rendimiento del árbol de decisión en función de α	42
4.17. Métricas de rendimiento de Random Forest en función de los parámetros .	43
4.18. Métricas de rendimiento de KNN en función de k	44
4.19. Métricas de rendimiento de SVM en función del parámetro <i>coste</i>	45

4.20. Gráfico de métricas obtenidas sobre el conjunto de validación	46
4.21. Curvas ROC sobre el conjunto de validación	47
4.22. Gráfico de métricas obtenidas sobre el conjunto test	47
4.23. Curvas ROC sobre test	48
 5.1. Malla de puntos con una resolución de 10km por 10km.	49
5.2. Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de regresión logística con penalización	50
5.3. Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de SVM lineal	51
5.4. Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de Random Forest	52
5.5. Área recorrida por el fuego en el incendio de Sierra Bermeja	52
5.6. Aplicación del modelo de regresión logística con penalización al incendio de Sierra Bermeja	53
 A.1. Distribución espacial de $T2M$ por mes	57
A.2. Distribución espacial de $RH2M$ por mes	58
A.3. Distribución espacial de $GWETTOP$ por mes	58
A.4. Distribución espacial de $WS10M$ por mes	59
A.5. Distribución espacial de $PRECTOTCORR$ por mes	59
A.6. Distribución espacial de $WD10M$ por mes	60
A.7. Distribución espacial de $poblacion$	61
A.8. Distribución espacial de $dens_poblacion$	61
A.9. Distribución espacial de $NDVI$ por mes	62
A.10. Distribución espacial de $elevacion$	62
A.11. Distribución espacial de $pendiente$	63
A.12. Distribución espacial de $curvatura$	63
A.13. Distribución espacial de $dist_carretera$	64
A.14. Distribución espacial de $dist_sendero$	64
A.15. Distribución espacial de $dist_camino$	65
A.16. Distribución espacial de $dist_poblacion$	65
A.17. Distribución espacial de $dist_electr$	66
A.18. Distribución espacial de $dist_ferrocarril$	66
A.19. Distribución espacial de uso_suelo	67
A.20. Coeficientes del modelos de regresión logística lasso seleccionado	68
A.21. Importancia de las variables en el bosque aleatorio final	69

Índice de tablas

3.1.	Conjuntos de datos brutos, tipos de datos y fuentes.	21
3.2.	Códigos de uso de suelo	25
3.3.	Variables del conjunto de datos depurados	27
4.1.	Métricas de los modelos seleccionados sobre el conjunto de validación . . .	45
4.2.	Métricas sobre el conjunto test	46

Capítulo 1

Introducción

1.1. Objetivos

El objetivo de esta investigación es construir modelos de *Machine Learning* que permitan predecir incendios forestales en la Comunidad Autónoma de Andalucía.

Subobjetivos:

1. Construir un conjunto de datos que permita la realización de análisis estadísticos y la posterior construcción de modelos de Machine Learning para la predicción de incendios forestales en Andalucía a partir de un estudio previo del problema.
2. Modelizar el riesgo de incendio forestal usando distintos algoritmos de ML y comparar sus resultados
3. Analizar el desempeño de los modelos en la realidad estudiando potenciales casos de interés.

1.2. Hipótesis

1.3. Revisión bibliográfica

Capítulo 2

Preliminares

2.1. Datos georreferenciados

Todos los datos empleados en este trabajo son datos georreferenciados, lo que significa que están asociados a ubicaciones geográficas específicas. Por ello, resulta esencial introducir los tipos de datos más utilizados para trabajar con esta información, sus características y las herramientas disponibles para manipularlos. Se tratarán los datos vectoriales y los datos ráster, al ser los tipos de datos fundamentales en este contexto, con características bien diferenciadas entre ellos.

2.1.1. Datos Vectoriales

El modelo de datos vectoriales se basa en puntos ubicados dentro de un sistema de referencia de coordenadas (CRS, por sus siglas en inglés). Estos puntos pueden representar características independientes o pueden estar conectados para formar geometrías más complejas como líneas y polígonos. Para esta sección se han usado como referencia Lovelace et al. [12].

Simple Features

Las *Simple Features* son un estándar abierto ampliamente utilizado para la representación de datos vectoriales, desarrollado y respaldado por el *Open Geospatial Consortium* (OGC), una organización sin ánimo de lucro dedicada a la creación de estándares abiertos e interoperables a nivel global dentro del marco de los sistemas geográficos de información (GIS, por sus siglas en inglés) y de la *World Wide Web*.

El paquete *sf* proporciona clases para datos vectoriales geográficos y una interfaz de línea de comandos consistente para importantes bibliotecas de bajo nivel para geoprocесamiento (*GDAL*, *PROJ*, *GEOS*, *S2*,...). [13]

Los objetos *sf* son fáciles de manipular ya que son dataframes o tibbles con dos características fundamentales. En primer lugar, contienen metadatos geográficos adicionales: tipo de geometría, dimensión, *Bounding Box* (límites o extensión geográfica) e información sobre el Sistema de referencia de coordenadas. Y además, presentan una columna de geometrías. Algunas ventajas del uso de este modelo de datos en R son que en la mayoría

de operaciones los objeto *sf* se pueden tratar como data frames, los nombres de las funciones son consistentes (todos empiezan por `st_`), las funciones se pueden combinar con el operador tubería y además funcionan bien con el ecosistema de paquetes *tidyverse*.

El paquete *sf* de R soporta 18 tipos de geometrías para las *simple features*, de las cuales las más utilizadas son: *POINT*, *LINESTRING*, *POLYGON*, *MULTIPOINT*, *MULTILINESTRING*, *MULTIPOLYGON* y *GEOMETRYCOLLECTION*.

2.1.2. Datos Ráster

El modelo de datos ráster representa el espacio con una cuadrícula de celdas (o píxeles) a cada una de las cuales se le asocia un valor o varios, tratándose así de rásteres de una o varias capas, respectivamente. Lo más común es trabajar con cuadrículas regulares, es decir, formadas por celdas rectangulares de igual tamaño. Sin embargo, existen otros modelos de ráster más complejos en los que se usan cuadrículas irregulares (rotadas, truncadas, rectilíneas o curvilíneas). [12]

Los datos en formato ráster constan de una cabecera y una matriz cuyos elementos representan celdas equiespaciadas. En la cabecera del ráster se definen el sistema de referencia de coordenadas, la extensión (o límites espaciales del área cubierta por el ráster), la resolución y el origen. El origen son las coordenadas de uno de los píxeles del ráster, que sirve de referencia para los demás, siendo generalmente utilizado el de la esquina inferior izquierda (aunque el paquete *TERRA* [9], usado en este trabajo, usa por defecto el de la esquina superior izquierda). La resolución se calcula como:

$$\text{resolution} = \frac{x_{\max} - x_{\min}}{\text{ncol}}, \frac{y_{\max} - y_{\min}}{\text{nrow}}$$

La representación en forma de matriz evita tener que almacenar explícitamente las coordenadas de cada una de las cuatro esquinas de cada pixel, debiendo almacenar solamente las coordenadas de un punto (el origen). Esto, unido a las operaciones del álgebra de mapas hacen que el procesamiento de datos ráster sea mucho más eficiente que el de datos vectoriales.

Se usará el paquete *TERRA* para tratar los datos en formato ráster. Este paquete permite tratar el modelo de rásteres regulares con una o varias capas a través de la clase de objetos **SpatRaster**. Sin embargo, existen otras alternativas, como el paquete *stars* [14], que además de ser más potente, permite trabajar con rásteres no regulares y ofrece una mejor integración con el paquete *sf* y el entorno *tidyverse*.

2.1.3. Sistemas de Referencia de Coordenadas

Intrínseco a cualquier modelo de datos espaciales está el concepto de Sistema de referencia de coordenadas (CRS), que establece cómo la geometría de los datos se relaciona con la superficie terrestre. Es decir, es el nexo de unión entre el modelo de datos y la realidad, por lo que juega un papel fundamental. Los CRS pueden ser de dos tipos: geográficos o proyectados. En esta sección se usan como referencia el Capítulo 9 de Gimond [6] y el Capítulo 2 de Lovelace et al. [12].

Sistemas de Coordenadas Geográficas

Los sistemas de coordenadas geográficas (GCS) identifican cada punto de la superficie terrestre utilizando la longitud y la latitud. La longitud es la distancia angular al Meridiano de Greenwich medida en la dirección Este-Oeste. La latitud es la distancia angular al Ecuador medida en la dirección Sur-Norte.

Cualquier sistema de coordenadas geográficas se compone de tres elementos: el elipsoide, el geoide y el *datum*. El primero es el elipsoide (o esfera) utilizado para representar de forma simplificada la superficie terrestre, sobre el que se supone que se encuentran los datos y que permitirá realizar mediciones. El segundo, el geoide, es el modelo matemático que representa la verdadera forma de la Tierra, que no es suave sino que presenta ondulaciones debidas a las fluctuaciones del campo gravitatorio a lo largo de la superficie terrestre, que además cambian a una amplia escala temporal. Y el tercero, el *datum*, indica cómo se alinean el elipsoide y el geoide, es decir, cómo el modelo matemático se ajusta a la realidad. Este puede ser local o geocéntrico, en función de si el elipsoide se ajusta al geoide en un punto concreto de la superficie terrestre o de si es el centro del elipsoide el que se alinea con el centro de la Tierra. Ejemplos de *datum* geocéntricos usados en este trabajo son:

- *European Terrestrial Reference System 1989* (ETRS89), usado ampliamente en la Europa Occidental.
- *World Geodetic System 1984* (WGS84), usado a nivel global.

Sistemas de Coordenadas Proyectadas

Un Sistema de Coordenadas Proyectadas (PCS) es un sistema de referencia que permite identificar localizaciones terrestres y realizar mediciones en una superficie plana, es decir, en un mapa. Estos sistemas de coordenadas se basan en las coordenadas cartesianas, por lo que tienen un origen, un eje X y un eje Y y usan una unidad lineal de medida (en este trabajo se usará el metro). Pasar de una superficie elíptica (GCR) a una superficie plana (PCS) requiere de transformaciones matemáticas apropiadas y siempre induce deformaciones en los datos.

Al proyectar la superficie terrestre en una superficie plana siempre se modifican algunas propiedades de los objetos, como el área, la dirección, la distancia o la forma. Un PCS solo puede conservar alguna de estas propiedades pero no todas, por lo que es habitual clasificar los PCS en función de la propiedad que mantienen: las proyecciones de igual área preservan el área, las azimutales preservan la dirección, las equidistantes preservan la distancia y las conformales preservan la forma local. En función de como se realice la proyección, estas también se pueden clasificar en planas, cilíndricas o cónicas.

Un caso particular y ampliamente usado de PCS cilíndrico son los *Universal Transverse Mercator* (UTM), en los que se proyecta el elipsoide sobre un cilindro tangente a este por las líneas de longitud (los meridianos). De esta forma, se divide el globo en 60 zonas de 6° de longitud, para cada una de las cuales existe un PCS UTM correspondiente que está asociado al meridiano central. Se trata de proyecciones conformales, por lo que preservan ángulos y formas en pequeñas regiones, pero distorsionan distancias y áreas.

A lo largo de este trabajo se utilizará ampliamente el sistema de coordenadas proyectadas UTM30N (es habitual especificar el hemisferio para evitar confusión en los valores del eje Y, ya que miden distancia al ecuador, de ahí la N de hemisferio norte).

2.2. Análisis Exploratorio de Datos

El Análisis Exploratorio de Datos (EDA), es una parte fundamental de todo proyecto de *Machine Learning* y en general de cualquier proyecto en el que se deba trabajar con datos de cualquier procedencia para extraer de ellos conclusiones. Antes del procesamiento de los datos es siempre necesario explorar, entender y evaluar la calidad de estos, pues como indica la expresión inglesa *garbage in, garbage out*, si trabajamos con datos pobres, no podemos esperar obtener de ellos buenos resultados. [23]

El EDA hace referencia al conjunto de técnicas estadísticas (tanto numéricas como gráficas) con las que se pretende explorar, describir y resumir la naturaleza de los datos, comprender las relaciones existentes entre las distintas variables presentes, identificar posibles errores o revelar posibles valores atípicos. Todo esto con el objetivo de maximizar nuestra compresión sobre el conjunto de datos.

2.2.1. Depuración de los datos

La depuración de los datos o *data cleaning* es el proceso de detectar y corregir o eliminar datos incorrectos, corruptos, con formato incorrecto, duplicados o incompletos dentro de un conjunto de datos. Puede considerarse una fase dentro del EDA (como se sugiere en Wickham y Grolemund [23]) o una fase previa a este.

Puede entenderse que el *data cleaning* es el proceso de pasar de *raw data* o datos en bruto a datos técnicamente correctos y finalmente a datos consistentes.

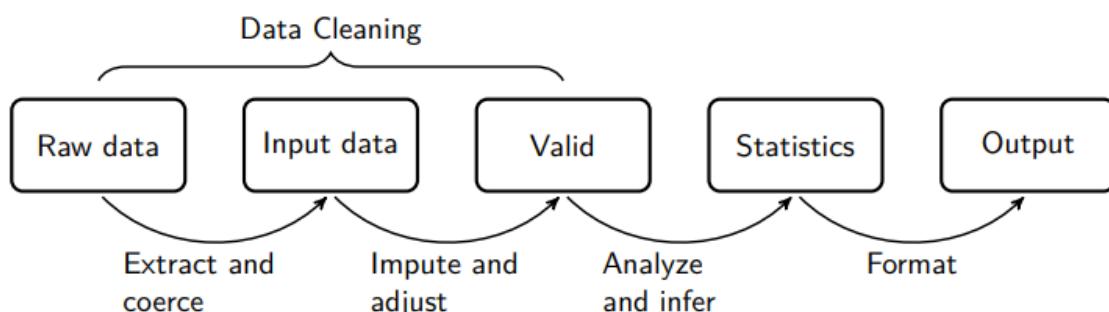


Figura 2.1: Cadena de valor estadística. Fuente: van der Loo y de Jonge [22]

Entendemos que un conjunto de datos es técnicamente correcto cuando cada valor pertenece a una variable y está almacenado en el tipo que le corresponde en base al conocimiento del dominio del problema. Para ello se debe reajustar el tipo de cada variable al que le corresponda en base al conocimiento que se tenga sobre esta, codificando los valores en las clases adecuadas si fuese necesario.

Decimos que un conjunto de datos es consistente cuando es técnicamente correcto y, además, adecuado para el análisis estadístico. Se trata, por tanto, de datos que han eliminado, corregido o imputado los valores faltantes, los valores especiales, los valores atípicos y los errores.

2.2.2. Análisis de Componentes Principales

El Análisis de Componentes Principales (PCA) es una técnica de reducción de la dimensionalidad ampliamente usada en el análisis de datos multivariante. Al emplear técnicas de reducción de dimensionalidad como PCA, se persiguen diversos objetivos: eliminar correlaciones redundantes, reducir el ruido presente en los datos y facilitar el uso de algoritmos cuya eficiencia computacional está fuertemente influenciada por la dimensionalidad de los datos. A continuación, se definen los conceptos fundamentales del Análisis de Componentes Principales siguiendo el enfoque expuesto en Shao y Deng [16].

Definición 2.2.1 Dadas $\underline{x}_1, \dots, \underline{x}_n \in \mathbb{R}^k$ realizaciones de un vector aleatorio \underline{X} en \mathbb{R}^k , se dice que los vectores $\underline{c}_1, \underline{c}_2, \dots, \underline{c}_p \in \mathbb{R}^k$ son las k componentes principales (muestrales) del vector aleatorio \underline{X} si forman una base ortonormal del espacio $V \subset \mathbb{R}^k$ de dimensión p que minimiza la media del cuadrado de la distancia euclídea entre los \underline{x}_i y su proyección $\pi_V(\underline{x}_i)$ sobre V .

De la propia definición se desprende que las componentes principales no son únicas.

Proposición 2.2.1 Las p primeras componentes principales se corresponden con los autovectores unitarios asociados a los p mayores autovalores λ_j de la matriz de covarianzas muestrales.

Definición 2.2.2 Se define la fracción de varianza explicada por las p primeras componentes principales como $\frac{\sum_{j=1}^p \lambda_j}{\sum_{j=1}^k \lambda_j}$.

El número de componentes principales necesarias para explicar un porcentaje dado de la varianza de los datos puede servir para caracterizar la magnitud del problema.

2.3. Modelos

El problema que se aborda en este trabajo se engloba dentro de lo que se conoce como aprendizaje supervisado, ya que para cada observación del conjunto de entrenamiento se conoce el valor de la variable objetivo (en este caso si ha habido incendio o no). Más concretamente, se trata de un problema de clasificación binaria, ya que el objetivo es asignar cada observación a una de las dos clases posibles (incendio o no incendio). Existen numerosas técnicas de clasificación binaria supervisada, en este trabajo se explorarán algunas de las de uso más común en problemas similares. Las principales fuentes consultadas para esta sección han sido Hastie et al. [7], el capítulo 6 de Suthaharan [19] y entradas del blog Awati [2].

2.3.1. Regresión logística (con penalización)

La regresión logística es un caso particular de modelo lineal generalizado basado en las siguientes hipótesis:

- Hipótesis distribucional. Dadas las variables explicativas, \underline{X}_i con $i = 1, 2, \dots, n$, se

verifica que las variables $Y|_{\underline{X}=\underline{x}_i}$ y su distribución pertenece a la familia Bernouilli, es decir,

$$Y|_{\underline{X}=\underline{x}_i} \sim Be(\pi(\underline{x}_i))$$

- Hipótesis estructural. La esperanza $E(Y|_{\underline{X}=\underline{x}_i}) = \pi_i$ está relacionada con un predictor lineal ($\eta_i = \underline{\beta}^t \underline{z}_i$) a través de la función *logit* (con $\underline{z}_i = (1, \underline{x}_i)$) Es decir, dado que

$$\eta_i = \underline{\beta}^t \underline{z}_i = \ln\left(\frac{\pi_i}{1 - \pi_i}\right)$$

O equivalentemente,

$$\pi_i = \frac{\exp(\underline{\beta}^t \underline{z}_i)}{1 + \exp(\underline{\beta}^t \underline{z}_i)}$$

Bajo estas hipótesis, la función de log-verosimilitud dada una muestra $\{(x_i, y_i)\}_{i=1,\dots,n}$ es:

$$l(\underline{\beta}) = \sum_{i=1}^n \left[y_i \ln\left(\frac{\pi_i}{1 - \pi_i}\right) + \ln(1 - \pi_i) \right]$$

En la regresión logística clásica se estima el vector de parámetros $\underline{\beta}$ maximizando la función de log-verosimilud, o lo que es equivalente, minimizando su opuesta. Por tanto, el problema de optimización a resolver será

$$\min_{\underline{\beta}} -l(\underline{\beta})$$

Sin embargo, con el objetivo de evitar el sobreajuste y construir modelos con mayor capacidad de generalización existen variaciones de la regresión logística que incluyen un término de penalización en la función objetivo. Las dos variantes de uso más extendido son la regresión *ridge* y *lasso*.

Sea $\underline{\beta} = (\beta_0, \underline{\beta}_1)$, donde $\underline{\beta}_1$ contiene los coeficientes de las covariables. En la regresión *ridge* el término de penalización es de la forma $\|\underline{\beta}_1\|_2^2$ mientras que en la regresión *lasso* el penalización es de la forma $\|\underline{\beta}_1\|_1$. Por tanto, el problema de optimización será

$$\min_{\underline{\beta}} -l(\underline{\beta}) + \lambda \sum \beta_i^2$$

en el caso de la regresión logística *ridge* y

$$\min_{\underline{\beta}} -l(\underline{\beta}) + \lambda \sum |\beta_i|$$

en el caso de la regresión logística *lasso*.

En este trabajo se usará el paquete *glmnet* [20], que implementa una combinación de ambos métodos (llamada *elastic net*), en la que se añade un parámetro de mixtura $\alpha \in [0, 1]$ que combina ambos enfoques. El problema de optimización resultante en este caso será:

$$\min_{\underline{\beta}} -l(\underline{\beta}) + \lambda \left[(1 - \alpha) \sum \beta_i^2 + \alpha \sum |\beta_i| \right]$$

2.3.2. Support Vector Machine

Las Máquinas de Vector Soporte (SVM) son una familia de modelos principalmente usados en problemas de clasificación binaria (si bien se pueden extender a problemas de clasificación multiclas o de regresión) que parten de la idea de encontrar el hiperplano que “mejor” separa al conjunto de puntos.

2.3.2.1. SVM lineal

Dada una muestra $\{(\underline{x}_i, y_i)\}_{i=1,\dots,n}$ con $\underline{x}_i \in \mathbb{R}^d$ y $y_i \in \{-1, 1\}$ para todo $i \in \{1, \dots, n\}$, el objetivo es encontrar al hiperplano de la forma

$$h(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b = \underline{w}^t\underline{x} = 0$$

que mejor separe a la muestra.

Definición 2.3.1 Se dice que la muestra es linealmente separable si existe un hiperplano, denominado hiperplano de separación, que cumple, para todo $i \in \{1, \dots, n\}$:

$$\underline{w}^t\underline{x}_i + b \geq 0 \text{ si } y_i = +1$$

$$\underline{w}^t\underline{x}_i + b \leq 0 \text{ si } y_i = -1$$

Definición 2.3.2 Dado un hiperplano de separación de una muestra linealmente separable, se define el margen como la menor de las distancias del hiperplano a cualquier elemento de la muestra. Se denotará por τ .

Proposición 2.3.1 Dado un punto \underline{x}_i y un hiperplano $h(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b = \underline{w}^t\underline{x} = 0$, la distancia entre ambos viene dada por:

$$d(h, \underline{x}_i) = \frac{|h(\underline{x}_i)|}{\|\underline{w}\|} = \frac{y_i(\underline{w}^t\underline{x}_i + b)}{\|\underline{w}\|}$$

donde $\|\cdot\|$ hace referencia a la norma euclídea.

Proposición 2.3.2 Dada una muestra linealmente separable $\{(\underline{x}_i, y_i)\}_{i=1,\dots,n}$ con $\underline{x}_i \in \mathbb{R}^d$ y $y_i \in \{-1, 1\}$ y un hiperplano de separación $h(x) = \underline{w}^t\underline{x} = 0$ con margen τ , se verifica que

$$\frac{y_i(\underline{w}^t\underline{x}_i + b)}{\|\underline{w}\|} \geq \tau \quad \forall i \in \{1, \dots, n\}$$

O equivalentemente,

$$y_i(\underline{w}^t\underline{x}_i + b) \geq \tau \|\underline{w}\| \quad \forall i \in \{1, \dots, n\}$$

Y, además, es posible reescribir el mismo hiperplano h de forma que $\tau \|\underline{w}\| = 1$.

De esta última expresión se deduce que maximizar el margen τ es equivalente a minimizar la norma euclídea de \underline{w} . Por tanto, para encontrar el hiperplano de separación óptimo para una muestra en las condiciones de la proposición anterior basta resolver el problema de optimización siguiente:

$$\begin{aligned}
 & \min_{w,b} \quad \frac{1}{2} w^t w \\
 \text{s.t.} \quad & w^t x_i + b \geq 1, \quad \forall i \in \{1, \dots, n\} \\
 & w \in \mathbb{R}^d, b \in \mathbb{R}
 \end{aligned} \tag{2.1}$$

En general, las muestras no son separables, por lo que es necesario permitir que pueda haber casos mal clasificados, y penalizarlos proporcionalmente a la distancia a la que se encuentren del subespacio correcto (holgura). Para ello, se introducen en la formulación del modelo las variables artificiales ξ_i , $i = 1, \dots, n$. Se habla entonces de hiperplano de separación *soft margin*. De esta forma, se llega al problema de optimización siguiente:

$$\begin{aligned}
 & \min_{w,b,\xi} \quad \frac{1}{2} w^t w + C \sum_{i=1}^n \xi_i \\
 \text{s.t.} \quad & w^t x_i + b \geq 1, \quad \forall i \in \{1, \dots, n\} \\
 & \xi \geq 0, \quad \forall i \in \{1, \dots, n\} \\
 & w \in \mathbb{R}^d, b \in \mathbb{R}
 \end{aligned} \tag{2.2}$$

donde $C > 0$ es un parámetro de regularización que permite controlar los errores de clasificación permitidos por el modelo, controlando así el sobreajuste. Este parámetro recibe el nombre de coste (*cost*).

2.3.2.2. SVM no lineal

Existen muchos casos en los que el SVM no es capaz de obtener buenos resultados, debido a la estructura de la distribución de las clases en la muestra. En estos casos, es común recurrir a una técnica llamada *kernel trick*. Esta técnica consiste en realizar una inmersión del conjunto de los vectores de la muestra en un espacio de dimensión superior (llamado *feature space*) en el que los casos sí sean separables (o al menos mejore la separabilidad de estos). Esta inmersión en un espacio de dimensión superior se hace indirectamente a través de funciones *kernel*, que calculan los productos escalares entre los vectores de la muestra en el espacio de inmersión. Existen distintos tipos de funciones *kernel* que se corresponden con distintas inmersiones en espacios de dimensión superior:

- Kernel polinomial: $k(x, z) = (\gamma(x^t z + c_0))^p$
- Kernel RDF (radial o gaussiano): $k(x, z) = \exp(-\gamma \|x - z\|^2)$

2.3.3. Decision Trees

Un árbol de decisión (DT) es un algoritmo de aprendizaje supervisado no paramétrico, que puede aplicarse tanto a problemas de clasificación como de regresión. La idea de este método es segmentar el espacio predictor en rectángulos, de forma que para predecir una observación se usa la moda (o la media) de la región a la que pertenece. Se trata de un modelo jerárquico con estructura de árbol, que consta de un nodo raíz, ramas, nodos internos y nodos hojas. Cada nodo representa un test sobre una variable, y cada las ramas que nacen de ese nodo representan los posibles valores que puede tomar esa

variable. De esta forma, para clasificar una nueva instancia basta comenzar en el nodo raíz e ir descendiendo por el árbol hasta llegar al nodo hoja correspondiente, que indicará la clasificación asignada a dicha instancia. La simplicidad del método muestra su principal ventaja, su fácil comprensión dada su estructura de árbol.

Existen diversas técnicas para construir árboles de clasificación (y regresión), aquí se ilustra una de las más usadas que recibe el nombre de CART (*Classification And Regression Trees*, Breiman et al. [4]). Se explica para el caso de árboles de clasificación binarios, es decir, en los que de cada nodo salen dos ramas.

Dada una muestra $\{(\underline{x}_i, y_i)\}$ con $\underline{x}_i = (x_{i1}, \dots, x_{id})$, un árbol de clasificación con J hojas se puede expresar como

$$f(\underline{x}) = \sum_{j=1}^J c_j I(\underline{x} \in R_j)$$

donde $\{R_j\}_{j=1, \dots, J}$ es una partición del espacio predictivo y c_j es la clase asignada en R_j para todo $j \in 1, \dots, J$.

En la práctica, c_j se estima asignando la clase mayoritaria en el recinto R_j . Es decir, $\hat{c}_j = \text{moda}(\{y_i | \underline{x}_i \in R_j\})$.

Para construir un árbol de clasificación, el algoritmo necesita decidir las variables tests y los puntos de corte en cada nodo, así como la topología del árbol. Para realizar esto, se vale de un método *greedy*, que en cada nodo elige la variable y el punto de corte que mejor separan los datos en base a una medida de impureza. Es decir, la construcción de un árbol de clasificación no se hace mediante la resolución de un solo problema de optimización global, si no a partir de la resolución de muchos problemas de optimización locales, con las implicaciones que esto pueda tener.

Las medidas de impureza más comúnmente usadas son:

- Error de clasificación: $\Phi(p) = 1 - \max(p, 1 - p)$
- Índice de Gini: $\Phi(p) = 2p(1 - p)$
- Entropía: $\Phi(p) = -p \log p - (1 - p) \log(1 - p)$

donde p denota la proporción de casos positivos en la muestra.

Así, el algoritmo de construcción de un árbol de clasificación es:

1. Comenzar con el nodo raíz, que incluye todos los casos.
2. Determinar el par (variable,corte) que conduce a una mayor reducción de la impureza. Es decir, dada una medida de impureza Φ se busca la variable $j \in 1, \dots, d$ y el corte $s \in \mathbb{R}$ solución de

$$\min_{j \in 1, \dots, d, s \in \mathbb{R}} \left[\frac{|R_1|}{|R_1| + |R_2|} \Phi(\{y_i | \underline{x}_i \in R_1(j, s)\}) + \frac{|R_2|}{|R_1| + |R_2|} \Phi(\{y_i | \underline{x}_i \in R_2(j, s)\}) \right]$$

donde $R_1(j, s) = \{X | X_j \leq s\}$ y $R_2(j, s) = \{X | X_j > s\}$.

3. Aplicar iterativamente el proceso anterior a cada nuevo nodo, hasta que se verifiquen las condiciones de finalización. En este caso, el criterio será finalizar el proceso de división en el nodo una vez que el número de casos en este sea igual o inferior a una cantidad n_{min} fijada de antemano. En los nodos hoja se asigna la clase mayoritaria en el nodo.
4. Podar o recortar el árbol obtenido en base a un criterio de coste-complejidad. Dado un árbol completo T y un valor del parámetro de coste-complejidad α , se elije el subárbol $T_0 \subset T$ obtenido a partir de T mediante poda, es decir, colapsando nodos no terminales, que minimice el criterio de coste complejidad definido como:

$$C_\alpha(T) = \Phi(T) + \alpha|T_0|$$

El parámetro α permite controlar la capacidad de generalización del modelo (*Bias-Variance tradeoff*) y se estima mediante Validación Cruzada.

El gran inconveniente de los árboles de decisión es que en general son modelos con una varianza elevada, por lo que tienden a ser inestables y a producir sobreajuste. Para evitar esto, se recurre al uso de técnicas de *Bagging* y *Boosting*. Una de las técnicas más extendida con árboles de decisión son los Bosques Aleatorios (*Random Forest* en inglés).

2.3.4. Random Forest

La idea detrás del modelo de bosques aleatorios es reducir la varianza de los árboles de decisión sin aumentar el sesgo. Para intentar conseguir este objetivo, la idea es aplicar *Bagging* (*Bootstrap Aggregating*) al modelo de árbol de decisión. Sin embargo, ya que al aplicar *Bagging* la reducción de la varianza es mayor cuanto más incorrelados sean los predictores individuales, en cada nuevo nodo de cada árbol construido se selecciona la variable que más disminuya la impureza de entre un conjunto aleatorio de $m_{try} < d$ predictores.

El algoritmo para construir un bosque aleatorio es el siguiente:

1. Para $b = 1, \dots, B$:
 - a) Seleccionar una muestra bootstrap Z^* de tamaño n del conjunto de entrenamiento.
 - b) Construir un árbol de decisión T_b a partir de la muestra bootstrap b , aplicando recursivamente los siguiente pasos para cada nodo terminan del árbol, hasta que se alcance el tamaño mínimo de nodo n_{min} :
 - I. Seleccionar aleatoriamente m_{try} variables de entre las d variables predictoras.
 - II. Elegir el mejor par variable/división de entre las m_{try} variables seleccionadas en función de la reducción del criterio de impureza.
 - III. Dividir el nodo en dos nodos hijos.
2. De esta forma se obtiene el conjunto de árboles de decisión bootstrap $\{T_b\}_{b=1}^B$.

Para predecir la clase de un nuevo punto \underline{x} se aplica la regla de la clase más votada al conjunto de clases predichas por los B árboles de decisión bootstrap para \underline{x} .

2.3.5. K-Nearest Neighbors

El método de k vecinos más cercanos (KNN) clasifica una nueva observación \underline{x} en base a las clases de las k observaciones del conjunto de entrenamiento más cercanas a estas en el espacio muestral aplicando la regla de la clase más votada. Es decir, dado un espacio muestral Θ con una distancia d definida sobre él, dado un conjunto de entrenamiento $T \subset Y$ y dado $k \in \mathbb{N}^{+*}$ la función calculada por el algoritmo para estimar la clase de $\underline{x} \in \Theta$ es:

$$f(\underline{x}) = \text{majority vote } \{y_i \mid \underline{x}_i \in N_k(\underline{x})\}$$

donde $N_k(\underline{x})$ es el conjunto de los k puntos $\underline{x}_i \in \Theta$ más próximos a \underline{x} en Θ en base a la distancia d .

El parámetro k permite controlar el sobreajuste del modelo.

2.4. Validación del ajuste

Para validar el ajuste de los modelos comentados en los datos, se utilizará una partición temporal en entrenamiento/ validación/ test. Es decir, se asignará el primer 60 % de los datos (de acuerdo al día de la observación) a entrenamiento, el 20 % siguiente a validación y el último 20 % a test. Este enfoque permite evitar el sesgo positivo debido al efecto *look-ahead* en la estimación de la capacidad de generalización de los modelos.

2.5. Evaluación de los modelos

Una vez construido un modelo predictivo es necesario conocer el rendimiento de este sobre nuevos datos, con el objetivo de estimar su capacidad de generalización. Esto es fundamental de cara a determinar si el modelo es adecuado para el propósito previsto o si necesita ajustes o mejoras. La evaluación del rendimiento permite comparar entre diferentes modelos y seleccionar el que mejor se adapte a las necesidades específicas del problema en cuestión. Para ello, se recurre al uso de distintas métricas, en función de las características propias de cada problema.

2.5.1. Clasificación binaria

En el presente trabajo el problema que se aborda es un problema de clasificación binaria, pues tenemos solo dos clases que son la clase positiva y la clase negativa. A la hora de clasificar una nueva instancia pueden darse 4 situaciones:

- Que se clasifique como positiva siendo realmente positiva, en cuyo caso se dirá que forma parte de las *True Positives (TP)*
- Que se clasifique como negativa siendo realmente negativa, en cuyo caso se dirá que forma parte de las *True Negatives (TN)*

- Que se clasifique como positiva siendo realmente negativa, en cuyo caso se dirá que forma parte de las *False Positives (FP)*
- Que se clasifique como negativa siendo realmente positiva, en cuyo caso se dirá que forma parte de las *False Negatives (FN)*

Se considerarán las siguientes métricas de rendimiento para problemas de clasificación binaria:

Tasa de acierto o exactitud. Mide la proporción de casos que han sido correctamente clasificados.

$$\text{Exactitud} = \frac{TP + TN}{TP + FP + TN + FN}$$

Precisión. Mide la proporción de casos clasificados como positivos que realmente lo son.

$$\text{Precisión} = \frac{TP}{TP + FP}$$

Especificidad. Mide la proporción de casos negativos que han sido correctamente clasificados por el modelo.

$$\text{Especificidad} = \frac{TN}{TN + FP}$$

Sensibilidad o recall. Mide la proporción de casos positivos que han sido correctamente clasificados por el modelo.

$$\text{Recall} = \frac{TP}{TP + FN}$$

AUC-ROC. Mide el área bajo la curva ROC (*Receiver Operating Characteristic* o Característica Operativa del Receptor en castellano). Esta curva es una representación gráfica del rendimiento de un modelo de clasificación binaria para todos los umbrales de clasificación. Representa la sensibilidad frente a la proporción de falsos positivos para cada posible umbral de clasificación. El AUC está comprendido entre 0 y 1, entendiéndose que el rendimiento del es mejor cuanto mayor sea su valor. En general, se suelen considerar aceptables modelos con un valor del AUC superior a 0.75.

2.6. Herramientas

Toda la parte práctica del presente trabajo se ha llevado a cabo empleado el lenguaje de programación R [15] a través del entorno de desarrollo integrado que ofrece RStudio. R es un lenguaje y entorno de programación de código abierto desarrollado dentro del proyecto GNU y orientado a la computación estadística. Este lenguaje puede extender sus funcionalidades fácilmente a través de la gran cantidad de paquetes disponibles dentro del repositorio de paquetes de CRAN (*The Comprehensive R Archive Network*), siendo este uno de sus puntos fuertes, dada la gran comunidad de usuarios y desarrolladores con la que cuenta. A continuación se mencionan los principales paquetes utilizados en el trabajo.

A lo largo de todo el trabajo, se ha utilizado de forma central el ecosistema de paquetes *tidyverse* [24]. Se trata de una colección de paquetes de R que comparten las mismas estructuras de datos y la misma filosofía de programación, orientados a facilitar las tareas centrales de cualquier proyecto de *data science* (Figura 2.2). Cuenta con numerosos paquetes, entre los que destacan: *readr* para la importación de datos tabulares; *tidyr* para la ordenación; *dplyr* para la manipulación; *lubridate* para las fechas; *forcats* para los factores; *ggplot2* para la visualización; *purr* para la programación funcional; y *tibble* que proporciona la estructura tabular de datos sobre la que trabajar.

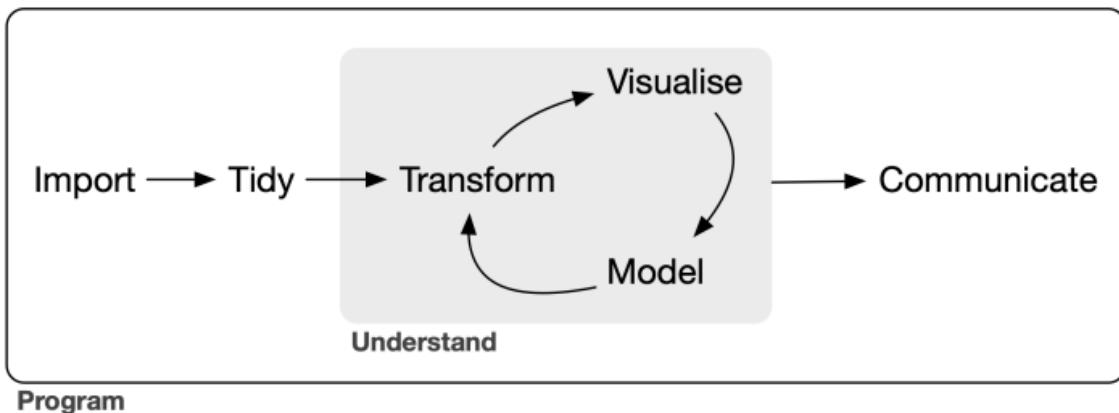


Figura 2.2: Flujo de trabajo en un proyecto de extitdata science. Fuente: Wickham et al. [24]

Para la modelización se ha utilizado *tidymodels* [10]. Se trata de un conjunto de paquetes de R orientados a facilitar el flujo de trabajo en el modelado estadístico de datos, siguiendo la filosofía “tidy data” de *tidyverse*. Integra herramientas para preprocesamiento de datos (*recipes*), ajuste de modelos (*parsnip*), combinación de flujos de trabajo (*workflows*), partición de datos y validación cruzada (*rsample*), cálculo de métricas de rendimiento (*yardstick*), optimización de hiperparámetros (*tune*) y conversión de resultados de modelos a formatos ordenados (*broom*). Los paquetes que componen *tidymodels* no implementan los modelos estadísticos por sí mismos, si no que los importan de librerías específicas. En su lugar, se enfocan en facilitar el flujo de trabajo del modelado, mejorando la cohesión y la eficiencia, y permitiendo la integración natural con otros paquetes del *tidiverse*.

Como se detalló en la Sección Datos georreferenciados, para el tratamiento de datos vectoriales se ha utilizado el paquete de R *sf* [13], que proporciona una forma estandarizada de codificar y manipular datos espaciales vectoriales a través de las *simple features*, integrándose dentro del ecosistema *tidyverse*. Para la manipulación de datos de tipo ráster se ha recurrido al paquete *terra* [9].

La descarga de información meteorológica satelital se ha realizado a través del paquete *nasapower* [17], que facilita el acceso a información meteorológica global de forma reproducible a través de R. El paquete *mapSpain*[8] ha facilitado el acceso a las fronteras administrativas de España a distintos niveles de desagregación (comunidad autónoma, provincia y municipio). Se basa en el *ISCO Eurostat database* (<https://ec.europa.eu/eurostat/web/gisco>) y en el *CartoBase SIANE* del Instituto Geográfico Nacional (<https://www.ign.es/>).

Capítulo 3

Construcción del conjunto de datos

El primer paso a la hora de construir cualquier modelo de predicción es disponer de datos adecuados que permitan explicar correctamente el fenómeno en estudio, en este caso los incendios forestales en Andalucía. Con este fin, se ha llevado a cabo un extenso estudio previo del dominio del problema para conocer qué variables son relevantes de cara a la predicción de incendios forestales, analizando estudios similares realizados anteriormente así como otras fuentes relativas a la ecología del fuego, que nos permitiesen conocer el efecto que cabría esperar de estas variables.

Se ha querido adoptar un enfoque dinámico, es decir, el objetivo no es construir un modelo estacionario que nos indique si una determinada zona se verá afectada por un incendio forestal a lo largo de un amplio periodo temporal, si no que se pretende ser capaz de predecir si un determinado punto del territorio se verá afectado por un incendio forestal en un momento concreto, en base a las covariables correspondientes a ese lugar en ese momento. Es decir, se considera no solo la dimensión espacial de los datos si no también la temporal, al mayor nivel de desagregación disponible. Este es un enfoque mucho menos explorado, debido fundamentalmente a dos factores:

1. La dificultad de disponer de información fiable y de calidad desagregada espacio-temporalmente
2. La dificultad de trabajar con datos de estas características de cara al análisis y principalmente a la modelización, ya que son datos correlados en el tiempo y en el espacio.

Queda claro, por tanto, que se trata de un problema complejo que requiere de simplificaciones para poder ser abordado, más aun dadas las limitaciones en los recursos computacionales disponibles y la enorme cantidad de datos que se están considerando y que requieren de un procesamiento sumamente costoso desde un punto de vista computacional.

Por todo ello, esta sección es probablemente la de mayor importancia y dificultad de todo el trabajo, ya que implica la toma de decisiones que serán determinantes de cara al correcto desempeño de los modelos que se construirán más adelante, requiere de un vasto conocimiento del problema que permita un enfoque adecuado que haga posible la consecución de los objetivos que se esperan conseguir, necesita del uso de técnicas específicas de procesamiento de datos espaciales que no han sido tratadas durante el grado y se ve fuertemente limitada por los escasos recursos computacionales disponibles.

3.1. Determinación del marco del estudio

El primer paso ha sido limitar el área y la franja temporal que abarcará el estudio. Para ello, ha sido necesario basarse principalmente en la disponibilidad y consistencia de la información requerida para el proyecto y en las limitaciones computacionales impuestas por el equipo disponible.

En cuanto a la disponibilidad de información, hay que diferenciar entre la información de incendios forestales y la información de variables que permitan explicar este fenómeno considerando la mayor desagregación espacial y temporal posible.

3.1.1. Incendios forestales

En lo referente a los datos sobre incendios forestales cabe mencionar que España cuenta con una de las mayores y más completas bases de datos sobre incendios forestales a nivel europeo. Se trata de la Estadística General de Incendios Forestales (EGIF), que en su versión definitiva actualmente contiene toda la información que se recoge en cada parte de incendio forestal que ha tenido lugar en España desde 1983 hasta 2015, incluyendo su información espacial con sus coordenadas de origen. Se ha explorado extensamente el uso de esta base de datos para el proyecto, dada su exhaustividad y completitud. Sin embargo, lamentablemente no ha sido posible en este caso incorporarla al trabajo por diversas razones.

La principal de ellas fue que hasta marzo de 2024 la base de datos de la EGIF solo se encontraba disponible en el Catálogo de Datos del Gobierno de España en formato TURTLE¹ y esto conllevó numerosas dificultades. Se exploraron distintas librerías de R (y alguna de Python) para el manejo de datos en este formato, como *RDFlib* [3]. Sin embargo, al tratarse de una base de datos de un tamaño considerable (aproximadamente 1GB y con más de una decena de millones de triplets), esta librería no era suficientemente eficiente para poder realizar consultas en un tiempo razonable al conjunto de datos. Tras explorar otras alternativas, se valoró la posibilidad de usar un *triplesstore*, es decir, una base de datos especialmente diseñada para el almacenamiento y recuperación de triplets a través de consultas semánticas. En este caso se usó *Apache Jena Fuseki*, ya que cuenta con una interfaz que facilita su uso. Sin embargo, aunque esto supuso una mejora considerable en la eficiencia y permitió realizar consultas sencillas a la base de datos, en este caso fue la complejidad del gráfico de datos (ontología) y la escasa documentación disponible sobre esta, la que impidió que se pudiesen realizar las consultas más complejas requeridas para llevar a cabo el proyecto. Además, se debe tener en cuenta que se trata de una base de datos muy heterogénea y con numerosos datos faltantes debida su naturaleza, por lo que requiere de un preprocesamiento que probablemente será complicado y costoso en tiempo y en recursos computacionales. Al no disponer de ninguno de estos, finalmente se optó por buscar una alternativa más abordable dada las limitaciones con las que cuenta un Trabajo de Fin de Estudios, aunque queda abierta la posibilidad de explorar esta base de datos en futuros estudios, la cual podrá aportar nuevas dimensiones al estudio de los incendios forestales en España gracias a la enorme cantidad de información que ofrece.

¹TURTLE es una sintaxis para RDF con compatible con SPARQL. RDF (*Resource Description Framework*) es un estándar de semántica web utilizado para el intercambiar de datos en la Web.

Ante esta situación, la solución planteada fue limitar el área en estudio a la Comunidad Autónoma de Andalucía, aprovechando la enorme disponibilidad de información medioambiental que ofrece la Red de Información Ambiental de Andalucía (REDIAM). En particular, se emplea la cartografía generada por la REDIAM sobre las áreas recorridas por los incendios forestales entre 1975 y 2022. Esta contiene los perímetros de incendios forestales mayores de 100 ha en Andalucía obtenidos a partir de imágenes de satélite y datos de campo. Se trata por tanto de una información que no es exhaustiva, pues los incendios con una extensión inferior a 100ha no han sido considerados. Sin embargo, frente a no disponer de otra información operativa de mayor calidad, se utilizará esta teniendo en cuenta que tendrá un efecto sobre las conclusiones que se puedan sacar de los modelos que se construyan.

De esta forma, se han recopilado los polígonos de 1090 incendios forestales ocurridos en Andalucía entre 2002 y 2022, junto con la fecha de inicio de cada uno de ellos (Figura 3.1).

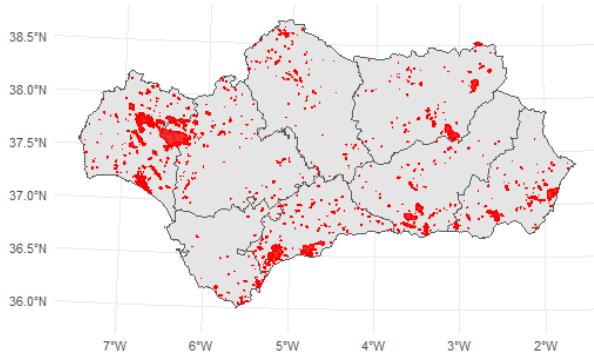


Figura 3.1: Áreas recorridas por el fuego entre 2002 y 2022 en incendios mayores de 100 hectáreas en Andalucía. Fuente: Elaboración propia a partir de las áreas recorridas por el suelo disponibles en la REDIAM.

3.1.2. Variables predictoras

Una vez limitada la extensión territorial del estudio, el siguiente paso era acotar la franja temporal que abarcaría el estudio en base a la disponibilidad de datos adecuados para explicar el fenómeno en cuestión, desagregando espacial y temporalmente.

Los incendios forestales son un proceso sumamente complejo, en el que actúan numerosos factores de muy distinta índole (citar estudio...). Además, dentro de un incendio forestal se pueden distinguir distintas fases que presentan características muy diversas y sobre las que actúan distintos agentes: ignición, propagación y extinción. Dada la información sobre incendios forestales disponible, se está obligado a adoptar un enfoque global, pues no se dispone de los puntos de ignición u origen de los incendios forestales. El enfoque será, por tanto, intentar predecir si una determinada localización se verá afectada por un incendio forestal (de más de 100 ha) en un momento concreto.

Además, es importante tener en cuenta que existen factores estructurales que tienen una influencia directa sobre los regímenes de incendios forestales como son las tendencias de uso y explotación de los bosques, la presencia de interfaz urbano forestal, los tipos y técnicas de agricultura que se llevan a cabo, la presencia e intensidad del pastoreo,

los cambios en los usos de suelo e incluso conductas sociales y tendencias demográficas diversas. Se trata de variables que cambian a lo largo de periodos relativamente largos de tiempo y que muy difícilmente pueden ser incluidos en los modelos, dada la falta de datos sobre ellas, así como su carácter transversal. Por ello, se ha considerado conveniente no extender en exceso el periodo de estudio, reconocida la imposibilidad de incluir en el modelo todas las variables que tienen un impacto relevante en la aparición de incendios y que son cambiantes en el tiempo.

Todo ello hace necesario que el conjunto de datos utilizado contenga información sobre todas las dimensiones (o al menos las principales) que influyen en cualquiera de las fases de un incendio forestal. Es decir, se deben incluir la dimensión antropogénica, la demográfica, la hidrográficas, la topográfica, la meteorológica y la vegetación. Es importante recalcar que siempre se hace referencia a datos geoespaciales pues debe ser la información relativa al lugar (y al momento) del incendio, con la dificultad posterior que esto supondrá.

Por último, es importante diferenciar entre características que se considerarán estructurales (y por tanto invariantes a lo largo del periodo de estudio) y aquellas que se considerarán variables en el tiempo. Dentro de las primeras se encuentran todas las características relacionadas con la topografía del terreno, las infraestructuras y los usos del suelo, como por ejemplo el modelo de elevaciones, la distribución de asentamientos de población, la red de carreteras y el uso de suelo. Todas las demás variables de carácter demográfico, meteorológico o de vegetación se considerarán, por tanto, desagregadas temporalmente.

En base a todo lo mencionado y a la disponibilidad de información de calidad de las categorías comentadas, se ha decidido limitar la franja temporal del estudio a los 20 años que van de 2002 a 2022, ambos inclusive.

3.2. Fuentes de datos

Como se ha comentado en la sección anterior, los datos sobre los incendios forestales se han obtenido de los perímetros de incendios forestales mayores de 100 ha en Andalucía entre 1975 y 2020 disponibles la REDIAM. De cada incendio registrado se dispone de su fecha de inicio y del polígono del área recorrida por el fuego, así como de otras variables que dependen del año de la campaña y que no serán relevantes de cara al presente estudio.

Tomando como base estudios similares (...) y partiendo de las 6 categorías ya mencionadas se han recopilado 23 conjuntos de datos de distinto tipo que se usarán para explicar y predecir los incendios forestales en Andalucía. Estos conjuntos se recogen en la Tabla 3.1, donde también se indica la fuente de la que ha sido obtenido cada uno de ellos, el tipo de datos que contiene (indicando su resolución en el caso de los datos ráster) y la frecuencia de las observaciones (o resolución temporal) en el caso de las variables temporales. En realidad, el número de archivos de datos que se manejan es mucho mayor, ya que, por ejemplo, para la variable *NDVI* se dispone de un archivo *tiff* para cada mes del periodo de estudio, resultando en un total de unos 240 archivos ráster diferentes solo para esta variable. Inevitablemente, esto añade cierta complejidad al manejo de los datos.

Es relevante la heterogeneidad de los datos recopilados, pues se dispone tanto de datos tabulares como de datos espaciales y dentro de estos últimos de datos vectoriales y datos ráster, con distintas resoluciones, distintas frecuencias y distintos sistemas de referencia

Categoría	Datos	Fuente	Tipo de dato	Frecuencia
Topográficas	Altitud	DERA ^a	TIFF (100m)	-
	Orientación	REDIAM ^b	TIFF (100m)	-
	Pendiente	REDIAM	TIFF (100m)	-
	Curvatura	REDIAM	TIFF (100m)	-
Vegetación	NDVI	REDIAM	TIFF (250m)	Mensual
Antropogénicas	Uso de suelo	DERA	Shapefile	-
	Red de carreteras	DERA	Shapefile	-
	Red de ferrocarril	DERA	Shapefile	-
	Línea eléctrica	DERA	Shapefile	-
	Espacio protegido	DERA	Shapefile	-
	Senderos / Vías Verde / Carriles Bici	DERA	Shapefile	-
	Caminos / Vías Pecuarias	DERA	Shapefile	-
Demográficas	Número de habitantes por municipio y año	IECA ^c	csv	Anual
	Extensión municipal	IECA ^c	csv	-
Hidrográficas	Principales Ríos	MAGRAMA ^d	Shapefile	-
Meteorológicas	Precipitación (mm/day)	NASA POWER ^e	df (0.5° x 0.625°)	Diaria
	Temperatura a 2m sobre la superficie (°)	NASA POWER	df (0.5° x 0.625°)	Diaria
	Humedad del suelo (%)	NASA POWER	df (0.5° x 0.625°)	Diaria
	Dirección del viento a 10 metros sobre la superficie terrestre(°)	NASA POWER	df (0.5° x 0.625°)	Diaria
	Humedad relativa a 2m sobre la superficie (%)	NASA POWER	df (0.5° x 0.625°)	Diaria
	Cantidad de precipitaciones (mm/day)	NASA POWER	dfdf (0.5° x 0.625°)	Diaria

Fuente: Elaboración propia

^a Datos Espaciales de Referencia de Andalucía (DERA)

^b Descargas Rediam

^c Instituto de Estadística y Cartografía de Andalucía (IECA)

^d Ministerio de Agricultura, Alimentación y Medio Ambiente (MAGRAMA)

^e NASA Prediction Of Worldwide Energy Resources (NASA POWER)

Tabla 3.1: Conjuntos de datos brutos, tipos de datos y fuentes.

de coordenadas. Esto hará que el procesamiento de estos datos hasta obtener datos adecuados para el análisis estadístico sea costoso y que deban utilizarse técnicas específicas de geocomputación.

Cabe también mencionar que se ha optado por el uso de datos meteorológicos basados en modelos y en observaciones satelitares, en lugar del uso de datos provenientes de estaciones meteorológicas. Si bien la información de estaciones meteorológica puede ser más precisa, la dificultad de disponer de datos consistentes y continuos en el tiempo a lo largo del periodo de estudio de las variables meteorológicas seleccionadas ha hecho que este enfoque no sea viable. En esta dirección se ha explorado la API de la AEMET y algunos paquetes de R como *climate* [5], sin llegar a resultados satisfactorios. Por otro lado, el paquete *nasapower*[17] permite la descarga de una gran cantidad de variables meteorológicas con frecuencia diaria y con una resolución de aproximadamente 0.5×0.625 grados de latitud y longitud (unos 50km). Si bien es cierto que no es lo ideal, es la única opción que se ha considerado viable y, de cara a la construcción de unos primeros modelos approximativos podría ser suficiente. Si quisiese extenderse el estudio, sería conveniente profundizar en la búsqueda de alternativas que permitan obtener información meteorológica de una mayor calidad y detalle.

3.3. Procesamiento de los datos

Una vez se dispone de todos los conjuntos de datos que se usarán en el estudio, el siguiente paso será combinarlos de manera adecuada y transformarlos a un formato tabular apto para el análisis estadístico y la construcción de modelos predictivos. Dado que el objetivo que se persigue es predecir si, dada unas condiciones meteorológicas concretas en

un momento dado, un punto del territorio andaluz se verá afectado o no por un incendio forestal, será necesario disponer de una cantidad suficiente de muestras negativas y positivas distribuidas espacial y temporalmente que tengan asociadas las variables explicativas correspondientes.

Intuitivamente, las muestras positivas serán aquellas observaciones (puntos definidos en el tiempo y en el espacio) dentro del marco espacio-temporal del estudio en las que se ha detectado un incendio forestal en el día de la observación. Es decir, son observaciones dentro de los polígonos de incendios el día que estos se han producido. Por tanto, las muestras negativas serán observaciones dentro del marco espacio-temporal definido en las que no se ha detectado un incendio forestal. Es importante tener en cuenta que dado que solo se dispone de los incendios con una extensión mayor a 100 ha, la muestra cuenta con un importante sesgo, ya que los casos positivos están infrarepresentados. Por ello, no podremos hacer inferencia a todos los incendios forestales, si no solo a los de una extensión superior a 100ha. Un comentario relevante es que al usar información meteorológica basada en modelos, la presencia de un incendio forestal no altera los valores meteorológicos de esa zona.

A continuación se detalla el proceso seguido para generar el conjunto de datos depurado sobre el que desarrollar el estudio a partir de los distintos conjuntos de datos en bruto:

1. Generación de una muestra balanceada de casos positivos y negativos.
2. Asignación de las variables descriptivas a cada observación.
3. Depuración de la muestra.

3.3.1. Generación de una muestra balanceada de casos positivos y negativos.

Para poder construir cualquier modelo de clasificación binaria se necesita disponer de una muestra que cuente con un número suficiente de casos positivos y negativo. Además, es aconsejable trabajar con conjuntos de datos balanceados para evitar sesgos en los modelos de clasificación [21].

Como ya se ha comentado, se considerarán observaciones positivas aquellas que se hayan visto afectadas por un incendio forestal en el día y lugar de la observación. En cambio, serán observaciones negativas aquellas que no se hayan visto afectadas por un incendio forestal en el día y lugar de la observación. Estas observaciones deberán generarse a partir de los polígonos de incendios disponibles. Para ello, se usará un enfoque similar al utilizado en [?], con algunas diferencias importantes. En el estudio citado se usan los puntos de ignición como muestras positivas, con el objetivo de predecir los puntos de origen de los incendios forestales. En cambio, en el presente trabajo no se disponen de los puntos de ignición de los incendios, por lo que el enfoque adoptado es ligeramente diferente; el objetivo es predecir las zonas que pueden verse afectadas por un incendio forestal (superior a 100ha) bajo unas circunstancias concretas. De esta forma, para construir la muestra de casos positivos se han generado 10 puntos aleatorios dentro de cada polígono de incendio y se les ha asignado la fecha del día de inicio del incendio. En ?] la muestra de casos negativos se genera de la siguiente manera: se toman fechas aleatorias dentro del periodo de estudio y a cada una de ellas se le asocia una localización aleatoria dentro del área de estudio satisfaciendo que deben estar a al menos 15km de de cualquier

incendio detectado en un margen de ± 3 días. Esta forma de tomar los casos negativos asegura que estén lo suficientemente alejados de los incendios forestales para representar condiciones no influidas por estos, dando prioridad así a las áreas con una menor prioridad de ocurrencia de incendio en un período definido. En el presente trabajo se utilizarán los mismos parámetros (franja de ± 3 días y distancia superior a 15km), sin embargo, sería conveniente en estudios futuros plantearse si esos parámetros son adecuados o tal vez sería más adecuado tomar otros valores, basados, por ejemplo, en la duración media de los incendios en Andalucía y otras características propias de los incendios en la región.

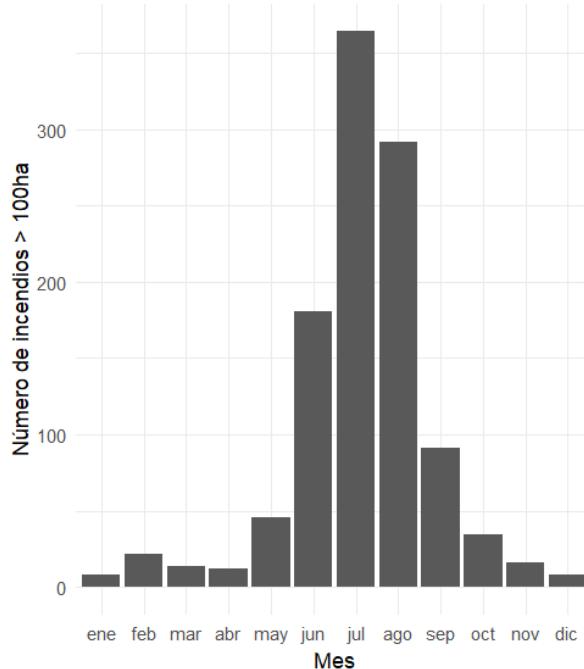


Figura 3.2: Número total de incendios por mes durante el período de estudio. *Fuente: Elaboración propia.*

Tanto en ?] como en otros estudios consultados se generan los casos negativos tomando fechas completamente aleatorias dentro de la franja temporal del estudio. Sin embargo, esto induce un claro sesgo en los datos, ya que las observaciones positivas no se distribuyen uniformemente entre los 12 meses, si no que se concentran marcadamente en los meses de verano, como puede observarse en la Figura 3.2. Generar el conjunto de datos sin tener en cuenta este hecho hace que las muestras positivas y negativas tengan características meteorológicas muy diferenciadas que no responden al verdadero proceso latente de aparición de incendios forestales sino al proceso de selección de la muestra, pudiendo provocar un marcado sesgo positivo en las medidas de evaluación de los modelos que en realidad no estarían reflejando la realidad si no los sesgos introducidos en los conjuntos de datos. Por ello, en el presente trabajo, para generar aleatoriamente los días de las muestras negativas se seguirá una distribución de probabilidad proporcional a la cantidad de incendios observados a lo largo del período de estudio en cada mes. Mediante este enfoque se espera obtener una muestra balanceada y estratificada por mes que permita construir modelos de clasificación capaces de captar los patrones latentes de aparición de incendios forestales. Además, este enfoque permite centrar el esfuerzo computacional en los períodos en los que más incendios se producen.

3.3.2. Asignación de las variables descriptivas a cada observación

Una vez generada una muestra balanceada de 22170 observaciones dentro de Andalucía entre el 1 de enero de 2002 y el 31 de diciembre de 2022 siguiendo el enfoque apenas descrito, el siguiente paso es asignar a cada una de ellas los valores correspondientes a ese día y a esa localización concreta de todas las variables predictoras a partir de los conjuntos de datos que se han recopilado (recogidos en la Tabla 3.1). Dado que será necesario calcular distancias, se usa un Sistema de Referencia de Coordenadas proyectado al generar la muestra. En particular se usa una variante del UTM30N por ser el que traen los archivos *raster* de las variables topográficas (es conveniente, siempre que sea posible, no transformar el crs de los datos *raster* pues al hacerlo se deben interpolar los valores de los píxeles, produciendo pérdida de información).

A continuación se detalla el proceso seguido para manipular construir cada variable:

- Variables topográficas: Simplemente se extrae el dato del píxel en el que cae cada observación para cada uno de los conjuntos de datos.
- Variables antropológicas:
 - Se calculan las distancias de cada observación a la red de carreteras, a la red de ferrocarril, a las poblaciones y a la línea electrica, creando así las variables *dist_carreteras*, *dist_ferrocarril*, *dist_poblaciones* y *dist_electr*, respectivamente. Las geometrías de los senderos, de las vías verdes y de los carriles bici se unen y se calcula la distancia de cada observación a este conjunto de geometrías, generando así la variable *dist_sendero*. De la misma forma se procede con los caminos y las vías pecuarias, que dan origen a la variable *dist_camino*. Estas uniones se han llevado a cabo por considerar que sus elementos tienen características similares. Siempre se hace referencia a la distancia euclídea.
 - Para construir una variable dicotómica *enp* que indique si la observación se encuentra o no dentro de un Espacio Natural Protegido primero se ha rasterizado el conjunto de polígonos de Espacios Naturales Protegidos de Andalucía (de forma que en cada píxel se indica 1 si el centro de este está dentro del polígono de un ENP o 0 si no) y posteriormente se ha extraído el valor del píxel que contiene a cada observación. En la rasterización se ha usado como modelo el ráster de elevaciones con una resolución de $100m \times 100m$. Proceder de este modo hace que se pierda algo de resolución pero resulta muchísimo más eficiente computacionalmente que comprobar para cada observación la relación espacial “estar dentro del polígono de algún ENP” (este enfoque resultaba computacionalmente inviable dado el equipo disponible).
 - Para construir la variable *uso_suelo* se ha procedido de manera similar. Primero se ha rasterizado, usando como modelo el mapa de elevaciones con una resolución de $100m \times 100m$, asignando a cada píxel la categoría de uso de suelo del polígono que cubriese el centro del píxel. Y posteriormente, para cada observación se ha extraído el valor del píxel sobre el que estuviese. De nuevo, se ha procedido así por cuestiones de eficiencia. Es necesario hacer algunos comentarios más sobre esta variable. La información de uso de suelo proviene

Nivel 1	Nivel 2	Código
Superficies artificiales	Zonas urbanas	11
	Zonas industriales, comerciales y de transporte	12
	Zonas de extracción minera, vertederos y de construcción	13
	Zonas verdes artificiales, no agrícolas	14
Zonas agrícolas	Tierras de labor	21
	Cultivos permanentes	22
	Prados y praderas	23
	Zonas agrícolas heterogéneas	24
Zonas forestales con vegetación natural y espacios abiertos	Bosque	31
	Espacios de vegetación arbustiva y/o herbácea	32
	Espacios abiertos con poca o sin vegetación	33
Zonas húmedas	Zonas húmedas continentales	41
	Zonas húmedas litorales	42
Superficies de agua	Aguas continentales	51
	Aguas marinas	52

Tabla 3.2: Códigos de uso de suelo. Fuente: *Elaboración propia a partir de <https://land.copernicus.eu/content/corine-land-cover-nomenclature-guidelines/html/>*

del mapa de Ocupación de Uso de Suelo CORINE Land Cover 2018, donde se establece 3 niveles de clasificación con 5, 15 y 44 clases, respectivamente. La primera clase se corresponde con el primer dígito del código, las segunda con el segundo y la tercera con el tercero. En este trabajo se ha decidido trabajar con el segundo nivel de clasificación, por lo que se consideran solo los dos primeros dígitos del código de cada observación. En la Tabla 3.2 se recoge la clase correspondiente a cada código.

- Para construir la variable *poblacion* se ha asignado a cada observación el código del municipio en el que está y se ha hecho un *left_join* con el código del municipio y el año de la observación. Para la variable *dens_población* se ha procedido de la misma manera pero se ha dividido por la extensión del municipio en km^2 .
- La distancia a los ríos *dist_rios* se ha obtenido simplemente calculando la distancia de cada observación al conjunto de geometrías de los principales ríos de España.
- El NDVI viene en archivos *raster* mensuales (lo que supone un total de unos 240 archivos en formato *.tiff*). Para cada observación se ha extraído el valor del píxel correspondiente (en función de las coordenadas del punto) del archivo correspondiente (que depende del mes y año de la observación).

3.3.3. Depuración de la muestra

Una vez construido el conjunto de datos “en bruto”, se tratan los valores perdidos y se ajustan adecuadamente los tipos de las variables. En primer lugar se convierten en factores las variables *fire*, *enp* y *uso_suelo*. A continuación, se codifican las variables numéricas *WD10M* y *orientacion* en los 4 puntos cardinales y sus bisectrices, generando así 8 clases (“N”, “NW”, “W”, “SW”, “S”, “SE”, “E”, “NE”). En el caso de la variable orientación se añade también la clase “plano”, si la pendiente en ese punto es 0.

El conjunto de datos construido tiene 200 registros incompletos, lo cual supone un 0.1 % del total de registros. De estos, el 68 % son casos negativos y el 32 % son casos positivos.

Los valores perdidos se encuentran en las variables demográficas (85), en *uso_suelo* (8), en NDVI (85) y en las variables topográficas (53). Las causas de los datos faltantes son:

- El dato no está disponible para esa observación. Esto sucede con las variables demográficas (hay años para los que no está disponible el número de habitantes de algunos municipios) y el NDVI (para algunos meses no se dispone del archivo correspondiente).
- En el caso de las variables topográficas los valores perdidos se encuentran todos en los límites de la comunidad (Figura 3.3). Al proceder de datos en formato *raster*, los píxeles con información no se ajustan exactamente a los límites de Andalucía (ya que son cuadrados). Esto provoca que para algunos puntos situados en los bordes del polígono no esté disponible la información de las variables topográficas.

Tras explorar otras alternativas y teniendo en cuenta tanto el reducido número de registros incompletos como la naturaleza de los valores desconocidos, se opta simplemente por eliminar estos registros.

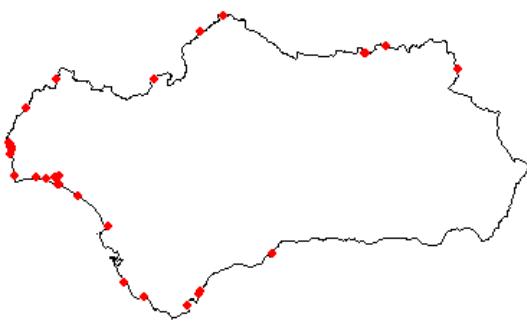


Figura 3.3: Observaciones para los que no está disponible alguna de las variables topográficas. *Fuente: Elaboración propia.*

El resultado de todo este proceso es un conjunto de datos con 21546 registros y 27 variables, las cuales se detallan en la Tabla 3.3.

Categoría	Nombre	Descripción	Tipo
Topográficas	elevacion	Elevación sobre el nivel del mar (m)	numérica
	orientacion	Orientación de la pendiente descendiente	categórica
	pendiente	Pendiente del terreno (°)	numérica
	curvatura	Curvatura de la superficie	numérica
Vegetación	NDVI	Índice de vegetación de diferencia normalizada	numérica
Antropogénicas	uso_suelo	Clasificación del uso del suelo	categórica
	dist_carretera	Distancia a la carretera más cercana (m)	numérica
	dist_ferrocarril	Distancia a la vía de ferrocarril más cercana (m)	numérica
	dist_electr	Distancia a la línea electrica más cercana (m)	numérica
	enp	Espacio Natural Protegido	categórica
	dist_sendero	Distancia a la vía verde, al carril bici o al sendero más cercano (m)	numérica
	dist_camino	Distancia al camino o a la vía pecuaria más cercano (m)	numérica
Demográficas	poblacion	Número de habitantes del municipio	numérica
Hidrográficas	dens_poblacion	Densidad de población del municipio	numérica
Meteorológicas	dist_rios	Distancia al río más próximo (m)	numérica
	PRECTRORCORR	Promedio corregido del total de precipitaciones en la superficie	numérica
	T2M	Temperatura promedio del aire a 2 metros sobre la superficie de la tierra (°C)	numérica
	GWETTOP	Porcentaje de humedad del suelo	numérica
	WD10M	Promedio de la dirección del viento a 10 metros sobre la superficie de la tierra	categórica
	WS10M	Promedio de la velocidad del viento a 10 metros sobre la superficie de la tierra (m/s)	numérica
	RH2M	Humedad relativa a 2 metros sobre la superficie de la tierra	numérica
Variable Objetivo	fire	Incendio forestal	categórica
Identificadoras	date	Fecha de la observación	fecha
	municipio	Nombre del municipio	texto
	cod_municipio	Código del municipio	texto
	geometry	Geometría de los puntos	sfc

Tabla 3.3: Variables del conjunto de datos depurados. *Fuente: Elaboración propia.*

Capítulo 4

Cuerpo

4.1. Análisis exploratorio de datos

En esta sección se aplicarán distintos métodos numéricos y gráficos de análisis de datos a la muestra generada siguiendo el procedimiento detallado en el capítulo anterior. Se usarán principalmente técnicas de estadística descriptiva para comprender las características del conjunto de datos y extraer conocimiento útil para el problema que se intenta abordar, predecir incendios forestales. Es importante tener presente que se trata de datos correlados espacial y temporalmente, lo que hace necesario el uso de métodos específicos para este tipo de datos. Los objetivos de esta etapa son:

1. Generar conocimiento sobre el conjunto de datos que nos permita evaluar la calidad de este, sin olvidar las limitaciones que ya se han comentado en la sección anterior.
2. Conocer, al menos de forma descriptiva, el impacto de cada variable en la variable objetivo. Este conocimiento será necesario para evaluar e interpretar los modelos que se construirán en la próxima sección.
3. Analizar las características de las distintas variables, de cara a usar posteriormente técnicas de preprocesamiento adecuadas para cada modelo.

Antes de abordar el estudio detallado de cada una de las variables y las relaciones entre estas, en la Figura 4.1 se recoge un resumen de todo el conjunto de datos, sin incluir la columna de geometría. En este resumen se puede observar que en el conjunto de datos hay 4 tipos de variables (además de la variable *geometry* que es de tipo *simple feature column POINT*, abreviado como *sfc_POINT*): cadenas de caracteres, fechas, factores y variables numéricas.

Se puede observar que hay registros en 749 municipios diferentes (de los 785 municipios de que hay en Andalucía). Probablemente el hecho de que en algunos municipios no haya habido observaciones sea debido a los datos faltantes. Las variables *municipio* y *cod_municipio* no se incorporarán a los modelos. De la misma forma, se puede ver que hay observaciones en 3691 días diferentes. El conjunto cuenta con 5 variables de tipo factor: *fire* (la variable objetivo), *WD10M*, *orientacion*, *enp* y *uso_suelo*; y con 18 variables numéricas. Aunque cada una de ellas se analizará a continuación con detalle, ya cabe hacer algunos comentarios:

- El 38 % de las observaciones se encuentran en espacios de vegetación arbustiva y/o herbácea (código 32).
- Como era de esperar, por la forma en la que se ha tomado la muestra, el conjunto está balanceado.
- El 81 % de las observaciones se encuentran fuera de Espacios Naturales Protegidos.
- Todas las variables, salvo *T2M* y *curvatura*, son positivas y la mayoría de ellas presentan una marcada distribución asimétrica hacia la derecha.
- Las variables muestran escalas muy diversas entre ellas, siendo *GWETTOP* la que presenta menor desviación típica (0.145) y *poblacion* la que tiene una desviación típica mayor (64453). Se evidencia la necesidad de incluir algún método de normalización de las variables en el preprocesamiento de los datos.

— Data Summary —										
	Values									
Name	datos									
Number of rows	21546									
Number of columns	26									
Column type frequency:										
character	2									
Date	1									
factor	5									
numeric	18									
Group variables										
	None									
— Variable type: character —										
	skim_variable n_missing complete_rate min max empty n_unique whitespace									
1 cod_municipio	0	1	5	5	0	749	0			
2 municipio	0	1	3	32	0	749	0			
— Variable type: Date —										
	skim_variable n_missing complete_rate min max median n_unique									
1 date	0	1	2002-01-02	2022-11-29	2012-08-04		3691			
— Variable type: factor —										
	skim_variable n_missing complete_rate ordered n_unique top_counts									
1 fire	0	1	FALSE	2	1: 10794, 0: 10752					
2 WD10M	0	1	FALSE	8	SW: 4965, W: 4867, S: 3786, SE: 3316					
3 orientacion	0	1	FALSE	9	S: 3267, SW: 3090, SE: 2956, W: 2544					
4 emp	0	1	FALSE	2	0: 17393, 1: 4153					
5 uso_suelo	0	1	FALSE	15	32: 8068, 21: 3128, 24: 2798, 22: 2786					
— Variable type: numeric —										
	skim_variable n_missing complete_rate mean sd p0 p25 p50 p75 p100 hist									
1 T2M	0	1	24.6	5.20	-0.8	22.7	25.6	28.0	36.0	
2 GWETTOP	0	1	0.318	0.145	0.09	0.2	0.29	0.42	0.91	
3 RH2M	0	1	46.4	16.1	9.23	33.8	44.6	57.5	96.8	
4 WS10M	0	1	3.64	1.42	0.99	2.67	3.34	4.25	13.3	
5 PRECTOTCORR	0	1	0.229	1.38	0	0	0	0	46.1	
6 elevacion	0	1	494.	400.	0	169.	425.	711.	3351.	
7 pendiente	0	1	12.3	11.8	0	3.46	8.24	17.9	98.0	
8 curvatura	0	1	0.00549	0.0563	-0.294	-0.0193	-0.000100	0.0255	0.420	
9 dist_carretera	0	1	1799.	1907.	0.0706	507.	1200.	2425.	17662.	
10 dist_poblacion	0	1	902.	800.	0	379.	713.	1170.	8215.	
11 dist_electric	0	1	5253.	5640.	0.220	1216.	3462.	7276.	48069.	
12 dist_ferrocarril	0	1	15311.	13521.	0.824	5086.	11855.	21467.	85242.	
13 dist_camino	0	1	762.	741.	0.0238	240.	539.	1060.	7090.	
14 dist_sendero	0	1	5619.	4805.	0.0465	1781.	4361.	8276.	25103.	
15 poblacion	0	1	22095.	64453.	114	2252	4860	16759	704414.	
16 dens_poblacion	0	1	105.	320.	2.30	12.2	30.8	71.1	4974.	
17 dist_rios	0	1	6781.	5836.	1.94	2162.	5237.	10123.	37074.	
18 NDVI	0	1	0.412	0.136	0	0.314	0.393	0.495	0.944	

Figura 4.1: Resumen numérico del conjunto de datos depurados. Fuente: *Elaboración propia*.

4.1.1. Distribución de la variable objetivo

En primer lugar, se estudiará la distribución de la *fire* espacial y temporalmente.

En la Figura 4.2 se muestran los histogramas de la variable objetivo en función del día de la semana, del mes y del año, respectivamente. En primero de ellos se observa que mientras que la distribución de los casos negativos es uniforme entre los días de la semana,

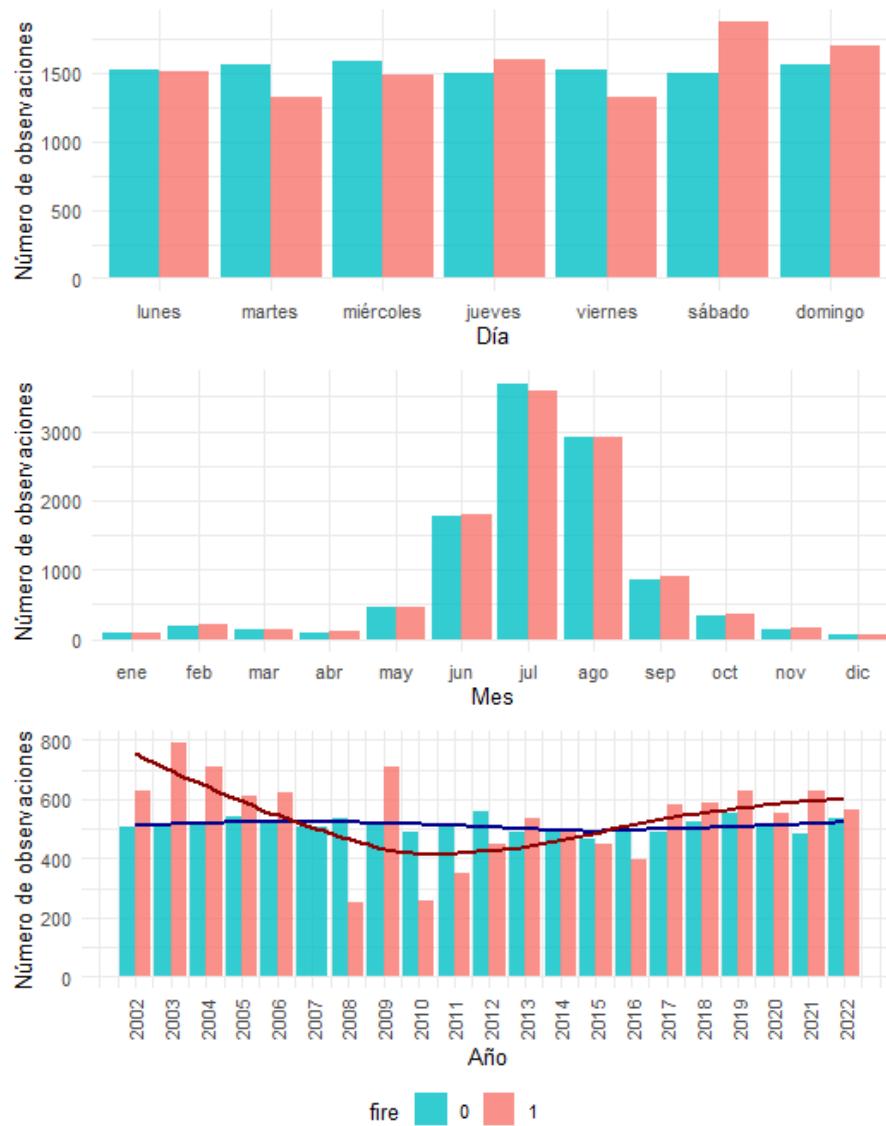


Figura 4.2: Distribución temporal de la variable objetivo. *Fuente: Elaboración propia.*

en los casos positivos se aprecia un ligero aumento en el fin de semana, especialmente en el sábado. En el segundo histograma, se observa como las observaciones se concentran en los meses de verano y en cada mes hay una cantidad balanceada de muestras de ambas clases (esto es fruto del proceso de muestreo de las observaciones negativas, que como se ha explicado en la sección anterior, se ha llevado acabo asegurando que la proporción de casos negativos en cada mes sea igual a la de los casos positivos). En el tercer histograma es remarcable que, mientras las observaciones negativas están uniformemente distribuidas entre los 20 años del estudio, las positivas muestran una disminución importante en los años 2008 y 2010. En el año 2007 no hay observaciones positivas, debido a que los 4 polígonos de incendios mayores de 100ha que había registrados ese año no disponían de la fecha de inicio del incendio, por lo que no pudieron usarse para el estudio. Se desconoce la causa del reducido número de incendios (mayores de 100ha) en 2007, 2008 y 2010.

Dada la clara influencia el mes y la aparente influencia del día de la semana en la aparición de incendios, estas variables serán incluidas en los modelos a través del procesamiento de la variable *date*.

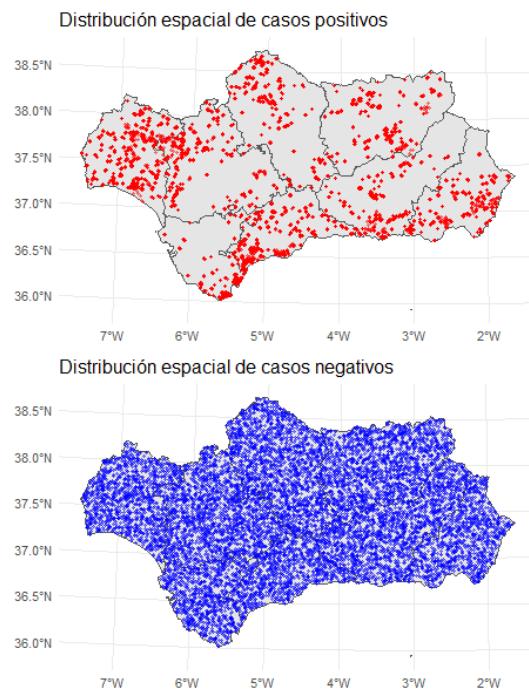


Figura 4.3: Distribución espacial de la variable objetivo. *Fuente: Elaboración propia.*

En la Figura 4.3 se observa claramente como las 10752 muestras negativas están uniformemente distribuidas dentro de los límites de la Comunidad Autónoma de Andalucía, mientras que las 10794 muestras positivas se concentran a ambos lados de la cuenca del río Guadalquivir, con una mayor densidad de observaciones en la provincia de Huelva y en algunas zonas de la costa mediterránea (como ya se apreciaba en la Figura 3.1).

4.1.2. Análisis univariantes variables numéricas

El análisis univariante de las variables numéricas se lleva a cabo desde 3 enfoques complementarios:

1. A través de los resúmenes numéricos recogidos en la Figura 4.1 y del análisis gráfico de los diagramas de caja y bigote (4.4).
2. Estudiando la media mensual de cada variable en función de la variable *fire*.
3. Analizando la distribución espacial de cada variable separando por mes si corresponde.

En los *boxplots* de las variables numéricas en función de la variable *fire* (4.4) destacan varios aspectos. Por un lado, como ya se había comentado anteriormente, que las variables presentan escalas muy diferentes y que la mayoría de las variables tienen una marcada asimetría hacia la derecha. Por otro lado, es evidente la gran cantidad de valores *outliers* que se observan en los datos, lo que tendrá implicaciones en los modelos que se construyan con ellos. Sin embargo, es importante destacar que no se trata de observaciones erróneas, si no que son inherentes a la naturaleza de los datos. Por ejemplo, en el caso de la variable *PRECTOTCORR* el valor máximo observado es 46.06mm en un día, un valor

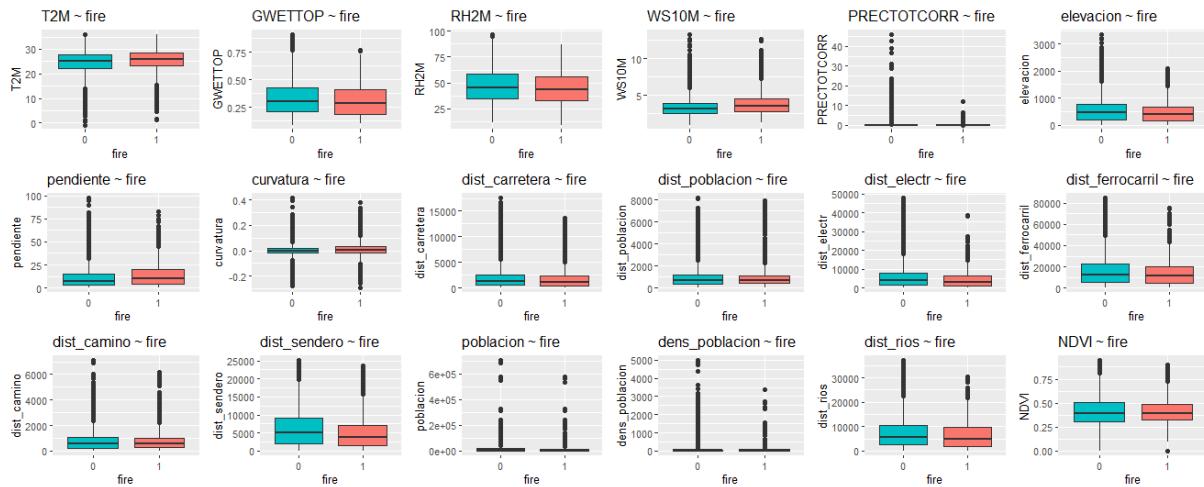


Figura 4.4: Boxplot de cada variable numérica en función de la variable objetivo. *Fuente: Elaboración propia.*

elevado que sin duda es atípico en esta región de clima seco, pero sin embargo, posible. Es también remarcable que todas las variables presentan una variabilidad similar en ambos niveles del factor *fire*, lo que indica que no será un problema de clasificación trivial. A priori, solo con los diagramas de caja y bigotes y los resúmenes numéricos es difícil llegar sacar más conclusiones, sin embargo, sí pueden observarse sutiles diferencias entre las distribuciones de algunas variables para ambos niveles del factor *fire*.

Dada la naturaleza temporal de algunas variables, el análisis gráfico de los *boxplots* resulta insuficiente. Con el fin de considerar la componente estacional de las variables climáticas y de vegetación, a continuación, se estudiará la media mensual de cada una de estas variables función de la variable objetivo.

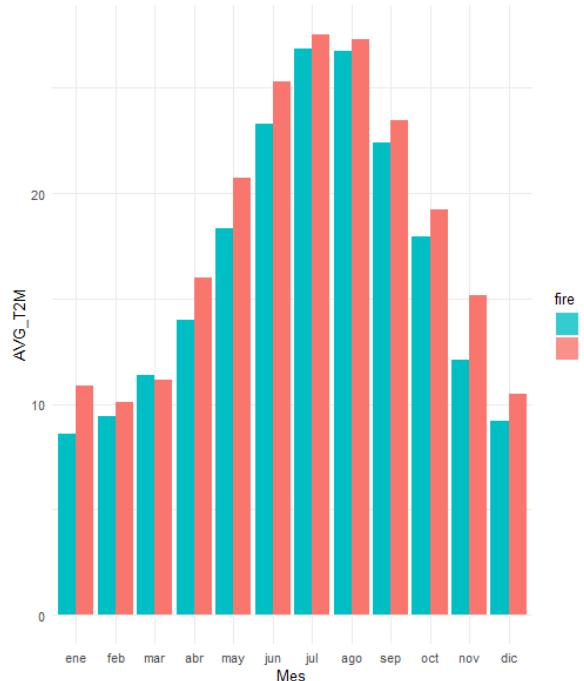


Figura 4.5: Media mensual de *T2M* en función de *fire*. *Fuente: Elaboración propia.*

En la Figura 4.5 se puede observar como en casi todos los meses, la temperatura media mensual es superior en las observaciones en las que se ha registrado un incendio forestal.

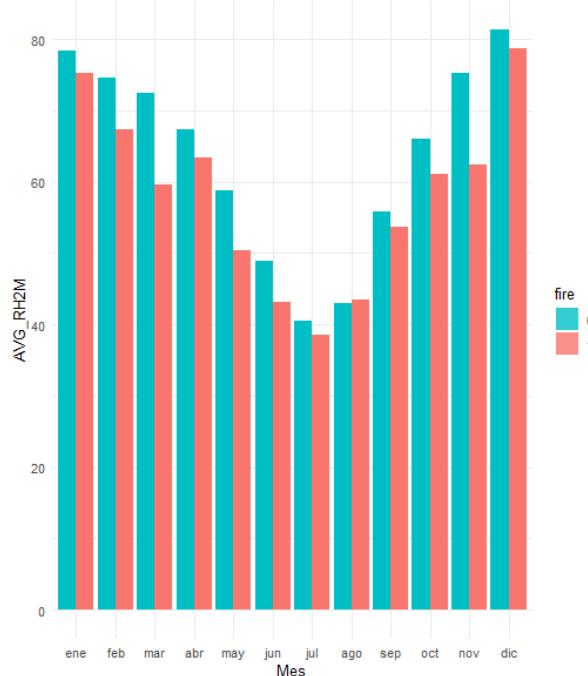


Figura 4.6: Media mensual de $RH2M$ en función de $fire$. Fuente: Elaboración propia.

En la Figura 4.6 se puede observar que en todos los meses la media mensual de la humedad relativa del aire a 2m sobre la superficie es menor en las observaciones en las que se ha registrado un incendio forestal. Sin embargo, las diferencias se reducen durante los meses de verano, en los que la humedad presenta valores bajos en ambas clases.

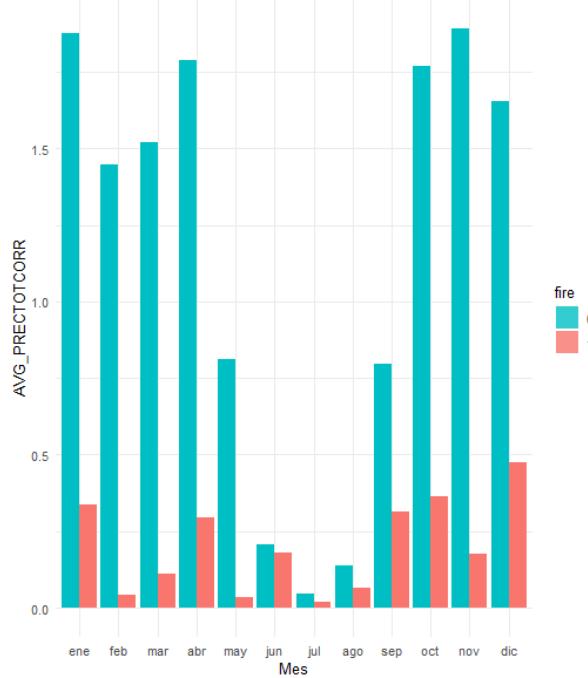


Figura 4.7: Media mensual de $PRECTOTCORR$ en función de $fire$. Fuente: Elaboración propia.

En la Figura 4.7 se observa una clara diferencia en la media mensual de las precipitaciones diarias en función del de si se ha registrado o no un incendio forestal en la observación, siendo significativamente mayor en este último caso.

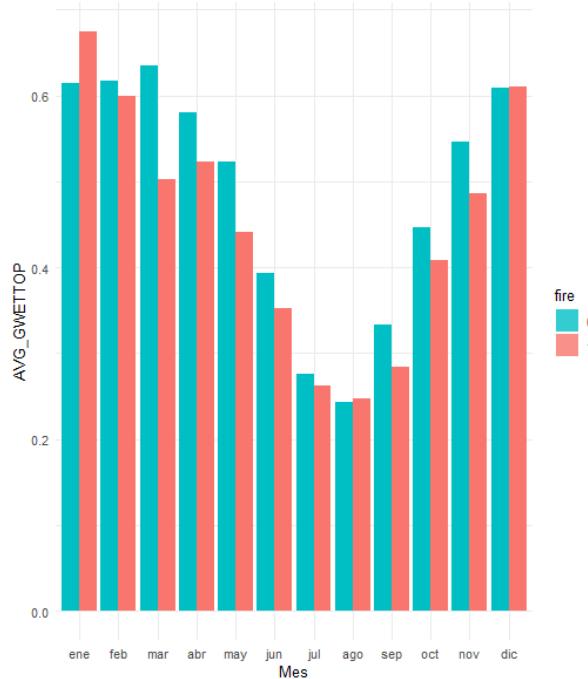


Figura 4.8: Media mensual de $GWETTOP$ en función de $fire$. Fuente: Elaboración propia.

En la Figura 4.8 se observa un gráfico similar al de la humedad relativa del aire, con

valores medios más elevados en las observaciones en las que no se han registrado incendio forestal. Sin embargo, también parece que las diferencias son más reducidas durante la estación estival.

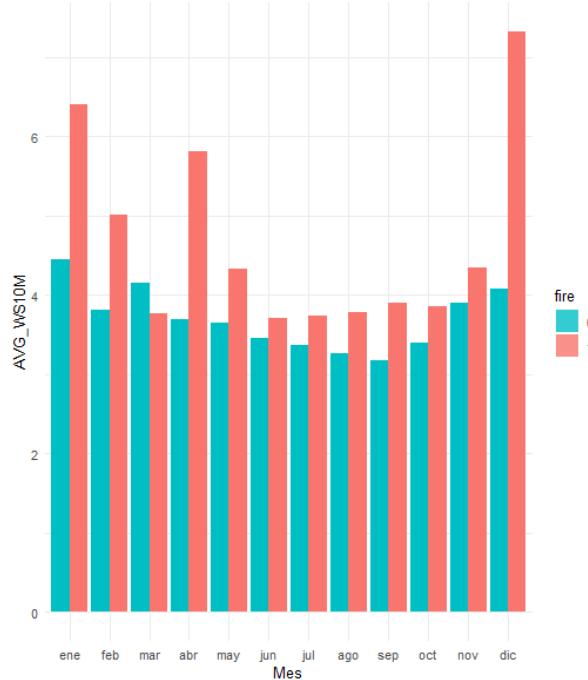


Figura 4.9: Media mensual de $WS10M$ en función de $fire$. Fuente: Elaboración propia.

En la Figura 4.9 se observa como durante todos los meses, la media mensual de la velocidad del viento a 10 metros sobre la superficie es mayor en los registros en los que ha habido un incendio forestal.

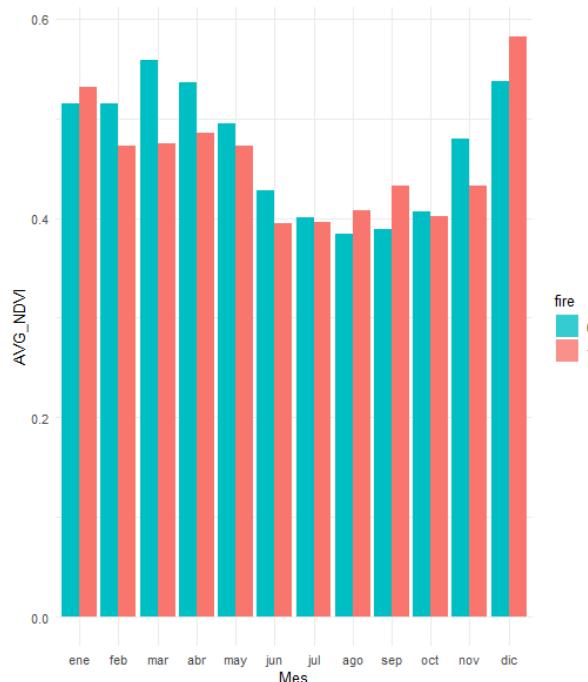


Figura 4.10: Media mensual de $NDVI$ en función de $fire$. Fuente: Elaboración propia.

Como se observa en la Figura 4.10, las diferencias entre los casos en los que se ha registrado incendio y los que no en términos del *NDVI* no están claras.

En el [Apéndice: Gráficos espaciales EDA] se recogen los gráficos espaciales y espacio_temporales de todas las variables numéricas. En ellos se refleja como la valores de las variables en estudio son coherentes con lo que cabría esperar de la realidad. Además, permiten una comprensión mayor de la distribución espacial (y temporal) de las variables en el área de estudio, lo que será útil de cara a interpretar los modelos que se construyan.

4.1.3. Análisis multivariantes de las variables numéricas

En la Figura 4.11 se muestra un gráfico con las correlaciones entre las variables. La interpretación es sencilla, cuanto más intenso sea el color y cuanto mayor sea la excentricidad de la elipse, mayor será la correlación (en valor absoluto) para ese par de variables. El color de la elipse indica el signo del coeficiente de correlación. De esta forma, se observa que las variables más correlacionadas en la muestra son: *T2M* con *RH2M* (negativamente, -0.71), *T2M* con *GWETTOP* (negativamente, -0.69), *GWETTOP* con *R2HM* (positivamente, 0.68) y *poblacion* con *dens_poblacion* (positivamente, 0.63).

En la Figura 4.12 se muestra el gráfico de coordenadas paralelas de las variables tipificadas a una normal estándar, es decir, restándoles la media y dividiendo por la desviación típica. Este gráfico complementa la información de los *boxplots*, pues refleja también las relaciones entre las variables. Si bien es cierto que al tener un número bastante elevado de observaciones el gráfico no es tan claro, pueden hacerse algunas observaciones importantes.

En primer lugar, se observa que la variable con mayor variabilidad (una vez tipificada) es *PRECTOTCORR*, que presenta bastantes valores atípicos, todos ellos en observaciones en las que no se ha registrado incendio. También destacan en este sentido *dens_poblacion* y *poblacion*, entre las que además puede observarse que no hay una relación lineal clara (hay municipios con un elevado número de habitantes pero con una densidad de población reducida y viceversa). Además, puede verse que todas las variables tienen una marcada asimetría positiva (salvo *curvatura*, *T2M* y *NDVI*). Este gráfico es útil también pues permite ver a qué clase de *fire* corresponden los valores más atípicos de cada variable. Por ejemplo: la mayor parte de los valores más elevados de *WS10M*, *dist_poblacion*, *curvatura* y *dist_camino* se dan en observaciones positivas, mientras que en *PRECTOTCORR*, *elevacion*, *GWETTOP*, *dist_Carretera*, *dist_electr* y *dist_rios* sucede lo contrario.

Los resultados de aplicar análisis de componentes principales sobre la matriz de correlaciones de las 18 variables numéricas se muestran en la Figura 4.13. Como se puede observar, se necesitan al menos 11 componentes principales para lograr explicar el 80 % de la varianza de la muestra, y 14 para alcanzar el 90 % de la varianza de los datos. Estos resultados se aplicarán más adelante en los modelos, pero a nivel meramente explicativo ya indican que se trata de un conjunto de datos complejo en cuanto a la dimensión real de estos.

4.1.4. Análisis de las variables categóricas

Las variables categóricas se analizarán a través de los histogramas de cada variable en función de la variable *fire* (Figura 4.14).

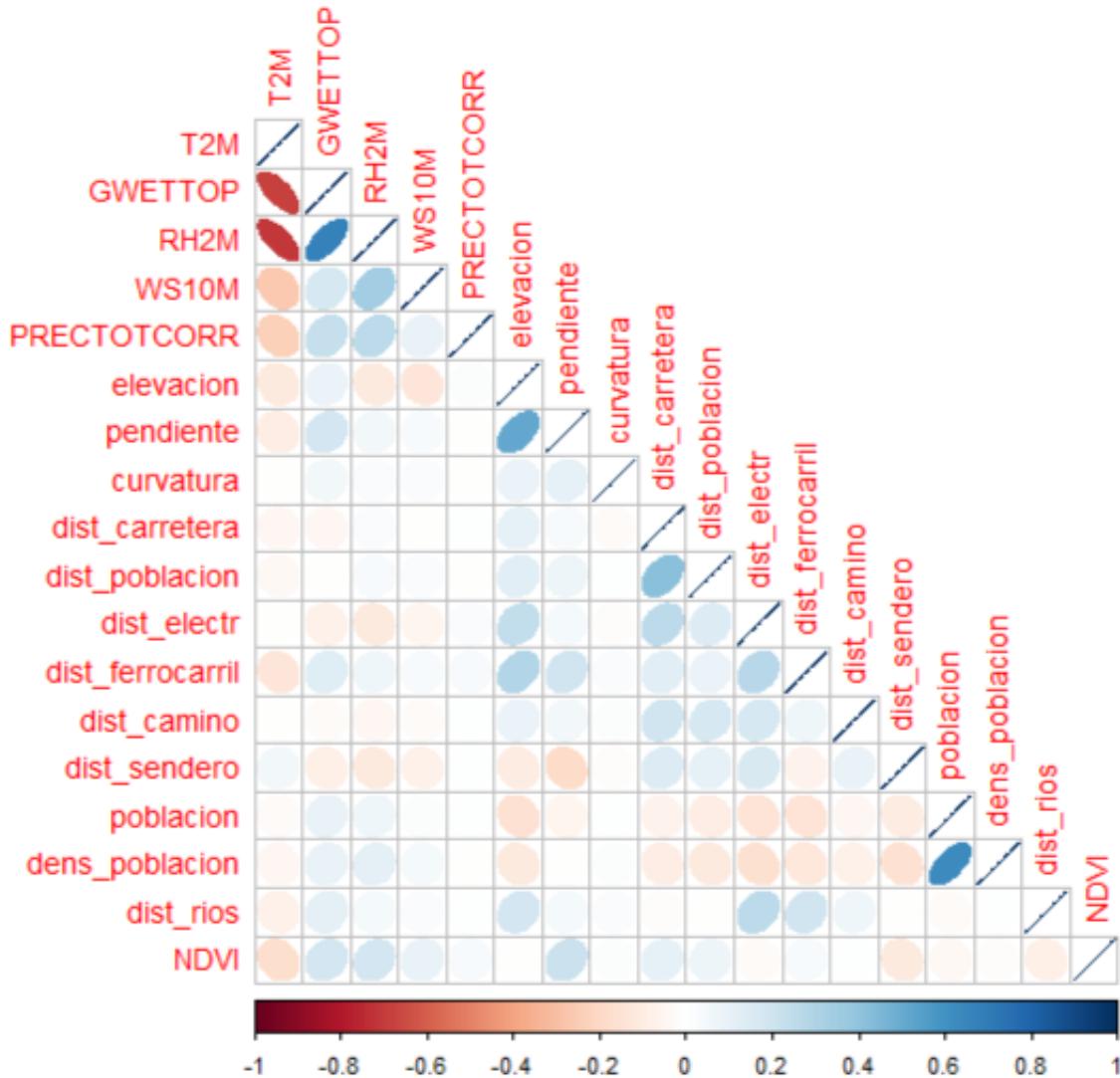


Figura 4.11: Correlaciones entre variables numéricas. Fuente: Elaboración propia.

En la variable *WD10M* cabe destacar la escasez de observaciones con dirección del viento norte. En el histograma no se observa una clara relación de esta variable con la variable objetivo, aunque entre las observaciones con viento con dirección sur o suroeste hay más observaciones negativas y entre las que tienen dirección noroeste o este hay una mayor presencia de observaciones positivas.

En el caso de la variable *orientación*, la relación tampoco está clara, aunque puede verse una mayor proporción de observaciones positivas en las superficies con orientación sur (sureste, sur y suroeste).

En términos de la variable *enp* por si sola no se observan diferencias significativas entre ambas clases.

La variable *uso_suelo* sí que muestra una distribución marcadamente diferenciada entre ambas clases. La mayoría de las observaciones positivas se dan en espacios de vegetación arbustiva y/o herbácea, clase en la que hay casi el doble de observaciones positivas que negativas. En tierras de labor y cultivos permanentes la proporción de observaciones

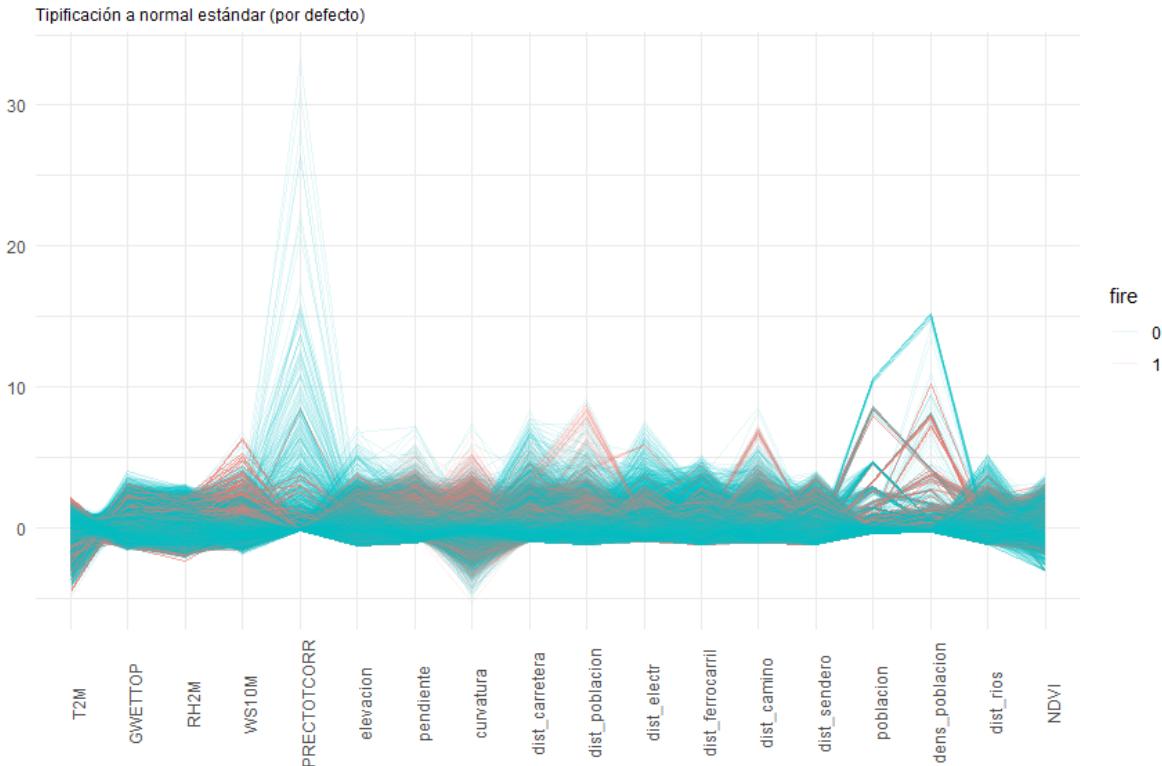


Figura 4.12: Gráfico de coordenadas paralelas de las variables numéricas tipificadas. *Fuente: Elaboración propia.*

Importance of components:										
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10
Standard deviation	1.6830901	1.5509109	1.26704790	1.18114361	1.13860921	1.00151777	0.98498153	0.92948032	0.92853341	0.91036573
Proportion of Variance	0.1573773	0.1336291	0.08918946	0.07750557	0.07202394	0.05572433	0.05389937	0.04799631	0.04789857	0.04604254
Cumulative Proportion	0.1573773	0.2910065	0.38019595	0.45770152	0.52972546	0.58544979	0.63934916	0.68734547	0.73524404	0.82314912
	Comp.11	Comp.12	Comp.13	Comp.14	Comp.15	Comp.16	Comp.17	Comp.18	Comp.19	Comp.20
Standard deviation	0.86805852	0.85739620	0.78965012	0.72570700	0.65041713	0.60872397	0.52343451	0.48001944	0.40486253	0.3464152
Proportion of Variance	0.04604254	0.04186253	0.04084046	0.03464152	0.02925837	0.02350236	0.02058583	0.01522132	0.01280104	0.02314912
Cumulative Proportion	0.78128658	0.82314912	0.86398958	0.89863109	0.92788946	0.95139182	0.97197765	0.98719896	1.00000000	1.00000000

Figura 4.13: PCA sobre la matriz de correlaciones de las variables numéricas. *Fuente: Elaboración propia.*

negativas es mucho mayor, mientras que en zonas agrícolas heterogéneas y en espacios abiertos con poca o sin vegetación hay una mayor presencia de observaciones positivas. También es relevante el hecho de que casi la totalidad de las observaciones se encuentran en zonas agrícolas y en zonas forestales, mientras que en las demás clases la proporción de observaciones es mucho menor (3.5 % de total). Es por ello que antes de construir los modelos, todas las categorías de uso de suelo que no se corresponden con zonas agrícolas o forestales (es decir, todas cuyo código no comienza por 2 o 3) se agruparán en el nivel *Otro*. En A.19 puede observarse la distribución espacial de esta variable.

4.2. Modelización

A continuación se va a utilizar el conjunto de datos construido en el capítulo Construcción del conjunto de datos para entrenar los modelos de clasificación binaria explicados en la sección Modelos. Es evidente que el rendimiento de los modelos debe evaluarse en

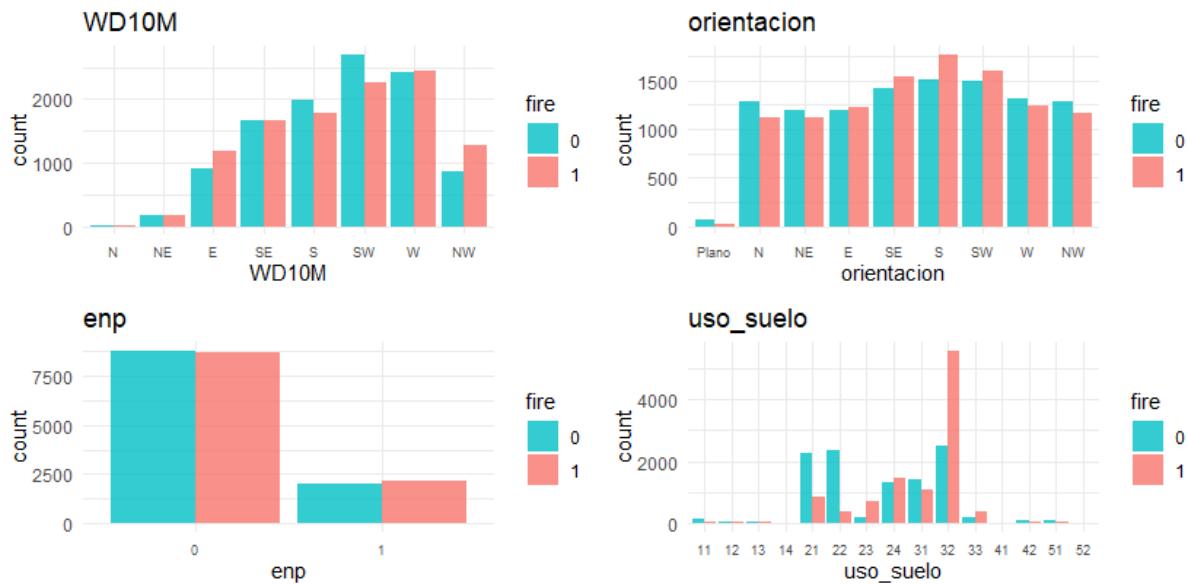


Figura 4.14: Histogramas de las variables categóricas en función de *fire*. Fuente: Elaboración propia.

observaciones futuras, por lo que las técnicas habituales de validación cruzada o partición aleatoria en entrenamiento/test no son adecuadas para este problema, ya que sufrirían el llamado efecto *look-ahead*. Por tanto, el enfoque que se seguirá en este trabajo será trabajar con una partición en entrenamiento/validación/test construida a partir de la ordenación temporal de las observaciones.

Se compararán los resultados obtenidos en 7 modelos diferentes: Regresión logística con penalización, Regresión logística con penalización + PCA, Árboles de decisión, Bosques aleatorios, KNN, SVM lineal y SVM radial.

Se ha seguido el flujo de trabajo habitual de *tidymodels*:

1º. Crear una partición temporal en entrenamiento (60 %), validación (20 %) y test (20 %), que será utilizada en todos los modelos.

2º. Definir cada uno de los modelos, indicando los parámetros del modelo deberán ajustarse.

3º. Crear la receta (*recipe*) con el preprocessamiento que se usará en cada modelo. Como se observó en la sección Análisis exploratorio de datos, se incluirán en todos los modelos variables categóricas que indiquen el día de la semana y el mes de cada observación, haciendo uso de la función `step_date`. Igualmente, como también se indicó en el EDA, se modificará la variable *uso_suelo* para unificar todos los niveles que no sean agrícolas o forestales en un solo nivel que se llamará *Otro*. Esto último se hará fuera del *workflow*, antes de realizar la partición del conjunto de datos, haciendo uso de la función `fct_lump` del paquete *forcats*. Los detalles del preprocessamiento que se ha llevado a cabo en cada modelo pueden consultarse en el código, que se adjunta en el Apéndice 2, donde aparecen debidamente comentados. 4º. Crear el *workflow* con el modelo y la receta.

5º. Crear la rejilla (*grid*) con los posibles valores de los parámetros que se deben ajustar.

6º. Entrenar el modelo para cada combinación de los valores de los parámetros a ajustar sobre los datos de entrenamiento.

7º. Evaluar el rendimiento de cada modelo sobre los datos de validación y seleccionar el mejor en base a las medidas de rendimiento ya mencionadas. El objetivo con estos modelos es predecir incendios forestales, por lo que es de vital importancia que los modelos funcionen especialmente bien en la clase positiva, es decir, que si un incendio se va a producir, que el modelo lo detecte. Sin embargo, es fundamental que el modelo tenga un buen desempeño general (un modelo que todo lo clasifique como incendio no serviría de nada, poniendo un ejemplo extremo). Por tanto, cada modelo se valorará de forma individual, considerando todas las métricas de rendimiento mencionadas y priorizando la sensitividad (o *recall*). Sin embargo, en la mayoría de los casos maximizar la tasa de acierto maximiza también la sensitividad, garantizando además un buen desempeño general. Por ello, en la mayoría de modelos se maximizará la tasa de acierto, pero porque analizando las salidas individualmente se ha considerado que es la mejor opción ya que produce la mayor sensitividad sin bajar demasiado las otras medias.

4.2.1. Regresión logística con penalización

Antes de aplicar este modelo, se han transformado las variables categóricas a variables *dummy* y se han tipificado todas las variables (media 0 y varianza 1). Los parámetros a ajustar son λ (parámetro de penalización o *penalty*) y α (parámetro de mixtura o *mixture*). Se consideran 10 valores equiespaciados para cada parámetro (en el caso de λ entre 10^{-4} y 10^{-1} y el caso de α entre 0 y 1) y se construye el grid tomando todas las combinaciones de estos valores.

Las métricas obtenidas por cada combinación de parámetro sobre los datos de validación se representan en la Figura 4.15

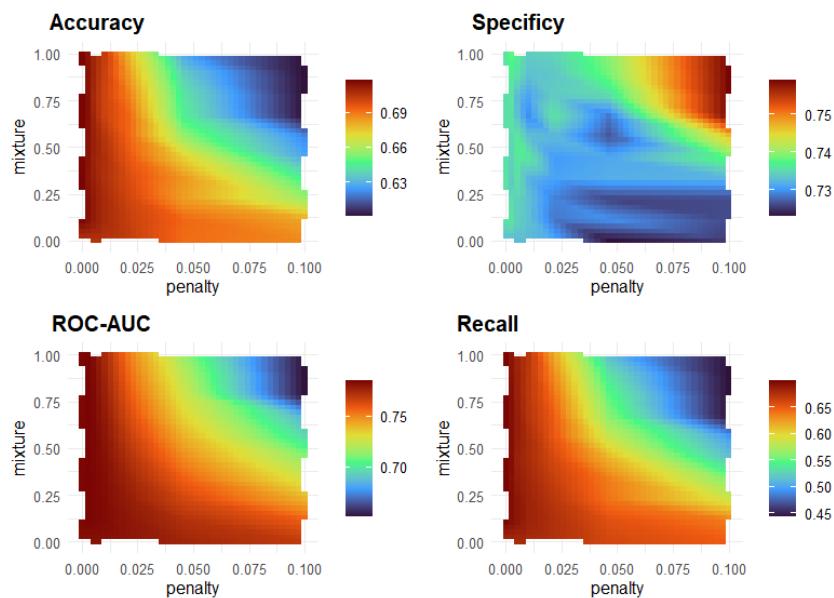


Figura 4.15: Métricas de rendimiento de los modelos de regresión logística con penalización. *Fuente: Elaboración propia.*

Finalmente, se elige el modelo que maximiza la tasa de acierto, cuyos parámetros son: $\alpha = 1$ y $\lambda = 0.000464$. Es decir, un modelo de regresión logística *lasso* puro. Los coeficientes de este modelo se muestran en la Figura A.20.

4.2.2. Regresión logística con penalización + PCA

A continuación se considera el mismo modelo de regresión logística con penalización que en la sección anterior pero en lugar de trabajar con los datos directamente, se aplica análisis de componentes principales sobre los datos normalizados en el preprocesamiento, ajustando el número de componentes principales utilizadas. Para construir el *grid* de parámetros se consideran los mismos valores que en el modelo sin PCA para los parámetros de penalización y mixtura pero ahora se consideran también 7 posibles valores para el número de componentes principales ($\{20, 25, 30, \dots, 50\}$). Finalmente, el modelo que maximiza la tasa de acierto es el que tiene 40 componentes principales, $\alpha = 0.333$ y $\lambda = 0.00464$.

4.2.3. Árboles de decisión

Se construirán los árboles de decisión usando el índice de Gini como función de impureza y se elegirá el parámetro de coste-complejidad (α) que maximice la tasa de acierto. Se considera un *grid* con 10 valores del parámetro de coste-complejidad que oscilan entre $1.28e - 10$ y $3.02e - 2$. La mejor tasa de acierto en el conjunto de validación se obtiene con $\alpha = 0.00182$. En la Figura 4.16 se muestran las distintas métricas de rendimiento sobre los datos de validación para cada uno de los valores del parámetro a ajustar.

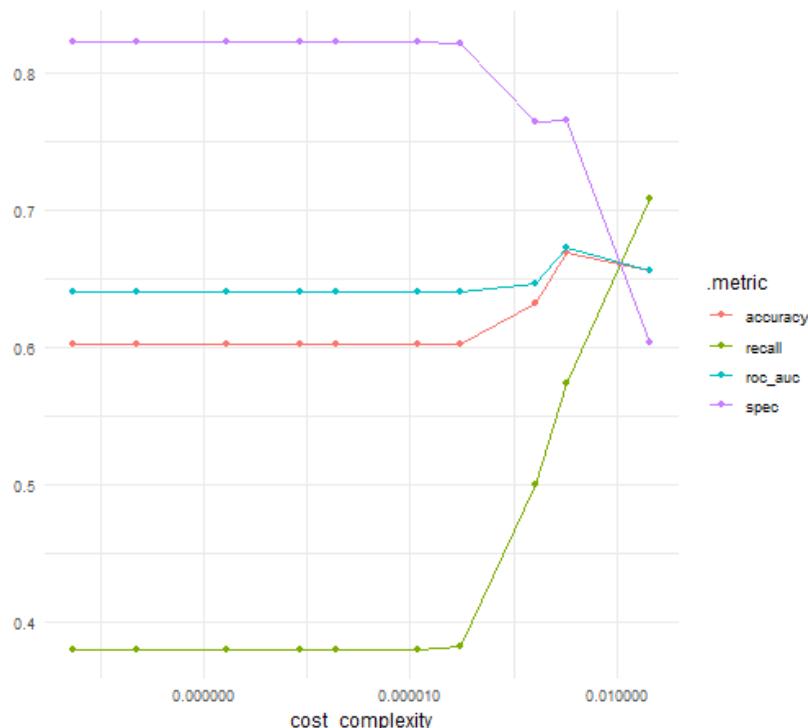


Figura 4.16: Métricas de rendimiento del árbol de decisión en función de el parámetro de costecomplejidad. *Fuente:* Elaboración propia.

4.2.4. Bosques aleatorios

En este modelo se ha fijado el número de árboles a 1000 y se han ajustado los parámetros *mtry* (el número de variables que se seleccionarán aleatoriamente en cada nodo) y *min_n* (el número de observaciones en un nodo a partir del cual no se sigue dividiendo y se convierte en nodo hoja). En este caso se ha optado por un enfoque diferente, motivado por el amplio rango de valores que puede tomar el parámetro *min_n* y por las limitaciones computacionales del equipo disponible.

De esta forma, la estimación de parámetros se ha hecho en dos etapas. En una primera etapa se ha fijado el parámetro *mtry* = 4 y se ha estimado el parámetro *min_n* considerando para ello un *grid* equiespaciado de 1000 a 2500 tomando valores de 100 en 100 (<{1000, 1100, ..., 2500}>). Para elegir entre los distintos modelos esta vez se ha usado como criterio la sensitividad, obteniendo el valor más elevado para *min_n* = 2100. En la segunda etapa, una vez *min_n*, este se ha considerado fijo y se ha estimado *mtry*, considerando una rejilla de 10 valores equiespaciados tomados del 1 al 10 (<{1, 2, ..., 10}>). De nuevo se ha utilizado la sensitividad para elegir el modelo final, eligiendo así *min_n* = 7. En la Figura ?? se recogen los resultados de las dos etapas de *tuning*. El modelo final elegido tiene *min_n* = 2100 y *min_n* = 7.

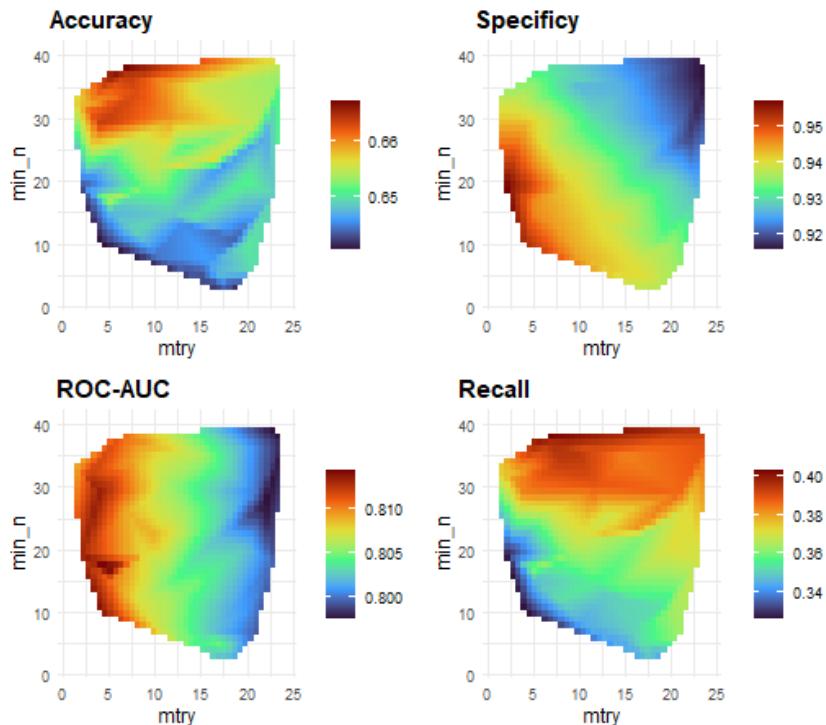


Figura 4.17: Métricas de rendimiento de Random Forest en función de los parámetros.
Fuente: *Elaboración propia*.

4.2.5. KNN

Para aplicar el modelo, primero se han transformado las variables categóricas en variables *dummy* y, posteriormente, se han tipificado las todas las variables. Se ha usado la distancia euclídea entre los vectores transformados. Para ajustar el parámetro *k* del

modelo se han tomado valores entre 1 y 400. La mayor tasa de acierto sobre los datos de validación se ha obtenido con $k = 275$. Los resultados del *tuning* se muestran en la Figura 4.18.

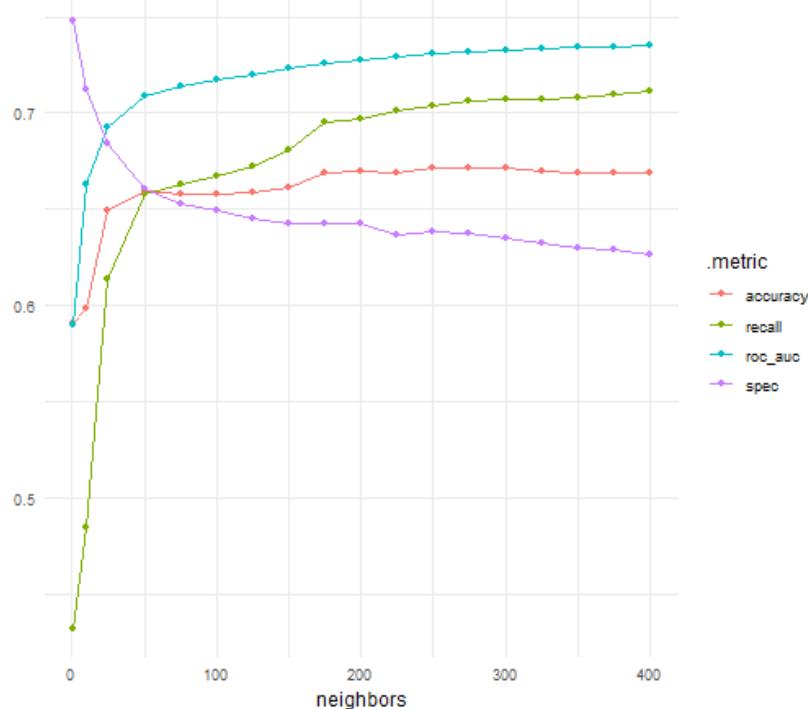


Figura 4.18: Métricas de rendimiento de KNN en función del número de vecinos. *Fuente: Elaboración propia.*

4.2.6. SVM lineal

Antes de construir el modelo, se han transformado las variables categóricas usando variables *dummy* y se han tipificado todas las variables. Se ha probado con 15 valores del parámetro *coste* entre 0.001949 y 24.666648. La mayor tasa de acierto y el mayor *recall* se han obtenido para $C = 0.0437$. Los resultados del *tuning* se muestran en la Figura 4.19

4.2.7. SVM radial

Por último, se ha construido el modelo de SVM usando un kernel gaussiano. El preprocesamiento ha sido el mismo que en el caso del kernel lineal. Dado el elevado tiempo de entrenamiento de este modelo solo se ha probado con 8 combinaciones de valores para los parámetros C y γ , que oscilan entre 0.005 y 31.7 y entre 0 y 0.05. La mayor tasa de acierto sobre los datos de validación se ha conseguido para $C = 31.7$ y $\gamma = 0.0000496$.

4.3. Comparación

A continuación se muestran las métricas de cada uno de los modelos seleccionados en los datos de validación en la Tabla 4.1 y en la Figura 4.20. Las curvas ROC de todos los

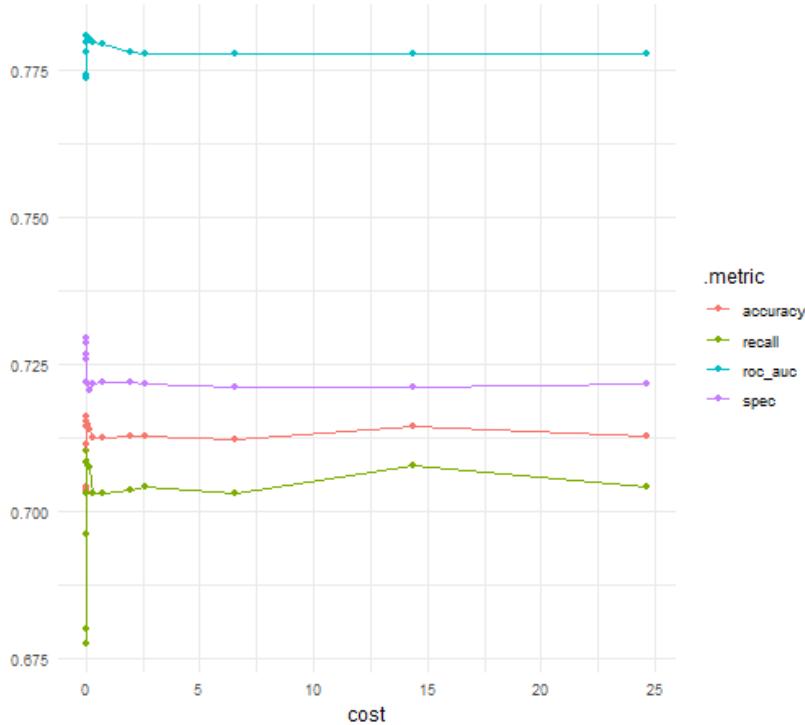


Figura 4.19: Métricas de rendimiento de SVM en función del parámetro C . *Fuente: Elaboración propia.*

model_name	roc_auc	accuracy	recall	specificity	precision
lr	0.785	0.719	0.700	0.738	0.727
lr_pca	0.727	0.659	0.624	0.694	0.670
dt	0.656	0.656	0.709	0.604	0.641
rf	0.781	0.725	0.758	0.693	0.711
svm_linear	0.781	0.716	0.710	0.722	0.718
svm_rbf	0.777	0.710	0.692	0.728	0.717
knn	0.732	0.672	0.706	0.637	0.660

Tabla 4.1: Métricas de los modelos seleccionados sobre el conjunto de validación. *Fuente: Elaboración propia.*

modelos se muestran en la Figura 4.21.

Puede observarse que los resultados obtenidos por todos los modelos son bastante similares. Destacan el modelo de bosque aleatorio y el de regresión logística con penalización, el primero por ser el que tiene la tasa de acierto y la sensitividad más elevadas, y el segundo por dar los mejores resultado en cuanto a precisión y especificidad. Las curvas ROC de los modelos de bosque aleatorio, regresión logística con penalización, SVM lineal y SVM radial son prácticamente iguales. Los modelos más pobres son la regresión logística aplicando PCA, KNN y el árbol de decisión.

Por último, para conocer la capacidad de generalización de los modelos construidos, estos se evaluarán sobre nuevas observaciones, el conjunto de datos test. Recuérdese que el entrenamiento de los modelos se ha realizado con el conjunto de entrenamiento, formado por 12927 observaciones tomadas entre el 2002 y mediados de 2014 y para ajustar los parámetros de cada modelo, se han utilizado 4309 observaciones tomadas entre mediados de 2014 y mediados de 2019. Finalmente, se evaluará la capacidad de predicción de los modelos sobre 4310 nuevas observaciones tomadas entre mediados de 2019 y 2022. Para

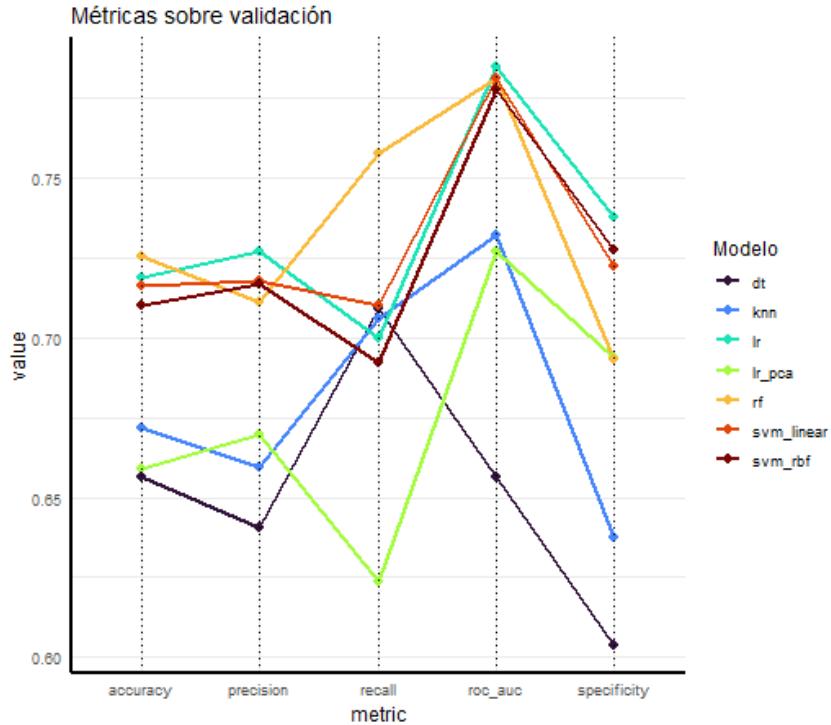


Figura 4.20: Gráfico de métricas obtenidas sobre el conjunto de validación por cada uno de los modelos seleccionados. *Fuente: Elaboración propia.*

model_name	roc_auc	accuracy	recall	specificity	precision
lr	0.795	0.710	0.726	0.693	0.718
lr_pca	0.715	0.645	0.631	0.661	0.667
dt	0.667	0.668	0.712	0.621	0.670
rf	0.762	0.699	0.727	0.669	0.703
svm_linear	0.790	0.705	0.729	0.680	0.710
svm_rbf	0.789	0.706	0.727	0.683	0.712
knn	0.744	0.673	0.719	0.624	0.673

Tabla 4.2: Métricas sobre el conjunto test. *Fuente: Elaboración propia.*

ello, primero se juntarán los conjuntos de entrenamiento y validación para reentrenar los modelos con la configuración de parámetros seleccionada en cada caso, y posteriormente se compararán los valores predichos por los modelos con los valores reales. Los resultados obtenidos se muestran en la Tabla 4.2 y en la Figura 4.22. Las curvas ROC de los distintos modelos sobre el conjunto de datos test se muestran en la Figura 4.23.

En este caso, los mejores resultados en todas las medidas los da el modelo de regresión logística con penalización. Los modelos de SVM muestran resultados bastante similares entre ellos y prácticamente iguales al modelo de regresión logística. Sobre los datos test, el modelo de bosque aleatorio ha dado un rendimiento peor que el obtenido en validación, quedando por detrás de los tres modelos ya comentados, aunque la sensibilidad de todos estos modelos es prácticamente igual. De nuevo, los peores resultados los dan los modelos de regresión logística aplicando PCA y el árbol de decisión, seguidos del KNN.

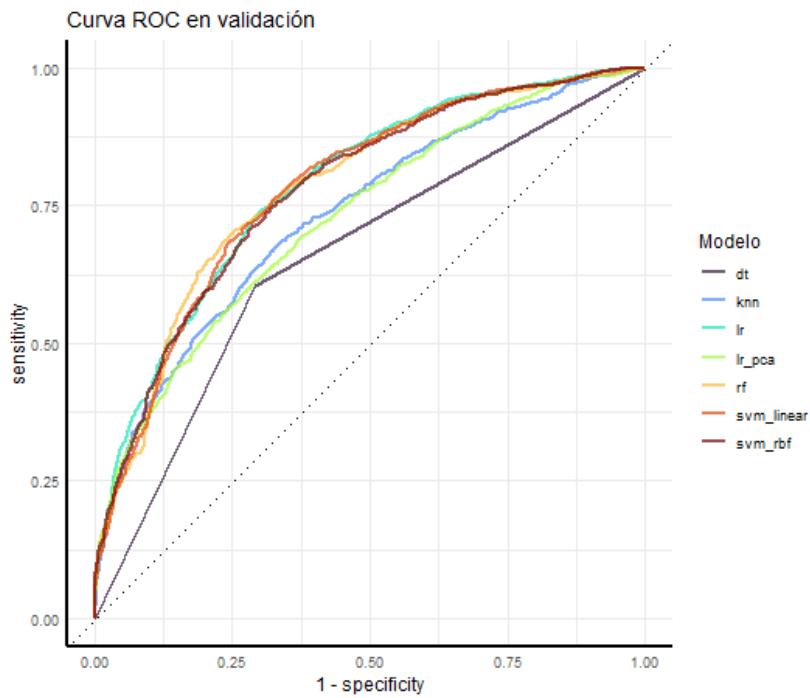


Figura 4.21: Curvas ROC sobre el conjunto de validación. *Fuente: Elaboración propia.*

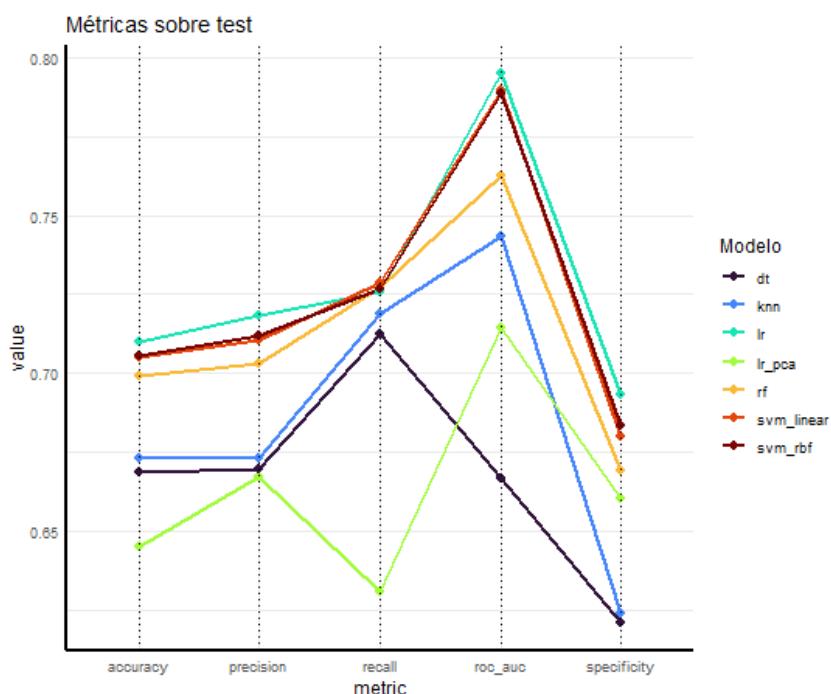


Figura 4.22: Métricas obtenidas sobre el conjunto test por cada uno de los modelos seleccionados. *Fuente: Elaboración propia.*

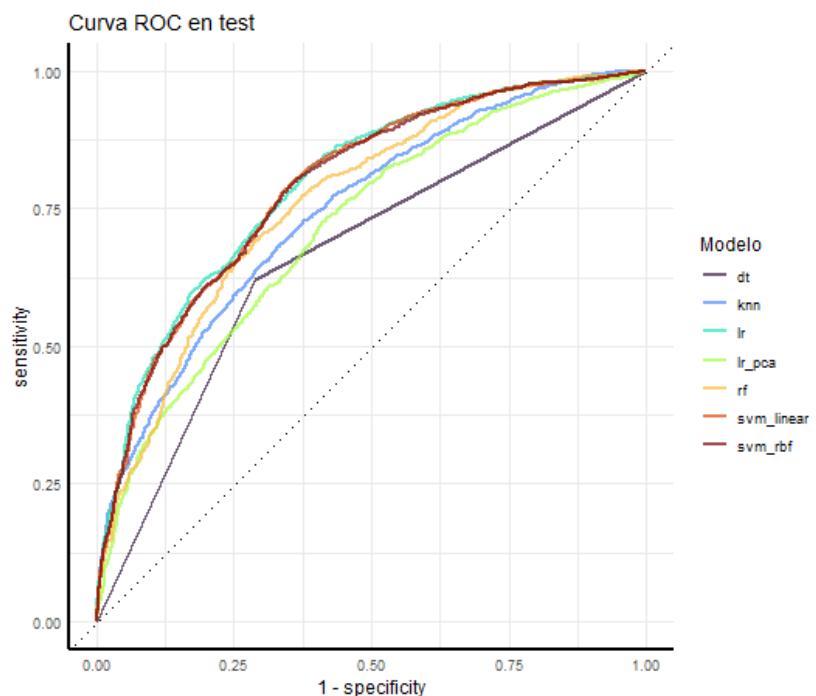


Figura 4.23: Curvas ROC sobre test. *Fuente: Elaboración propia.*

Capítulo 5

Aplicación de los modelos

En esta sección se pretende ilustrar el funcionamiento de los modelos construidos en la sección anterior, valorar su desempeño al ser aplicados en la realidad y conocer sus limitaciones.

5.1. Visión general del desempeño de los modelos

Dado que los modelos se han construido y evaluado sobre un conjunto de datos que es inevitablemente limitado y que se ve fuertemente influenciado por decisiones metodológicas, surge la incógnita de si las métricas de rendimiento obtenidas están realmente reflejando la capacidad de generalización de los modelos al ser aplicados en la realidad o de si, en cambio, tan solo están reflejando sesgos introducidos en el conjunto de datos a través del proceso de selección. Para entender mejor el funcionamiento de los modelos y conocer si realmente esto sirven para predecir incendios forestales en la vida real, se ha decidido adoptar el siguiente enfoque.

En primer lugar, se ha construido una malla de puntos con una resolución de 10km por 10km cubriendo toda la extensión de Andalucía (en este caso, se entiende como resolución la distancia entre los puntos en la dirección Este-Oeste y Norte-Sur) (Figura 5.1).

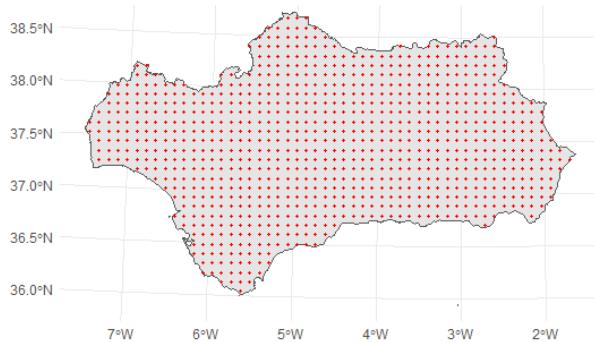


Figura 5.1: Malla de puntos con una resolución de 10km por 10km. *Fuente: Elaboración propia.*

A continuación, se ha asociado a cada uno de los puntos de la malla el valor de todas las variables predictoras el día 15 de cada mes del año 2022 en esa localización, usando

los métodos de preprocesamiento y depuración ya descritos. Estos datos se han utilizado para predecir el riesgo de incendio forestal en cada uno de los puntos de la malla el día 15 de cada mes, utilizando para ello los modelos finales de la sección anterior que mejor rendimiento mostraron sobre los datos test. Los resultados se muestran en las Figuras 5.2 (Regresión logística con penalización) y 5.3 (SVM lineal) y 5.4 (Random Forest).

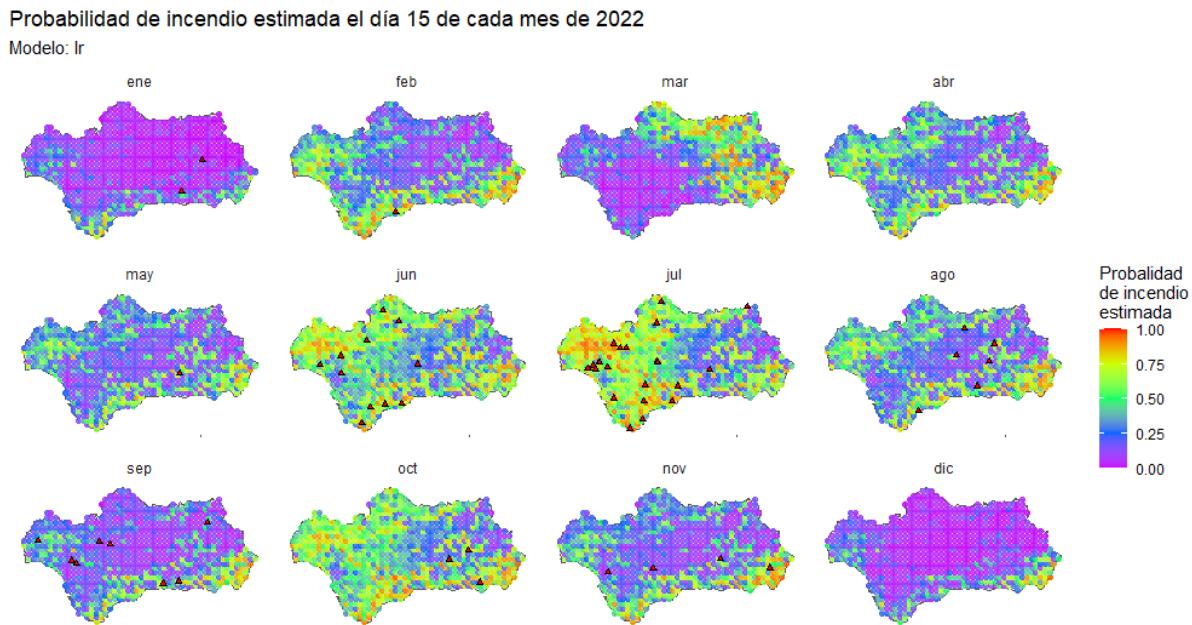


Figura 5.2: Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de regresión logística con penalización. Los triángulos indican los incendios de más de 100ha registrados en ese mes. *Fuente: Elaboración propia.*

Se puede observar que las predicciones del modelo de regresión logística con penalización y del SVM lineal son muy parecidos. A diferencia de las predicciones del modelo de Bosque aleatorio, que son bastante similares todos los meses, estos dos modelos muestran bastante variación mensual. Seguramente esta sea la causa de que, si bien el modelo de bosque aleatorio mostraba un buen rendimiento en validación, al evaluar su rendimiento sobre los datos test, este bajó significativamente. Es por ello que, a falta de la opinión de un experto en ecología del fuego, se opta por descartar el modelo de bosque aleatorio ya que no parece reflejar correctamente la variación estacional que se observa en la aparición de incendios forestales.

Se analizan por tanto las predicciones de los otros dos modelos (regresión logística y SVM). Se puede observar una clara componente estacional en las observaciones. En los meses de diciembre y enero se observan los niveles de riesgo más bajos a nivel global, mientras que los niveles de riesgo más elevados se encuentran en los meses de junio y julio, aunque por algún motivo también se observan niveles de riesgo elevado en los meses de marzo y octubre. Se puede observar también como las zonas con una probabilidad alta de incendio forestal varían en función del mes. Es curioso que en marzo ambos modelos den probabilidades de incendio tan elevadas en la zona oriental de la comunidad. Al margen del estudio específico y detallado de los mapas presentados, lo cual correspondería a los expertos en la materia y escapa de los objetivos de este trabajo, se puede observar que prácticamente todos los incendios se producen en zonas con una probabilidad de incendio elevada y que los modelos son capaces de ir más allá del mero estudio de las

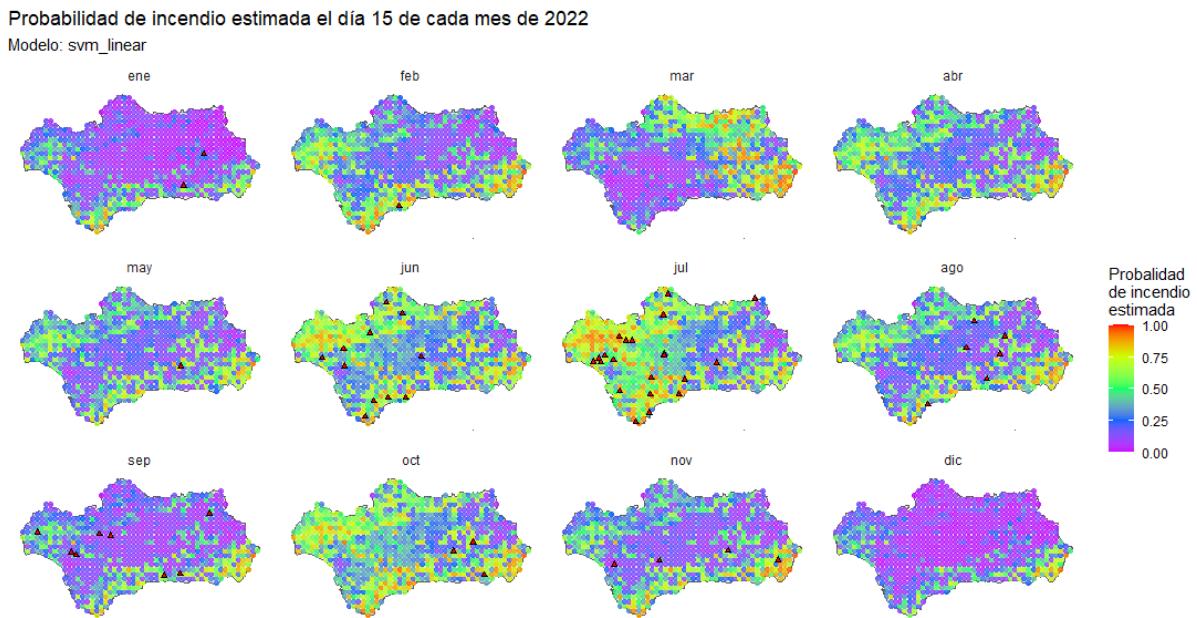


Figura 5.3: Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de SVM lineal. Los triángulos indican los incendios de más de 100ha registrados en ese mes. *Fuente: Elaboración propia.*

variables meteorológicas, ya que se observan zonas más o menos aisladas con una mayor probabilidad de incendio que se corresponden con las zonas en las que se ha observado un incendio. Esto indica que son otros factores los que el modelo está considerando para indicar riesgo de incendio, ya que la resolución espacial de las variables meteorológicas es bastante baja (50km), por lo que las variaciones a un mayor detalle son debidas a otros factores.

Este es, sin embargo, un enfoque bastante pobre, pues solo se está considerando el día 15 de cada mes, lo que podría llevar a conclusiones erróneas a la hora de evaluar los modelos (debidas, por ejemplo, a valores atípicos en ese día concreto). Esto es debido a las limitaciones computacionales del equipo disponible. Pese a ello, se ha podido ilustrar, aunque sin entrar en detalle, el desempeño de los modelos al aplicarlos para evaluar el riesgo de incendio en la realidad.

5.2. Caso estudio

A continuación, se pondrá a prueba el modelo de regresión logística construido con un caso real, el incendio de Sierra Bermeja, que se originó el 8 de septiembre de 2021 en el municipio de Jubrique en la provincia de Málaga (Figura 5.5). Se ha elegido este incendio por dos motivos. En primer lugar, porque fue el mayor incendio que hubo en España en el año 2021, con una superficie total afectada de 8607ha y una duración de 46 días hasta su extinción. Y en segundo lugar, porque fue un incendio intencionado, por lo que permitirá reflejar el comportamiento del modelo en incendios causados por el hombre.

Para analizar la capacidad de predicción del modelo para este incendio, se ha llevado a cabo el siguiente enfoque. Primero, se ha construido una malla de puntos con una

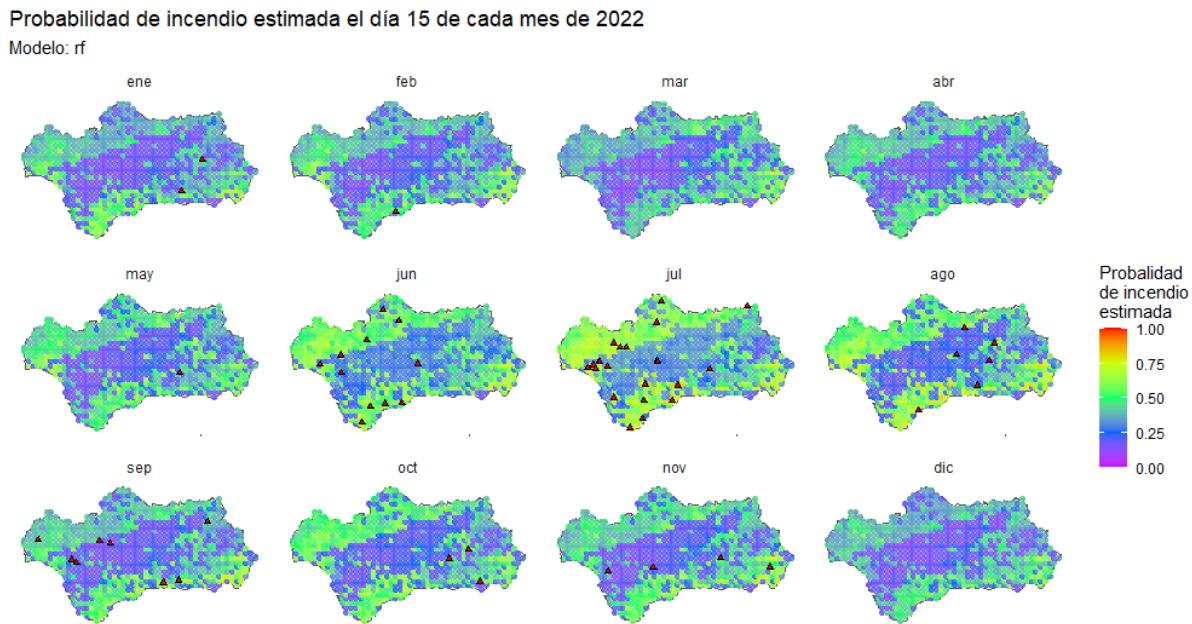


Figura 5.4: Probabilidades de incendios estimadas el día 15 de cada mes de 2022 con el modelo de Random Forest. Los triángulos indican los incendios de más de 100ha registrados en ese mes. *Fuente: Elaboración propia.*

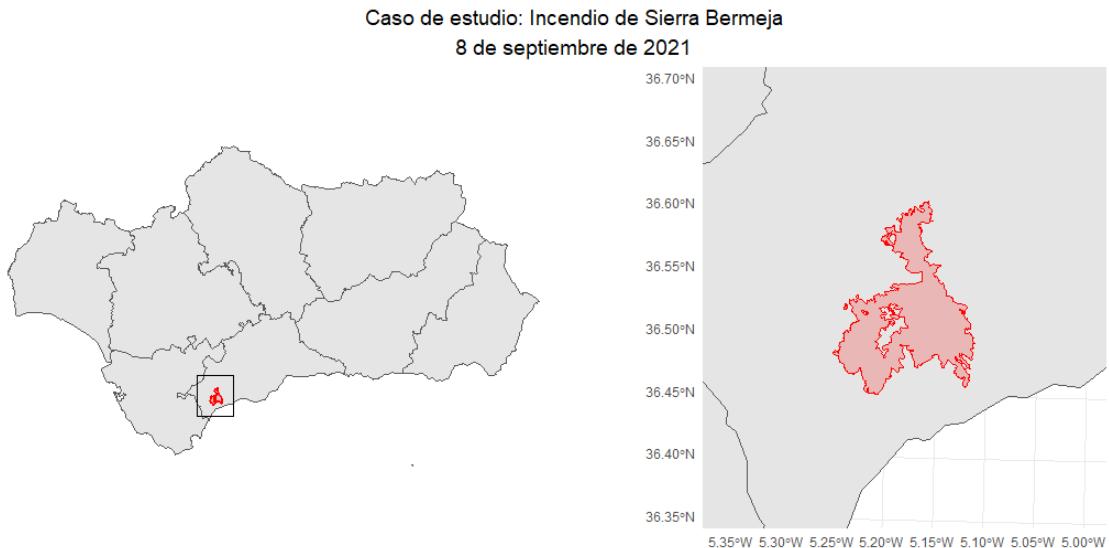


Figura 5.5: Área recorrida por el fuego en el incendio de Sierra Bermeja. *Fuente: Elaboración propia.*

resolución de 1km por 1km, cubriendo todo el *bounding box* de un *buffer* de 10km alrededor del perímetro del incendio. A continuación, en cada uno de estos puntos se han tomado todas las variables predictoras el día de origen del incendio, 15 y 30 días antes y 15, 30 y 45 días después. Con estos datos se ha utilizado el modelo de regresión logística con penalización para predecir la probabilidad de incendio forestal en cada uno de los días considerados en toda la malla de puntos. Los resultados se muestran en la Figura 5.6.

De este gráfico pueden extraerse varias conclusiones. Por un lado puede observarse que el día del origen del incendio se produce un aumento drástico de las probabilidades estimadas

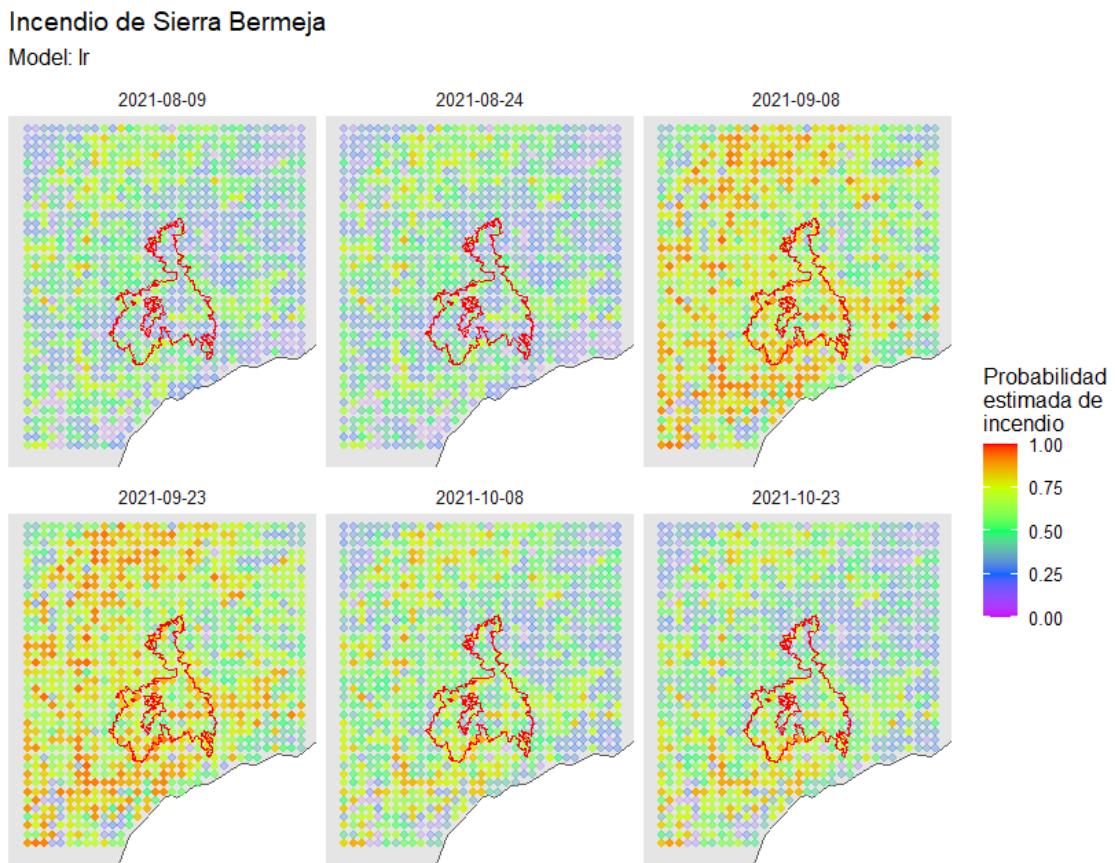


Figura 5.6: Mapa con las probabilidades de incendio estimadas en los días en torno al origen del incendio de Sierra Bermeja. El área total recorrida por el fuego se muestra en rojo. *Fuente: Elaboración propia.*

de incendio, las cuales continúan siendo muy altas 15 días después y, aunque disminuyen de forma general, se mantienen elevadas hasta 45 días después del inicio del fuego. Sin embargo, si bien es cierto que a nivel global el modelo sí parece aportar información estimando un riesgo muy alto de incendio en la región, al aumentar el nivel de detalle puede observarse que la capacidad discriminatoria del modelo disminuye significativamente. Esto es coherente, ya que la resolución de las variables climáticas es de aproximadamente 50km por 50km, por lo que no son adecuadas para trabajar con un nivel de detalle tan reducido.

Cabe mencionar que la variación observada en el riesgo de incendio estimado en las distintas fechas es debida, únicamente, a los cambios en las variables meteorológicas y en el NDVI. Esto es debido a que en el modelo de regresión logística construido no se han considerado las posibles interacciones entre las variables, lo cual podría ser de gran interés dadas las características del problema.

Capítulo 6

Conclusión y Trabajo futuro

6.1. Aportaciones

6.2. Conclusiones

6.3. Trabajo futuro

Apéndice A

Apéndice: Salidas

A.1. Gráficos espaciales EDA

A.1.1. Variables meteorológicas

Distribución espacial de T2M por mes

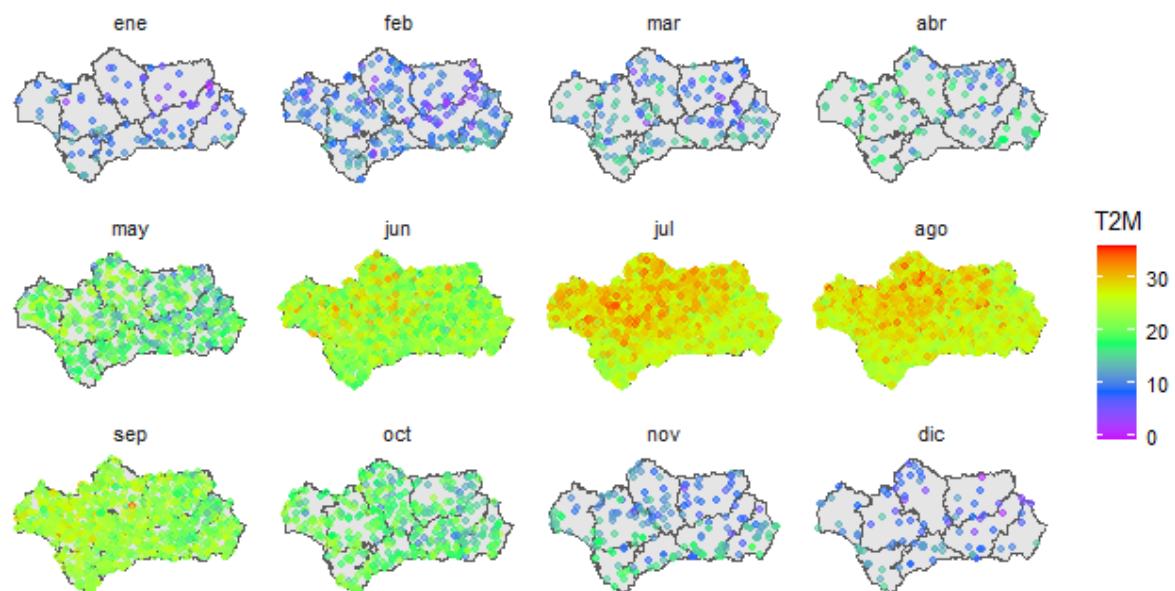


Figura A.1: Distribución espacial de $T2M$ por mes. *Fuente: Elaboración propia.*

Distribución espacial de RH2M por mes

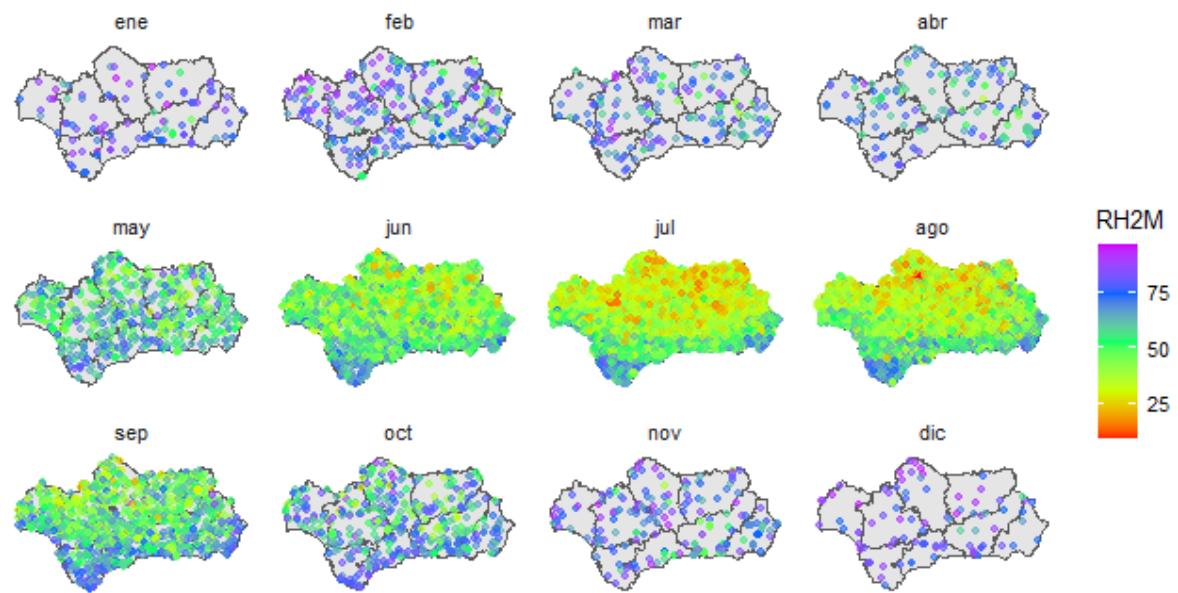


Figura A.2: Distribución espacial de $RH2M$ por mes. *Fuente: Elaboración propia.*

Distribución espacial de GWETTOP por mes

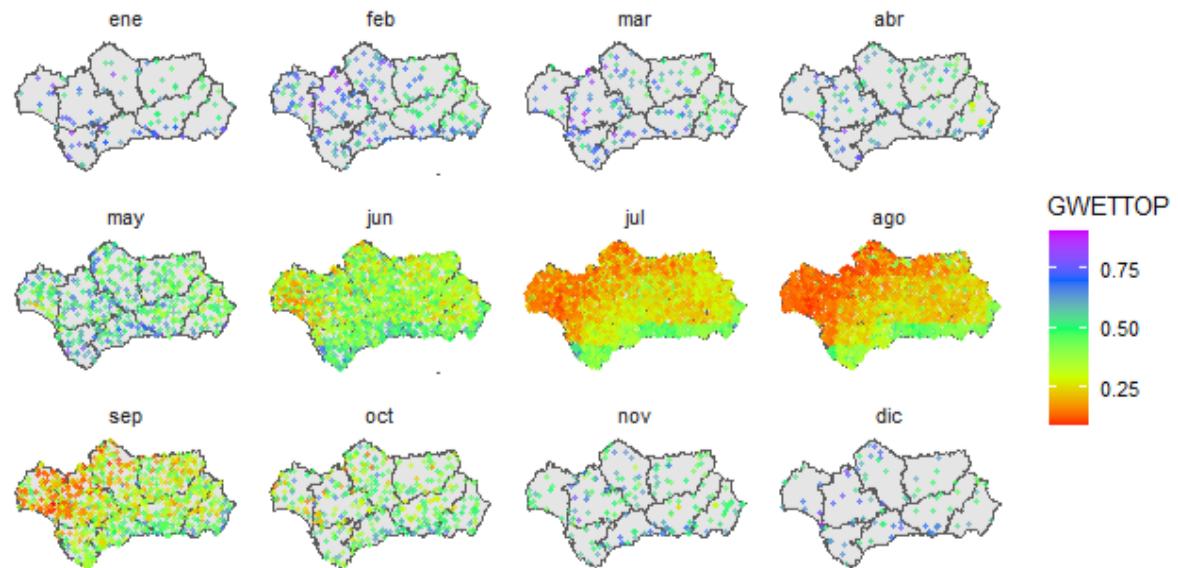


Figura A.3: Distribución espacial de $GWETTOP$ por mes. *Fuente: Elaboración propia.*

Distribución espacial de WS10M por mes

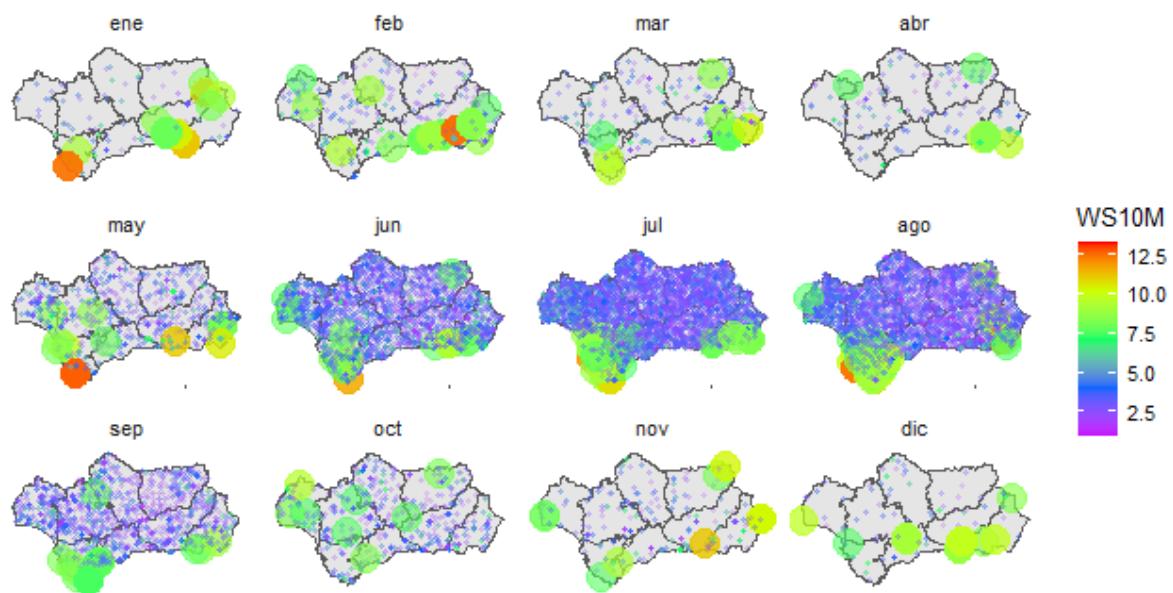


Figura A.4: Distribución espacial de *WS10M* por mes. *Fuente: Elaboración propia.*

Distribución espacial de PRECTOTCORR por mes

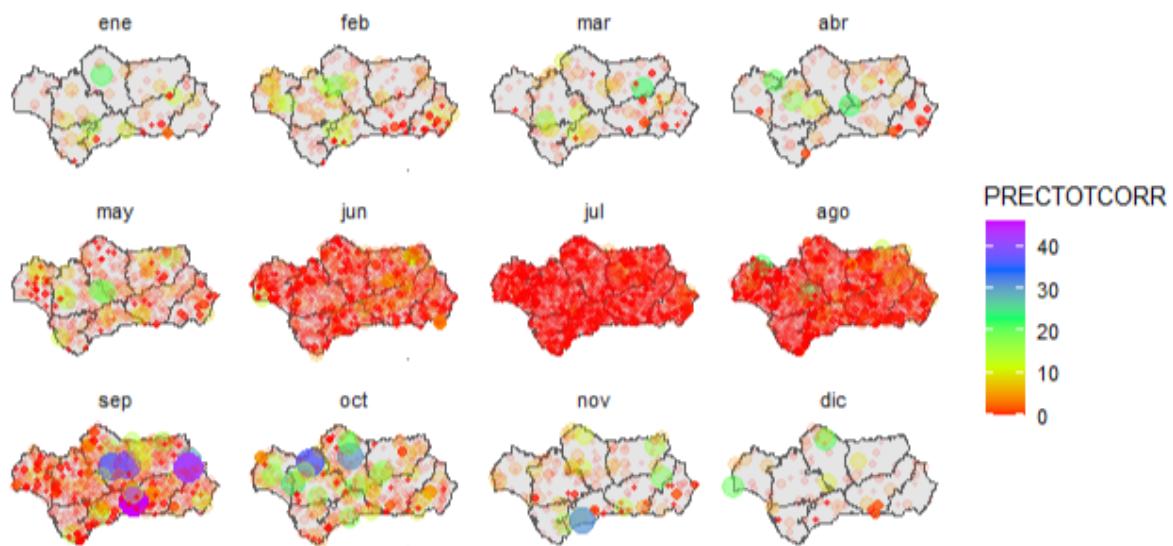


Figura A.5: Distribución espacial de *PRECTOTCORR* por mes. *Fuente: Elaboración propia.*

Distribución espacial de WD10M por mes

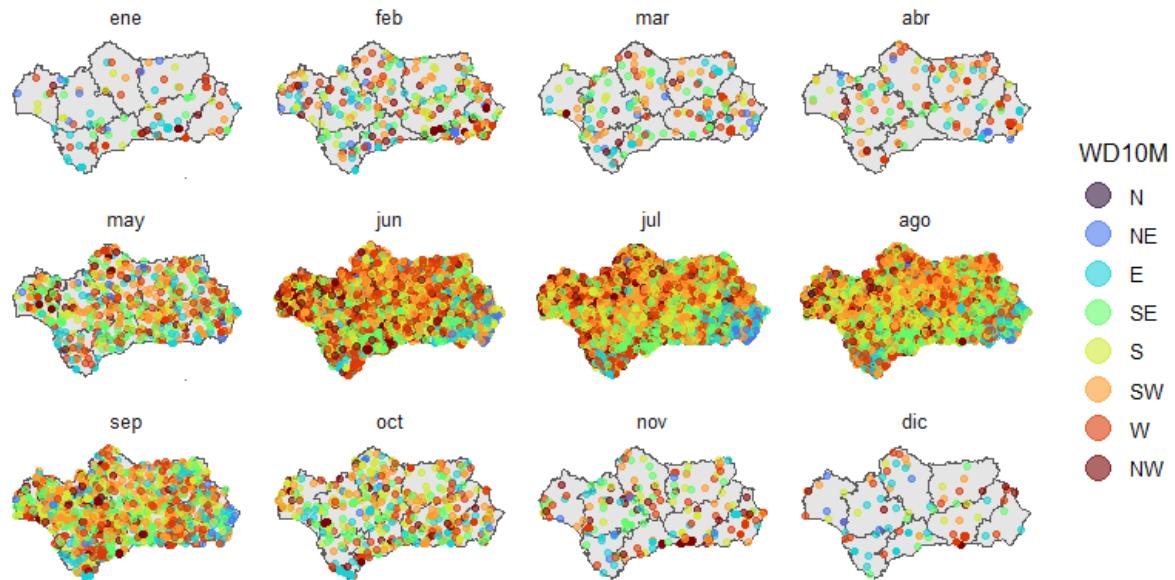


Figura A.6: Distribución espacial de $WD10M$ por mes. *Fuente: Elaboración propia.*

A.1.2. Variables demográficas

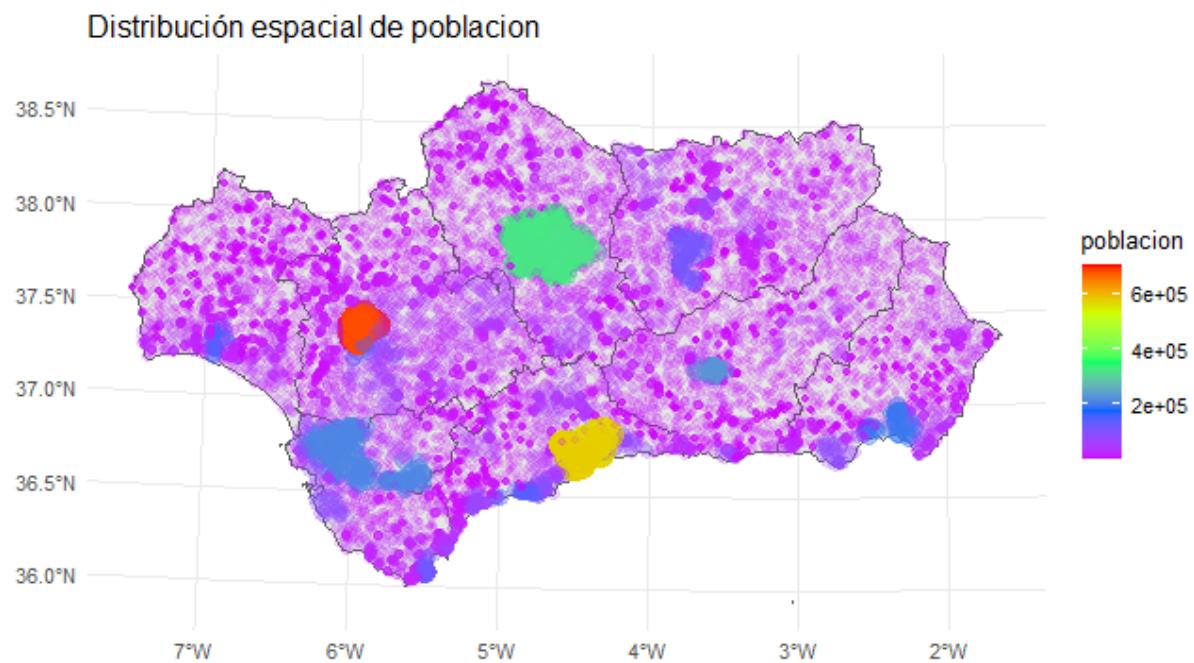


Figura A.7: Distribución espacial de *poblacion*. Fuente: Elaboración propia.

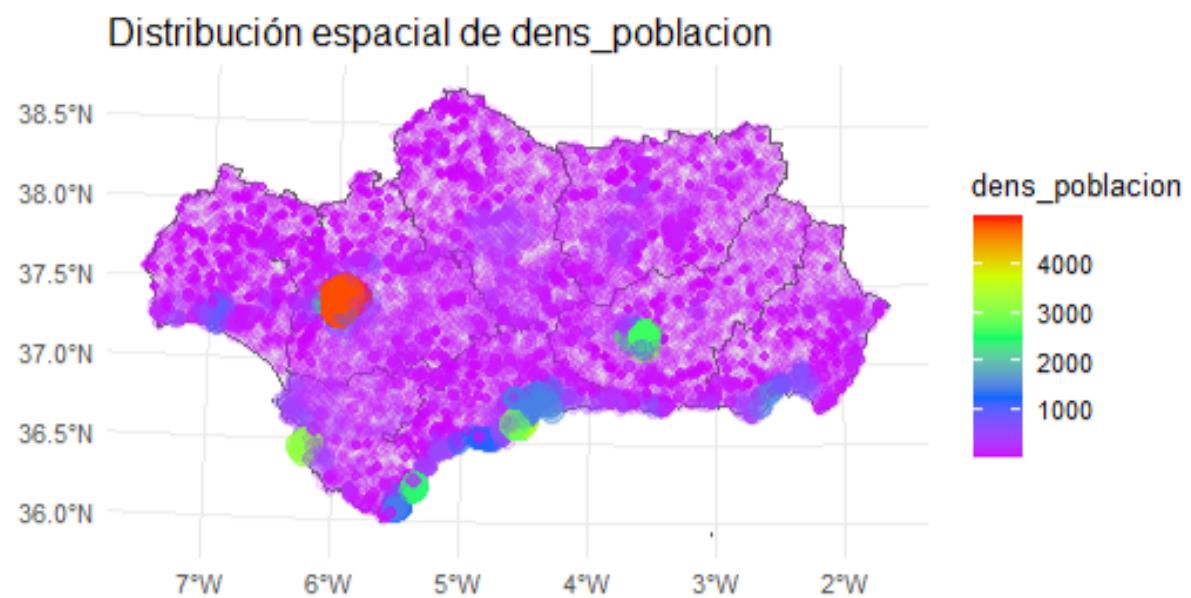


Figura A.8: Distribución espacial de *dens_poblacion*. Fuente: Elaboración propia.

A.1.3. Variable de vegetación

Distribución espacial de NDVI por mes

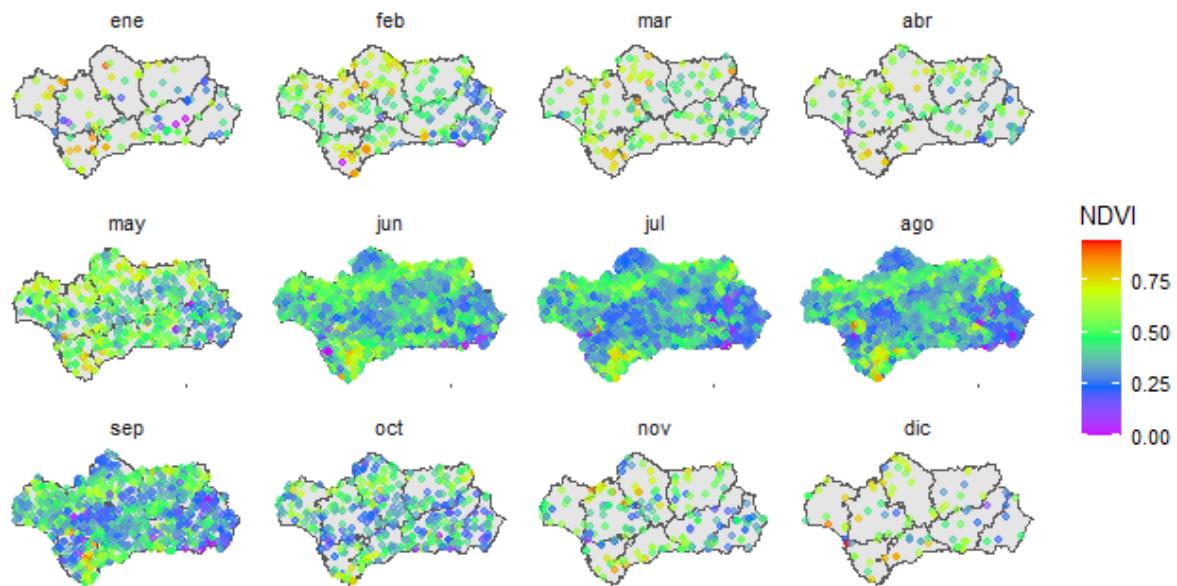


Figura A.9: Distribución espacial de *NDVI* por mes. *Fuente: Elaboración propia.*

A.1.4. Variables topográficas

Distribución espacial de elevacion

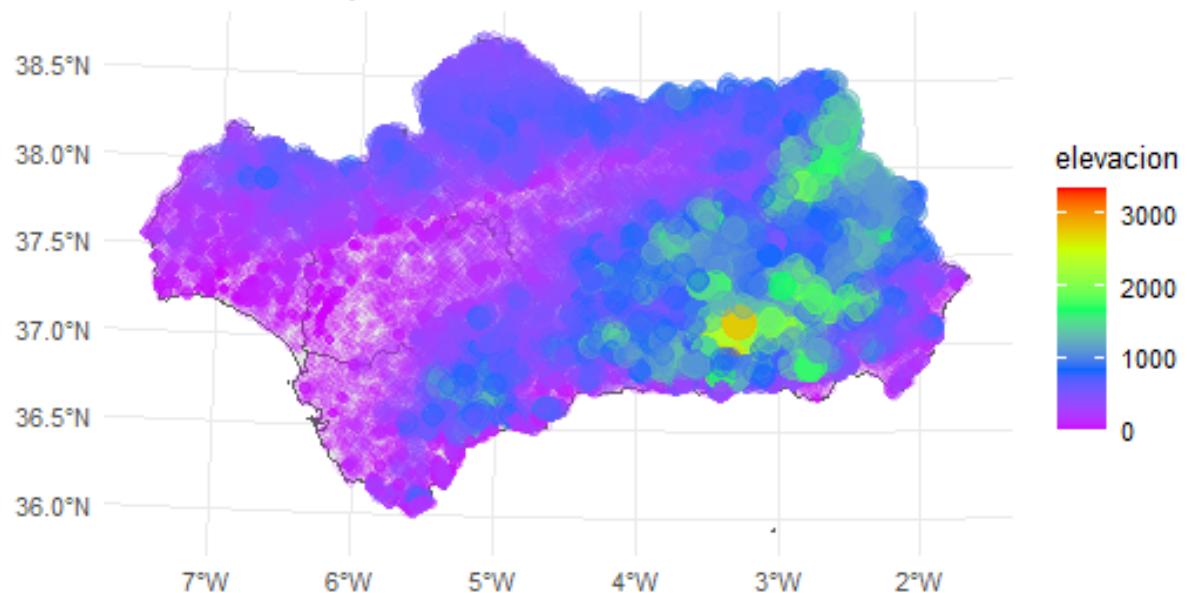


Figura A.10: Distribución espacial de *elevacion*. *Fuente: Elaboración propia.*

Distribución espacial de pendiente

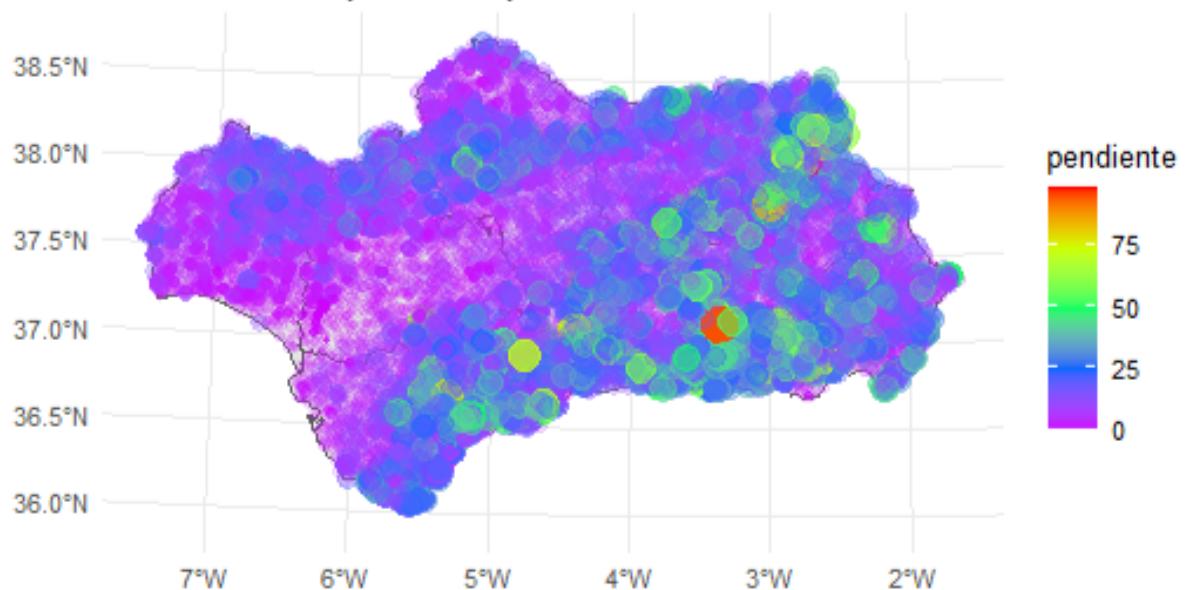


Figura A.11: Distribución espacial de *pendiente*. Fuente: *Elaboración propia*.

Distribución espacial de curvatura

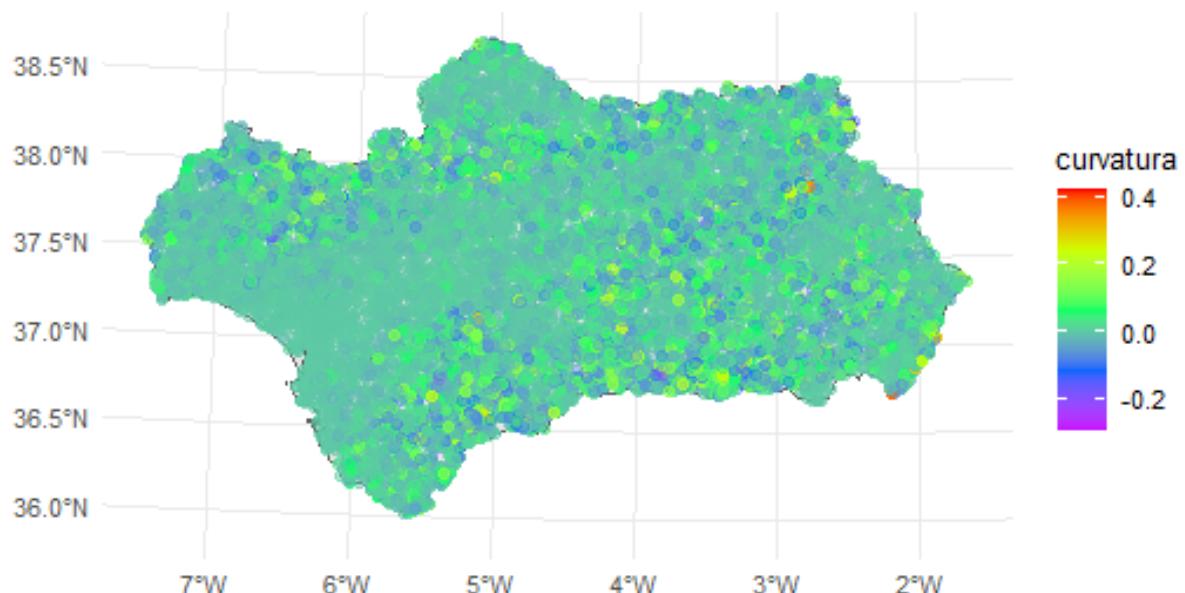


Figura A.12: Distribución espacial de *curvatura*. Fuente: *Elaboración propia*.

A.1.5. Variables antropológicas

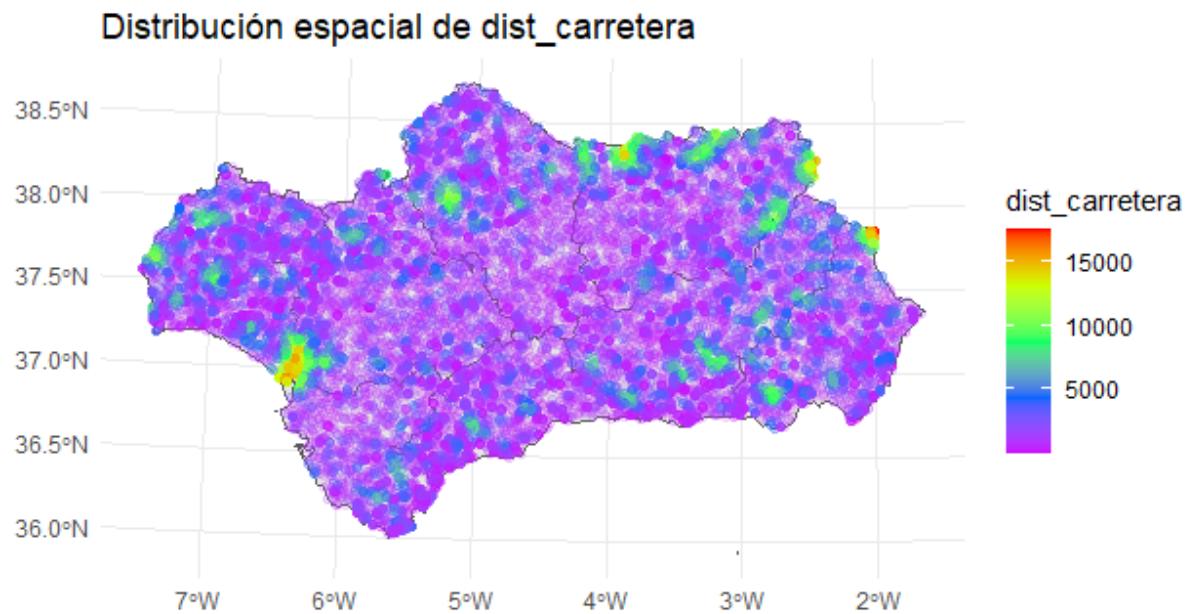


Figura A.13: Distribución espacial de *dist_carretera*. Fuente: *Elaboración propia*.

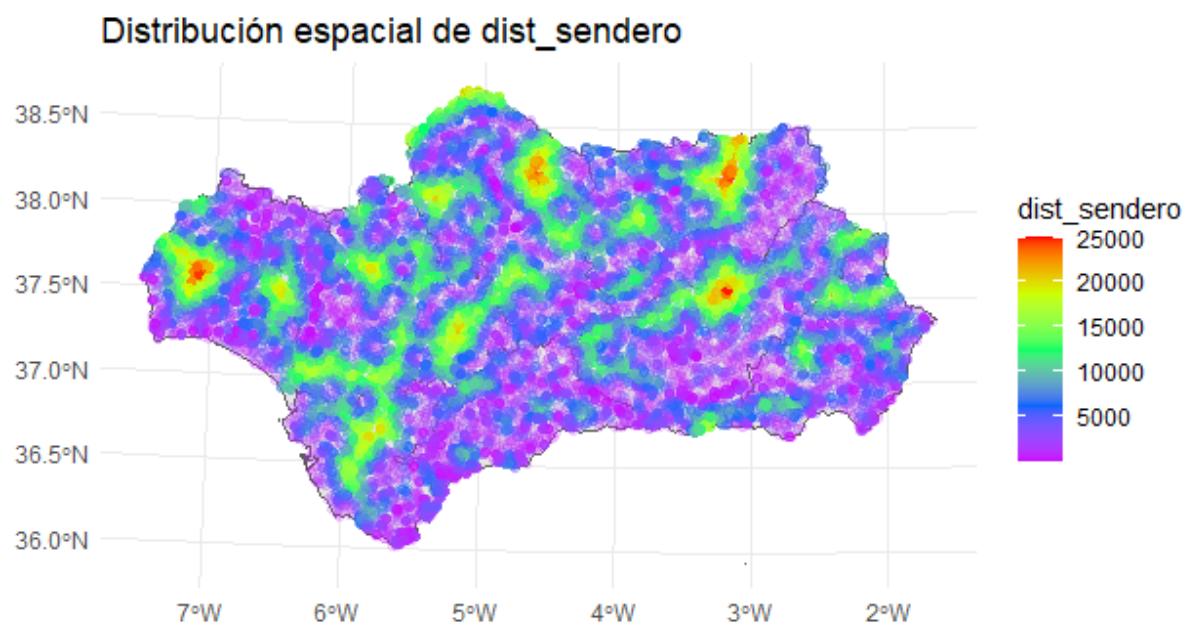
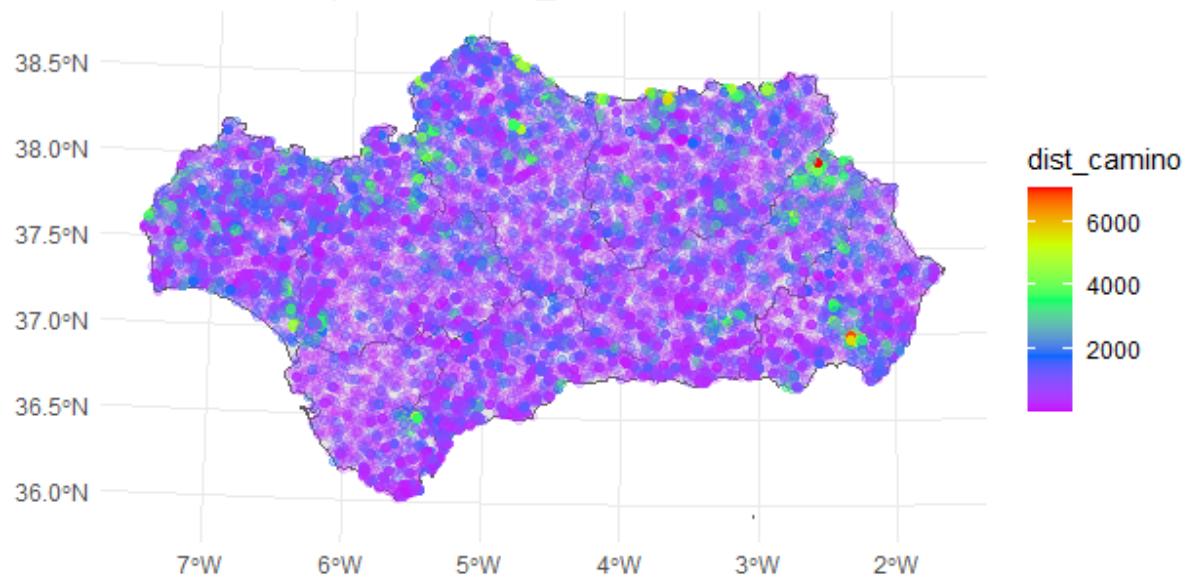
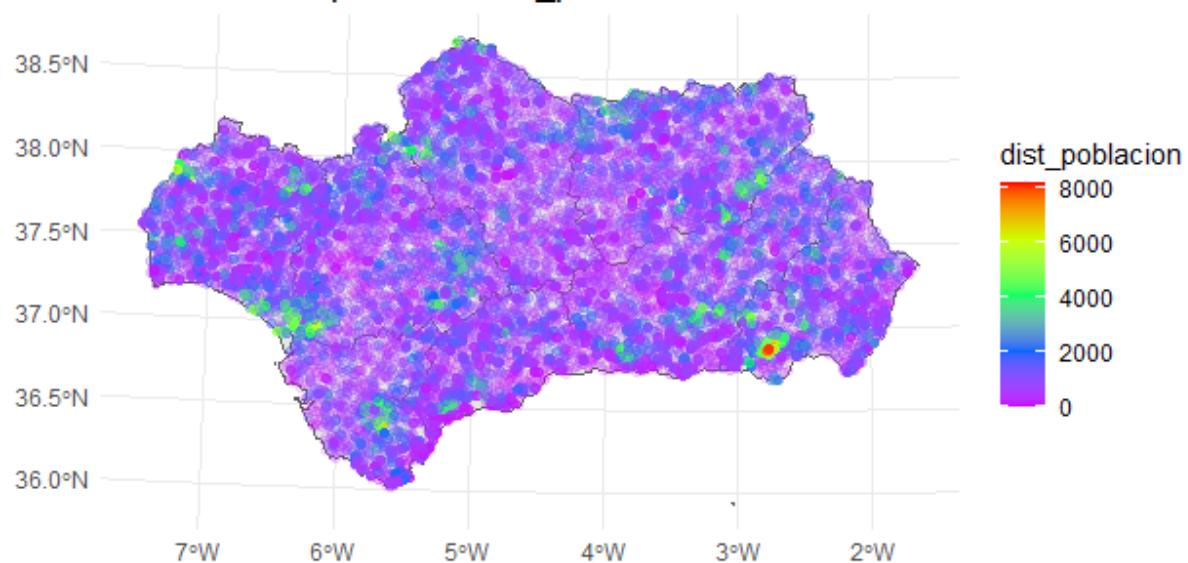


Figura A.14: Distribución espacial de *dist_sendero*. Fuente: *Elaboración propia*.

Distribución espacial de dist_caminoFigura A.15: Distribución espacial de *dist_camino*. *Fuente: Elaboración propia.***Distribución espacial de dist_poblacion**Figura A.16: Distribución espacial de *dist_poblacion*. *Fuente: Elaboración propia.*

Distribución espacial de dist_electr

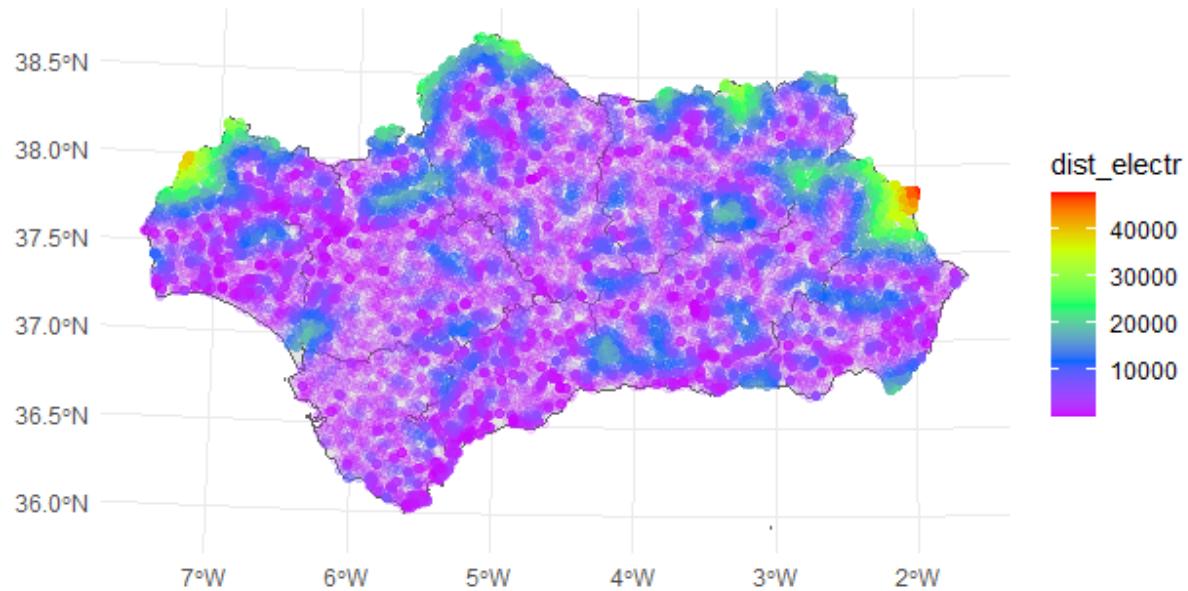


Figura A.17: Distribución espacial de dist_electr. Fuente: Elaboración propia.

Distribución espacial de dist_ferrocarril

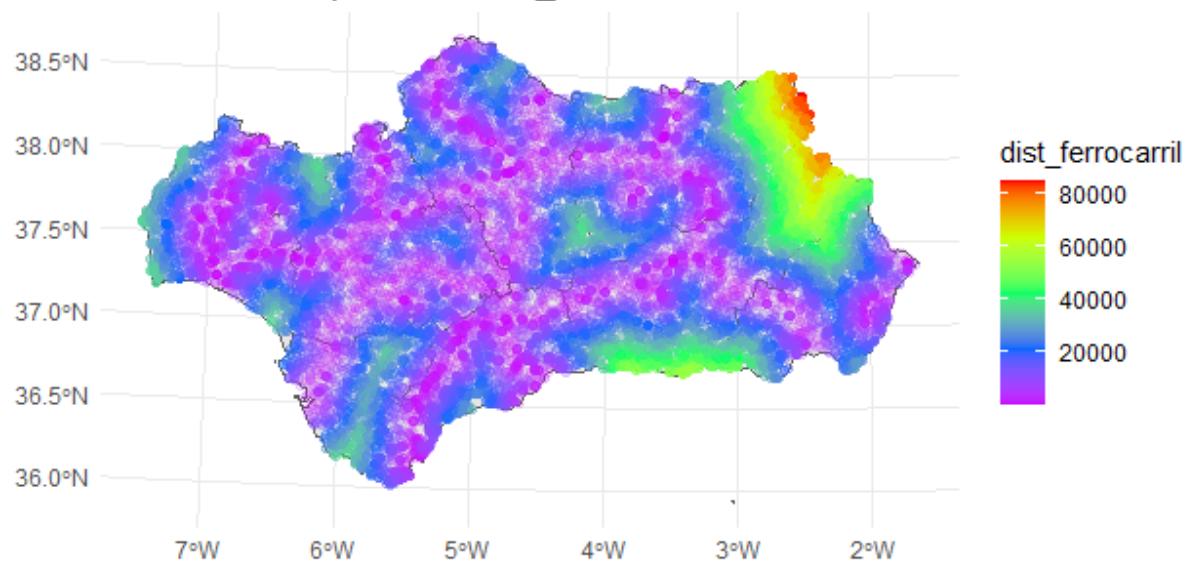


Figura A.18: Distribución espacial de dist_ferrocarril. Fuente: Elaboración propia.

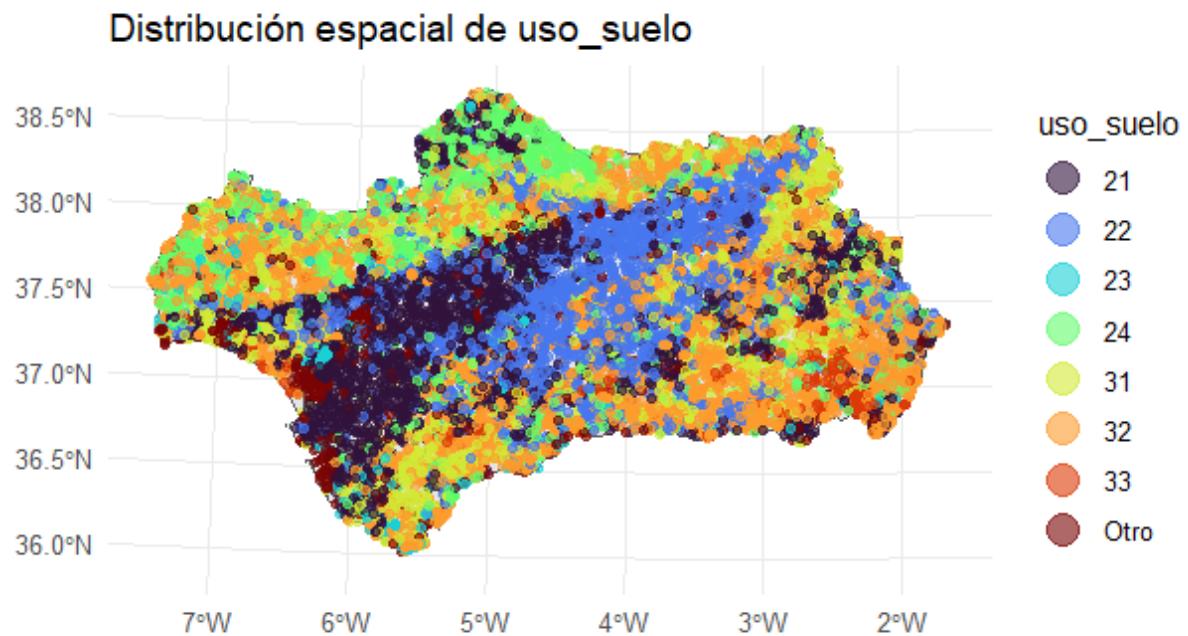


Figura A.19: Distribución espacial de *uso_suelo*. Fuente: *Elaboración propia*.

A.2. Salidas de los modelos

A.2.1. Coeficientes regresión logística con penalización

A.2.2. VIP Bosque Aleatorio

```
# A tibble: 59 x 3
  term      estimate
  <chr>     <dbl>
1 (Intercept) -0.0672
2 T2M           0.585 
3 GWETTOP      -0.0235
4 RH2M          -0.417 
5 WS10M          0.516 
6 PRECTOTCORR   -0.221
7 elevacion     -0.296 
8 pendiente      0.320 
9 curvatura      0.149 
10 dist_carretera -0.186
11 dist_poblacion -0.0631
12 dist_electr    -0.167 
13 dist_ferrocarril -0.182
14 dist_camino    -0.0636
15 dist_sendero    -0.244 
16 poblacion      -0.134 
17 dens_poblacion  0.0674
18 dist_rios       -0.0431
19 NDVI           -0.127 
20 WD10M_NE        -0.106
21 WD10M_E          0.0431
22 WD10M_SE         0.0740
23 WD10M_S          0.0711
24 WD10M_SW         -0.0302
25 WD10M_W          0.000845
26 WD10M_NW         -0.0260
27 orientacion_N    -0.0127
28 orientacion_NE   -0.000391
29 orientacion_E      0.0114
30 orientacion_SE      0.0228
31 orientacion_S       0.0454
32 orientacion_SW     -0.00710
33 orientacion_W     -0.0686
34 orientacion_NW      0
35 enp_X1            -0.183
36 uso_suelo_X22     -0.191
37 uso_suelo_X23      0.567
38 uso_suelo_X24      0.436
39 uso_suelo_X31      0.368
40 uso_suelo_X32      1.08 
41 uso_suelo_X33      0.356
42 uso_suelo_Otro      0.100
43 date_dow_Mon      -0.0554
44 date_dow_Tue      -0.0652
45 date_dow_Wed      0.00146
46 date_dow_Thu      0.0128
47 date_dow_Fri      -0.0116
48 date_dow_Sat      0.0545
49 date_month_Feb     0.209
50 date_month_Mar     0.123
51 date_month_Apr     0.0534
52 date_month_May     -0.0752
53 date_month_Jun     -0.366
54 date_month_Jul     -0.735
55 date_month_Aug     -0.600
56 date_month_Sep     -0.157
57 date_month_Oct     0.0359
58 date_month_Nov     0.107
59 date_month_Dec     0.00111
```

Figura A.20: Coeficientes del modelos de regresión logística lasso seleccionado. *Fuente: Elaboración propia.*

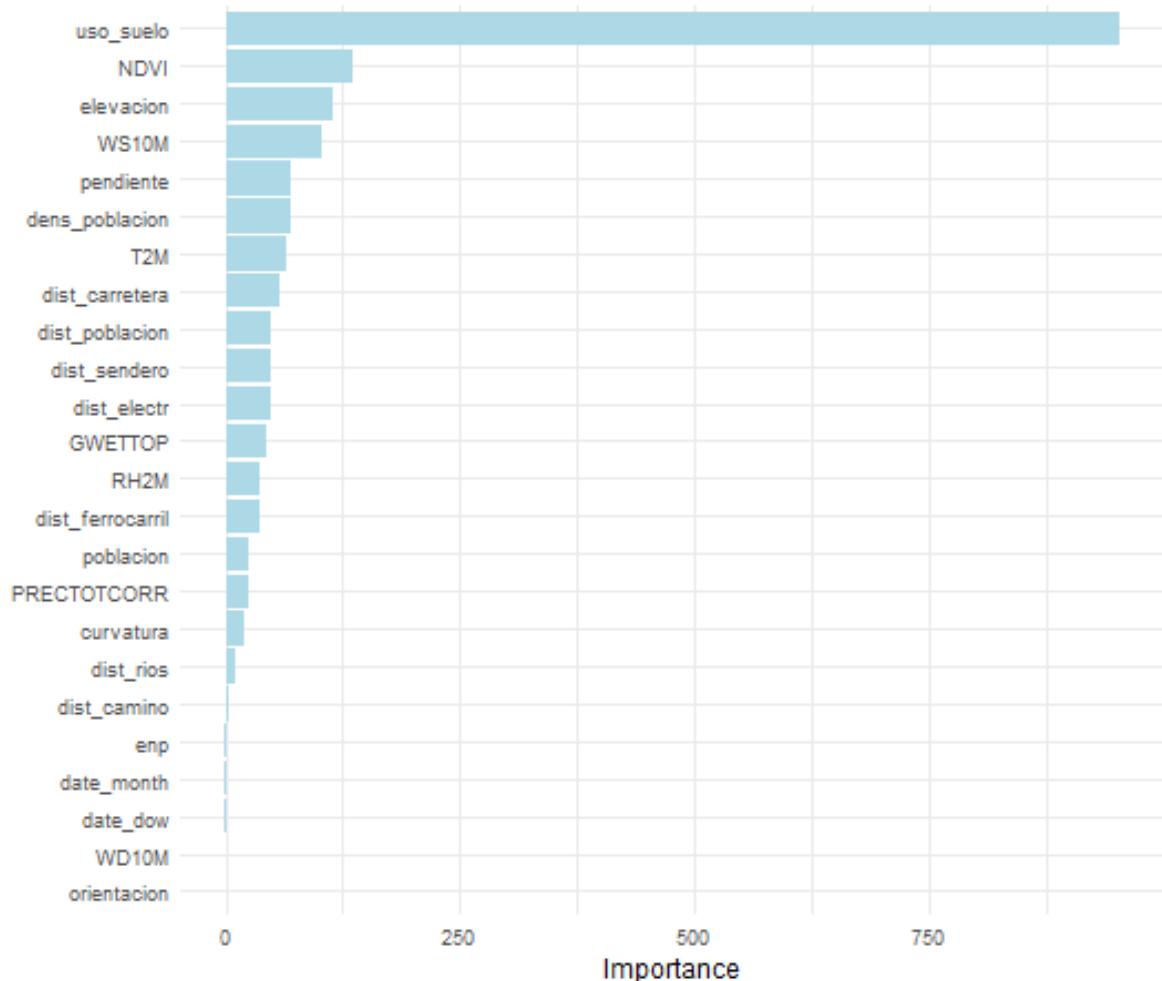


Figura A.21: Importancia de las variables en el Bosque Aleatorio final. *Fuente: Elaboración propia.*

Apéndice B

Apéndice: Código

B.1. Generación de la muestra

```
# Librerías -----
# Se cargan las librerías que se usarán en esta sección

library(terra) # Raster data
library(sf) # Vector data
library(mapSpain) # Polígonos de las regiones de España
library(tidyverse) # Manipulación de datos
library(lubridate) # Manipulación de fechas

# CRS de referencia -----
# Será el CRS que se use en todo el proyecto

pend <- rast("data_raw/topograficas/pendiente.tif")
crs_reference = crs(pend)
rm(pend) # Se elimina de la memoria para liberar espacio

# Polígono de Andalucía -----
Andalucia <- esp_get_ccaa(ccaa = "Andalucía") # Se obtiene el polígono
# de la comunidad autónoma de Andalucía
andalucia_proj <- st_transform(Andalucia,crs_reference) # Se transforma
# al sistema de referencia usado en el proyecto

# area_monte es el área donde se generarán las muestras negativas.

# Dado que no hay un mapa que indique claramente cuales son las zonas
# que se consideran "monte" en Andalucía y dado que los polígonos de
# incendios también cubren zonas agrícolas y urbanas (aunque menores
# en número que las zonas forestales), se considerará "monte" toda
# Andalucía, sin distinción. El sentido de esta variable es,
# precisamente, que pueda modificarse en futuros estudios
```

```
area_monte <- andalucia_proj

# Generación de la muestra ----

# Generación de la muestra estratificando por mes de forma que la
→ proporción de observaciones positivas y negativas por mes (en todo
→ el periodo) sea la misma

## Tamaño muestral -----
# Se dispone de 1089 incendios correctamente registrados entre 2002 y
→ 2022

n_in=10 # Número de puntos a muestrear dentro de cada polígono
n_out=1089*10 # Número de muestras negativas

## Generación aleatoria de fechas para las muestras negativas ---

# Primero se leen todos los datos de todos los archivos de incendios y
→ se almacenan en la variable incendios
incendios = NULL

for (year in 2002:2022) {
  incendios = rbind(incendios,
    → st_read(paste0("./data_raw/incendios_2000-2022/incendios_",
      year, ".shp")) %>%
    select("FECHA_INIC" =
      → matches("(?i)^FECHA_INIC$|^fecha_inic.$")))
}

# Se cuenta el número de incendios con fecha de inicio correcta en cada
→ mes
incendios_mes = incendios %>%
  mutate(FECHA_INIC = ymd(FECHA_INIC), .keep="unused") %>%
  filter(!is.na(FECHA_INIC)) %>%
  filter(year(FECHA_INIC)<=2022, year(FECHA_INIC)>=2002) %>%
  st_drop_geometry() %>%
  mutate(MES = month(month(FECHA_INIC))) %>%
  count(MES)

# Fechas posibles para las muestras negativas
possible_dates = tibble (date = seq(as.Date('2002/01/01'),
→ as.Date('2022/12/31'), by="day")) %>%
  mutate(MES = month(date)) %>%
  left_join(incendios_mes,
    join_by(MES))
```

```

set.seed(12345) # Se fija la semilla para que sea reproducible

# Se generan fechas aleatorias para las muestras negativas entre 2002 y
→ 2022 siguiendo con una distribución de probabilidad proporcional a
→ la cantidad de incendios observados en cada mes
dates = sample(possible_dates$date,
               n_out, replace = T,
               prob = possible_dates$n)

rm(incendios, possible_dates) # Se borran para liberar memoria

## Selección de localizaciones aleatorias -----
# Para la selección de la muestra se seguirá el siguiente
→ procedimiento:
# 1. Para las muestra positivas: Se tomarán n_in puntos aleatorios
→ dentro de cada polígono de incendio y se le asociará a cada uno de
→ ellos la fecha de inicio del incendio.
# 2. Para la muestras negativas: Se le asociará una localización
→ aleatoria dentro de area_monte a cada una de las fechas aleatorias
→ generadas dentro del periodo de estudio (dates). Se tendrá en cuenta
→ que no pueden haber muestras negativas a menos de 15km de una zona
→ en la que haya habido un incendio en una franja de 6 días alrededor
→ de la fecha de la observación (3 días antes a 3 días después).

points_in = NULL # Almacena las muestras positivas
points_out = NULL # Almacena las muestras negativas

for (year in 2002:2022) {

  cat("YEAR ", year, " : -----\\n")
  cat(" Generando muestras positivas...\\n")
  incendios <-
  → st_read(paste0("./data_raw/incendios_2000-2022/incendios_",year,
                  ".shp")),quiet=T) |>
    st_transform(crs = crs_reference) |>
    rename_with(.fn=tolower) |>
    mutate(fecha_inic=ymd(fecha_inic),geometry,.keep="none")

  ## Generación de puntos positivos

  for (i in 1:nrow(incendios)) {
    point_in_sfc <- st_sample(incendios[i,],size=n_in) # Se generan n_i
    → puntos dentro de cada incendio
    point_in_attr <- data.frame(fire = rep(1,n_in),date =
    → rep(incendios[i,]$fecha_inic,n_in))
    point_in <- st_sf(point_in_attr,geometry= point_in_sfc)
}
}

```

```
if (is.null(points_in)) {
  points_in <- point_in
} else {
  points_in <- points_in |>
    add_row(point_in)
}
}

## Generación de puntos negativos

cat(" Generando muestras negativas...\n")
# ---> Nota: los puntos se generan en area_monte

dates_year <- dates[year(dates) == year]
locations = NULL

for (day in dates_year) {
  incendios_day = filter(incendios, fecha_inic>=day-3 &
  ↪ fecha_inic<=day+3)
  if (nrow(incendios_day)==0){ # Si no ha habido incendios en una
  ↪ franja de 6 días en Andalucía
    if (is.null(locations)) {
      locations = st_sample(area_monte, size=1)
    } else {
      locations = c(locations, st_sample(area_monte, size=1))
    }
  } else { # Si ha habido algún incendio en una franja de 6 días en
  ↪ Andalucía (3 antes o 3 después)
    repeat {
      possible_location = st_sample(area_monte, size=1)
      if (!st_is_within_distance(possible_location,
                                st_union(incendios_day),
                                dist = 15000, sparse = FALSE)) {
        possible_location = st_sample(area_monte, size=1)
        if (is.null(locations)) {
          locations = possible_location
          break
        } else {
          locations = c(locations, possible_location)
          break
        }
      }
    }
  }
}

}
```

```

points_out_attr <- data.frame(fire = rep(0,length(dates_year)),date =
  dates_year)

if (is.null(points_out)) {
  points_out <- st_sf(points_out_attr,geometry= locations)
} else {
  points_out <- points_out |>
    add_row(st_sf(points_out_attr,geometry= locations))
}

}

sample <- rbind(points_in,points_out) # La muestra generada

# Comprobación y corrección -----
summary(sample) # Hay una fecha de un incendio errónea
max(sample$date,na.rm=T) # "2033-08-15"

# Se eliminan las observaciones con fecha de incendio errónea que se han
# detectado
sample <- sample[-which(sample$date==max(sample$date,na.rm=T)),]
summary(sample) # Corregido

# Almacenamiento de resultados -----
save(sample,file=paste0("salidas_intermedias/sample_strat_",
  Sys.Date(),".RData"))

```

B.2. Asignación de variables a localizaciones

A continuación se define la función `asignar_variables` que dada una muestra de puntos en Andalucía con fechas comprendidas entre 2002 y 2022 le asocia a cada observación todos los valores de las variables consideradas en el estudio. Esta función se usará varias veces a lo largo del trabajo.

```

# Librerías -----
# Se cargan las librerías que se usarán en esta sección
library(nasapower) # Para obtener la información meteorológica
library(raster, include.only = c("rasterFromXYZ")) # Función para
# construir rasters a partir de data.frames
library(tidyverse) # Manipulación de datos
library(sf) # Vector data
library(terra) # Raster data
library(mapSpain) # Polígonos de las regiones de España

```

```
library(lubridate) # Manipulación de fechas

asignar_variables = function(sample) {
  # Argumentos:
  # * sample: objeto sf con una columna de geometrías de tipo POINT
  #   (dentro de los límites de Andalucía) y fechas comprendidas entre
  #   01/01/2002 y 31/12/2022

  crs_reference = st_crs(sample) # Se usa el sistema de referencia de
  # coordenadas de la muestra
  and = esp_get_ccaa(ccaa = "Andalucía") %>% st_transform(st_crs(datos))
  # Polígono de Andalucía

  # Variables meteorológicas -----
  cat("Asignando variables meteorológicas...\n")

  # Transformamos los datos a WGS84
  andalucia_WGS84 <- st_transform(and,crs="WGS84")

  dataset = NULL # Variable en la que se almacenará el conjunto completo

  # Se trabaja anualmente pues la API de NASA POWER solo admite
  # consultas de hasta 366 días

  for (year in sort(unique(year(sample$date)))) {

    cat("YEAR ", year, " : -----\\n")

    # Los puntos de cada año
    points = filter(sample,year(date)==year)
    points_WGS84 <- st_transform(points,crs="WGS84")

    # Consulta a la api para obtener todo los valores del año
    daily_single_ag <- get_power(
      community = "ag",
      lonlat = c(-8,35.5,-1.5,39), # Límites de Andalucía
      pars = c("T2M","GWETTOP", "RH2M","WD10M","WS10M","PRECTOTCORR"),
      dates = paste0(year,c("-01-01","-12-31")),
      temporal_api = "daily")

    # Identificador
    daily_single_ag$clim_id <- 1:nrow(daily_single_ag)
    points$clim_id = NA # Inicializo el identificador

    for (day in unique(points$date)) {
```

```

points_day = points$date==day

# Seleccionar un día
clim_day <- filter(daily_single_ag, YYYYMMDD==day) |>
  dplyr::select(x = LON,y = LAT,clim_id= clim_id)

id_rast_day = rast(rasterFromXYZ(clim_day,crs="WGS84")) # Se crea
→ el raster con los identificadores

  points[points_day,]$clim_id <-
→ terra::extract(id_rast_day,points_WGS84[points_day,])$clim_id # Se
→ asocia a cada registro de la muestra el identificador
→ correspondiente
}

# Haciendo uso del identificador se asocian todas las variables
→ meteorológicas correspondientes a cada registro
points <- points |>
  left_join(select(daily_single_ag, -c(LAT,LON,DOY,YYYYMMDD)),
            by=join_by(clim_id)) |>
  select(-clim_id)

dataset = rbind(dataset,points)
}

rm(points,points_WGS84,daily_single_ag,clim_day,
  id_rast_day,points_day,day,year,andalucia_WGS84)

# Variables topográficas -----
cat("Asignando variables topográficas...\n")
elev <- rast("data_raw/topograficas/elevacion.tif")
pend <- rast("data_raw/topograficas/pendiente.tif")
orient <- rast("data_raw/topograficas/orientacion.tif")
curv <- rast("data_raw/topograficas/curvatura.tif")

# Es necesario pasarlas a numeric para poder trabajar con ellas y
→ extraer los valores
pend <- as.numeric(pend)
orient <- as.numeric(orient)
curv <- as.numeric(curv)

# Se extraen los valores de cada una de las capas
var_topograficas <- list(elevacion = elev,pendiente = pend,
                         orientacion = orient,curvatura = curv)

points_topograficas <- sapply(var_topograficas,function(x)
  terra::extract(x,dataset))[2,] |>

```

```
as_tibble()

dataset <- cbind(dataset,points_topograficas)

rm(elev,pend,orient,curv,var_topograficas,points_topograficas)

# Variables antropológicas -----
cat("Asignando variables antropológicas...\\n")

## Para optimizar el cálculo evitando que se repitan cálculos si hay
→ puntos repetidos:!!
dataset_geoms <- dataset %>%
  group_by(geometry) %>%
  group_keys() %>%
  st_sf(crs = st_crs(dataset))

### Carreteras: ----
carreteras <-
←  read_sf("data_raw/antropologicas/RedCarreteras/09_14_RedCarreteras.shp")
→  |>
  st_union()

dataset_geoms$dist_carretera <- st_distance(dataset_geoms,carreteras)
→  |>
  as.numeric()      # metres

rm(carreteras)

### Poblaciones: ----
poblaciones <-
←  read_sf("data_raw/antropologicas/Poblaciones/07_01_Poblaciones.shp")
→  |>
  st_union()

dataset_geoms$dist_poblacion <- st_distance(dataset_geoms,poblaciones)
→  |>
  as.numeric()      # metres

rm(poblaciones)

### Línea Eléctrica: ----
linea_electrica <-
←  read_sf("data_raw/antropologicas/LineaElectrica/10_14_LineaElectrica.shp")
→  |>
  st_union()
```

```

dataset_geoms$dist_electr <- st_distance(dataset_geoms,linea_electrica)
↪ |>
as.numeric() # metres

rm(linea_electrica)

### Ferrocarril: ----
ferrocarril <-
↪ read_sf("data_raw/antropologicas/Ferrocarril/09_21_Ferrocarril.shp")
↪ |>
st_union()
dataset_geoms$dist_ferrocarril <-
↪ st_distance(dataset_geoms,ferrocarril) |>
as.numeric()

rm(ferrocarril)

### Camino / Via: ----
camino <- read_sf("data_raw/antropologicas/Camino/09_19_Camino.shp")
viapec <-
↪ read_sf("data_raw/antropologicas/Camino/09_22_ViasPecuarias.shp")

camino_viapec <- c(st_geometry(camino),st_geometry(viapec))
rm(camino,viapec)

camino_viapec <- st_union(camino_viapec)

dataset_geoms$dist_camino <- st_distance(dataset_geoms,camino_viapec)
↪ |>
as.numeric()

rm(camino_viapec)

### Sendero / Vía Verde / CarrilBici: ----
viaverde <-
↪ read_sf("data_raw/antropologicas/Sendero_ViaVerde/09_24_ViaVerde.shp")
↪
sendero <-
↪ read_sf("data_raw/antropologicas/sendero_ViaVerde/09_20_Sendero.shp")
↪
carrilbic <-
↪ read_sf("data_raw/antropologicas/sendero_ViaVerde/09_23_CarrilBici.shp")

sendero_viaverde_carrilbici <-
↪ c(st_geometry(viaverde),st_geometry(sendero),st_geometry(carrilbic))
↪ |>
st_union()

```

```
dataset_geoms$dist_sendero <-
  → st_distance(dataset_geoms,sendero_viaverde_carrilbici) |>
    as.numeric()

rm(sendero,sendero_viaverde_carrilbici,viaverde,carrilbic)

#### ENP: ----
enp1 <-
  → read_sf("data_raw/antropologicas/ENP/11_07_Enp_FiguraProteccion.shp"
  → )
enp2 <-
  → read_sf("data_raw/antropologicas/ENP/11_07_Enp_RegimenProteccion.shp")

enp <- c(st_geometry(enp1),st_geometry(enp2)) |> st_union()
enp_sf <- st_sf(enp)

# Se rasteriza para aumentar la eficiencia computacional
enp_rast <- rasterize(enp_sf,
  → rast("data_raw/topograficas/pendiente.tif"), #
  →       Modelo
  → background = 0)
dataset_geoms$enp= terra::extract(enp_rast,dataset_geoms)[,2]

rm(enp,enp1,enp2,enp_sf,enp_rast)

#### Uso Suelo: ----
# Inicialmente se ha rasterizado para aumentar la eficiencia
# computacional
# UsoSuelo <-
  → read_sf("data_raw/antropologicas/UsoSuelo/06_01_UsoSuelo.shp")
# UsoSuelo_rast <- rasterize(UsoSuelo,
#   → rast("data_raw/topograficas/pendiente.tif"), # Modelo
#       field="cod_uso")

UsoSuelo_rast <- rast("data_cleaning/uso_suelo_rast.tiff")

dataset_geoms$uso_suelo =
  → terra::extract(UsoSuelo_rast,dataset_geoms)[,2]

# Hidrográficas -----
cat("Asignando variables hidrográficas...\n")

#### Distancia a ríos: ----
rios <- read_sf("data_raw/hidrograficas/Rios_Espana.shp") |>
  st_transform(st_crs(dataset)) |>
```

```

st_crop(xmin = 100394.4, # Esto se hace solo para no tener que
→ considerar todo el file y que sea más eficiente
→ computacionalmente
    ymin = 3976888.6,
    xmax = 690000.8,
    ymax = 4350000.0) |>
st_union()

dataset_geoms$dist_rios <- st_distance(dataset_geoms,rios) |>
as.numeric() # metres

rm(rios)

## Se vuelven a desagrupar los registros y se le asigna a cada
→ registro los valores correspondientes calculados! !
dataset <- dataset %>%
  st_join(dataset_geoms,left = TRUE) # Es un left join espacial

# Demográficas -----
cat("Asignando variables demográficas...\n")

#### Población y densidad de población: ----

poblacion <-
  read_csv2("data_raw/antropologicas/Población/poblacion_municipios.txt",
            locale=locale(decimal_mark = ","),
            col_select = 1:5,col_types = "ccifn") |>
  mutate(Valor=as.integer(round(Valor))) # Por algún motivo aparecen
→ decimales

area_municipios <-
  read_csv2("data_raw/antropologicas/Población/extencion_municipal.txt",
            locale=locale(decimal_mark = ","),
            col_select = 1:6, col_types = "fffffn")

area_municipios <- area_municipios %>%
  filter(!is.na(CODIGO_INE3)) %>%
  select(CODIGO_INE3,Valor) %>%
  rename("Area" = "Valor")

# Se calcula la densidad de población anual como el cociente del
→ número de habitantes entre la extensión del municipio
dens_poblacion <- poblacion %>%
  select(-Medida) %>%
  rename("Poblacion" = "Valor",
        "Municipio" = "Lugar de residencia") %>%

```

```
left_join(area_municipios,
          join_by("CODIGO_INE3")) %>%
  mutate(dens_poblacion = Poblacion/Area) %>%
  select(-Area)

municipios <- esp_get_munic(epsg = 4258,region = "Andalucía")

municipios <- municipios |>
  st_transform(crs_reference)

# Se asocia cada observación su código de municipio correspondiente

num_mun = st_intersects(dataset,municipios)

# Se eliminan las observaciones que no están en ningún municipio
if (any(sapply(num_mun,function(x) length(x) == 0))) {
  cat("Eliminamos las
  → observaciones:\n",which(sapply(num_mun,function(x) length(x) ==
  → 0)))
  dataset = dataset[-which(sapply(num_mun,function(x) length(x) ==
  → 0)),]
}

dataset$cod_municipio <-
  → municipios[unlist(st_intersects(dataset,municipios)),]$LAU_CODE

dataset <- dataset |>
  left_join(dens_poblacion,
            join_by(cod_municipio==CODIGO_INE3,YEAR==Anual)) |>
  rename("municipio" = "Municipio",
         "poblacion" = "Poblacion")

# Vegetación -----
cat("Asignando variables de vegetación...\n")

### NDVI ----
dataset$NDVI = NA

for (YEAR in 2002:2022) {
  for (MONTH in 1:12) {
    MM = str_pad(MONTH,2,"left",pad = "0")
    YY = substr(as.character(YEAR),3,4)
    # if (as.numeric(YY)<=01) {
    #   ruta <-
    #   paste0("data_raw/vegetacion/",YEAR,"NOAAVHMEDMNDVI/InfGeografica/InfRaster"
    # } else
    if (as.numeric(YY)<=06) {
```

```

ruta <-
→ paste0("data_raw/vegetacion/",YEAR,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIFF/TI
  } else if (as.numeric(YY)<=11) {
    ruta <-
→ paste0("data_raw/vegetacion/",YEAR,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIF/TE
  } else if (as.numeric(YY)<=21) {
    ruta <-
→ paste0("data_raw/vegetacion/",YEAR,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIFF/t
  }else {
    ruta <-
→ paste0("data_raw/vegetacion/",YEAR,"TERMODMEDMNDVI/InfGeografica/InfRaster/COG/te
  }

if (file.exists(ruta)) {
  cat(YEAR,MONTH,"\n")
  # Observaciones en ese mes y año
  isMY = dataset$YEAR==YEAR & dataset$MM==MONTH
  if (any(isMY)) {
    NDVI_rast = as.numeric(rast(ruta))
    if (MONTH==4 & YEAR==2011){
      # Ese archivo viene defectuoso y se le asigna el CRS de los
      → otros archivos del mismo año (todos los demás del año
      → tienen el mismo)
      crs(NDVI_rast) =
        "PROJCRS[\\"WGS 84 / UTM zone 30N\\",\n          BASEGEOGCRS[\\"WGS
        → 84\\"],\n            DATUM[\\"World Geodetic System 1984\\",\n        → ELLIPSOID[\\"WGS 84\\",6378137,298.257223563,\n        → LENGTHUNIT[\\"metre\\",1]]],\n        → PRIMEM[\\"Greenwich\\",0,\n        → ANGLEUNIT[\\"degree\\",0.0174532925199433]],\n        → ID[\\"EPSG\\",4326]],\n            CONVERSION[\\"UTM zone 30N\\",\n        → METHOD[\\"Transverse Mercator\\",\n        → ID[\\"EPSG\\",9807]],\n            PARAMETER[\\"Latitude of
        → natural origin\\",0,\n        → ANGLEUNIT[\\"degree\\",0.0174532925199433],\n        → ID[\\"EPSG\\",8801]],\n            PARAMETER[\\"Longitude of
        → natural origin\\",-3,\n        → ANGLEUNIT[\\"degree\\",0.0174532925199433],\n        → ID[\\"EPSG\\",8802]],\n            PARAMETER[\\"Scale factor
        → at natural origin\\",0.9996,\n        → SCALEUNIT[\\"unity\\",1],\n        → ID[\\"EPSG\\",8805]],\n            PARAMETER[\\"False
        → easting\\",500000,\n        → LENGTHUNIT[\\"metre\\",1],\n        → ID[\\"EPSG\\",8806]],\n            PARAMETER[\\"False
        → northing\\",0,\n            LENGTHUNIT[\\"metre\\",1],\n        → ID[\\"EPSG\\",8807]],\n            CS[Cartesian,2],\n        → AXIS[\\"(E)\\",east,\n            ORDER[1],\n        → LENGTHUNIT[\\"metre\\",1],\n        → AXIS[\\"(N)\\",north,\n            ORDER[2],\n        → LENGTHUNIT[\\"metre\\",1]],\n            USAGE[\n        → SCOPE[\\"Navigation and medium accuracy spatial
            reference\\"],\n            AREA[\\"Earth\\"],\n            GCS[\\"WGS 1984\\"]]\n        → ]\n      ]
    }
  }
}

```

```
}

      dataset[isMY,]$NDVI =
→ terra::extract(NDVI_rast,dataset[isMY,])[,2]
      }
    } else
      cat("No existe: ",YEAR,"-",MONTH,"\n")
  }
}

# Factores -----
# Codificación de las variables categóricas como factores:

dataset <- dataset |>
  mutate(enp = as.factor(enp),
         orientacion = cut(orientacion,
                           breaks =
                           → c(-Inf,-1,22.5,67.5,112.5,157.5,202.5,247.5,292.5,337.5,360),
                           labels =
                           → c("Plano","N","NE","E","SE","S","SW","W","NW","N")),
  WD10M = cut(WD10M,
               breaks =
               → c(0,22.5,67.5,112.5,157.5,202.5,247.5,292.5,337.5,360),
               labels =
               → c("N","NE","E","SE","S","SW","W","NW","N")),
  uso_suelo = uso_suelo |> as.character() |> str_sub(0,2) |>
  as.factor()
) %>%
select(-c(YEAR,MM,DD))

return(dataset)
}
```

Se usa la función definida para asignar las variables explicativas a la muestra generada:

```
# Se carga la muestra generada en el paso anterior:
load("salidas_intermedias/sample_strat_2024-04-26.RData")

# Se eliminan las observaciones que no tienen fecha pues se pueden usar
→ para el estudio
sample <- na.omit(sample)

# Se aplica la función a la muestra
dataset <- asignar_variables(sample)

# Se almacenan los resultados
```

```
save(dataset,
      file = paste0("salidas_intermedias/dataset_strat_completo",
                    Sys.Date(),".RData"))
```

B.3. Depuración de la muestra

En la propia función usada para generar la muestra ya se ajustaron los tipos de las variables y se codificaron adecuadamente las variables *WD10M* y *orientacion*. A continuación se estudian los casos faltantes y finalmente se decide eliminarlos.

```
# Librerías -----
# Se cargan las librerías que se usarán en esta sección
library(tidyverse) # Manipulación de datos
library(sf) # Vector data
library(terra) # Raster data
library(mapSpain) # Polígonos de regiones de España
library(magrittr) # Operador%<>%
library(skimr) # Resumen de datos

# Carga de los datos -----
# Se carga la muestra con todas las variables construida anteriormente
load("salidas_intermedias/dataset_strat_completo2024-04-26.RData")

# Codificación variable fire: -----
datos <- dataset |>
  mutate(fire = as.factor(fire))

# Análisis de valores perdidos: -----
datos %>% skimr()
dataset %>% apply(1,anyNA) %>% sum() # 200 registros con valores
→ perdidos

# Se observan:
#   8 nas en uso_suelo
#   32 nas en pendiente
#   36 nas en orientacion
#   32 nas en curvatura
#   85 nas en doblacion y en dist_poblacion
#   85 nas en NDVI

datos1 = datos %>% drop_na()

# Valores perdidos en variables topográficas -----
Andalucia <- esp_get_ccaa(ccaa = "Andalucía") # Se carga el mapa de
→ Andalucía
```

```
andalucia_proj <- st_transform(Andalucia,st_crs(dataset)) # Se proyecta
→ al sistema de referencia de los datos

# Mapa de valores perdidos en variables topográficas:
plot(st_geometry(andalucia_proj),reset=F)
datos |>
  filter(is.na(pendiente) | is.na(orientacion) | is.na(curvatura)) |>
  st_geometry() |>
  plot(pch=16,col="red",add=T) # Se encuentran en los límites de
→ Andalucía

datos |>
  st_drop_geometry() |>
  filter(is.na(pendiente) | is.na(orientacion) | is.na(curvatura)) |>
  count() # 53 registros con algún valor perdido entre las variables
→ topográficas

# Valores perdidos en variables demográficas -----
# Se observa que los valores perdidos se deben a que los datos no están
→ disponibles en el conjunto de datos disponible en el INE
datos |>
  st_drop_geometry() |>
  filter(is.na(poblacion)) |>
  mutate(Año = year(date)) |>
  select(Año,municipio,cod_municipio)

# Ejemplo:
pob =
  → read_csv2("data_raw/antropologicas/Población/poblacion_municipios.txt") [,c(1:5)]
  →
pob |> filter(CODIGO_INE3 == "18077") # Datos no disponibles

datos |>
  st_drop_geometry() |>
  filter(is.na(poblacion)) |>
  count() # 85 registros con valores perdidos en las variables
→ demográficas

# Valores perdidos en uso_suelo -----
dataset |> filter(is.na(uso_suelo)) %>% count() # 8 registros con
→ valores perdidos en la variable uso_suelo

dataset |> filter(is.na(uso_suelo)) |> st_geometry() |>
  → plot(pch=16,col="red",add=T)
# De nuevo se observa que se encuentran en los límites de Andalucía con
→ la costa
```

```

# Valores perdidos en NDVI -----
# Se muestran en el mapa
plot(st_geometry(andalucia_proj),reset=F)
dataset |> filter(is.na(NDVI)) |> st_geometry() |>
  ~ plot(pch=16,col="red",add=T)

# Gráfico de NDVI medio cada mes
NDVI_mes = datos |>
  st_drop_geometry() |>
  mutate(mes = month(date),
         año = year(date)) |>
  group_by(año,mes) |>
  summarise(NDVI_mes = mean(na.omit(NDVI))) |>
  ungroup() |>
  mutate(date = dmy(paste(01,mes,año,sep="/")))

NDVI_mes |>
  ggplot(aes(x=date,y=NDVI_mes)) +
  geom_line()+
  scale_x_date(date_breaks = "6 month",date_labels = "%y %b")+
  theme(axis.text.x = element_text(angle = 90))

# 2008 tiene un NDVI especialmente bajo, puede deberse a numerosos
~ factores:
#   - Que efectivamente sea un año especialmente seco
#   - Las características de la muestra seleccionada ese año
#   - Errores de medición
#   - ...

# La mayor parte de los valores perdidos en la variable uso_suelo son
~ debidos a que el archivo tiff para ese mes no está disponible en la
~ REDIAM. Los meses de los que no se dispone del ráster NDVI son:
~ "2003-01", "2003-04", "2017-02", "2018-11", "2020-11", "2021-12",
~ "2022-03" y "2022-12".

# Eliminación registros con valores perdidos -----
# Tras explorar otras opciones finalmente se opta por eliminar los
~ registros
# que contienen valroes perdidos:
datos = datos %>% drop_na()

# Almacenamiento del conjunto de datos depurados -----
# Se guarda el conjunto de datos depurados
save(datos,
      file = paste0("salidas_intermedias/datos_strat_depurados_geom_",

```

```
Sys.Date(),".RData"))
```

B.4. Análisis exploratorio

```
# Librerías -----
# Se cargan las librerías que se usarán en esta sección
library(tidyverse) # Manipulación de datos
library(sf) # Vector data
library(terra) # Raster data
library(mapSpain) # Polígonos de regiones de España
library(magrittr) # Operador %>%
library(skimr) # Resumen de datos
library(corrplot) # Gráfico de correlaciones
library(GGally) # Coordenadas paralelas
library(ggpubr) # Función ggarrange

# Carga de los datos -----
load("salidas_intermedias/datos_strat_depurados_geom_2024-05-03.RData")

# Resumen numérico -----
datos %>% st_drop_geometry %>% skim(.data_name = "datos")
```

B.4.1. Variable objetivo

```
# Distribución temporal -----
# Mensual:
g_mes <- datos %>%
  st_drop_geometry() %>%
  # filter(fire==1) %>%
  count(month(date,label=T),fire) %>%
  rename("mes" = "month(date, label = T)") %>%
  ggplot(aes(x = mes,y = n,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  # theme(axis.text.x = element_text(angle = 90)) +
  scale_fill_hue(direction = -1) +
  theme_minimal() +
  xlab("Mes") +
  ylab("Número de observaciones")

g_mes

# Anual
```

```

g_year <- datos %>%
  st_drop_geometry() %>%
  # filter(fire==1) %>%
  count(year(date),fire) %>%
  rename("año" = "year(date)") %>%
  ggplot(aes(x = año,y = n,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  scale_x_continuous(breaks=2002:2022) +
  geom_smooth(se=F,aes(color=fire),alpha=0.1) +
  scale_color_manual(values=c("darkblue","darkred")) +
  scale_fill_hue(direction = -1) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("Año") +
  ylab("Número de observaciones")
g_year

# En 2007 no se han generado observaciones positivas
# Causa: En el shapefile que contiene los incendios producidos en 2007
→ solo hay 4 observaciones y ninguna tenía la fecha, por lo que estas
→ no han podido utilizarse

# Valores muy bajos en 2008 y 2010, revisar
# 2010: correcto, en el archivo con los polígonos de incendios en 2010
→ solo hay 26 observaciones
# 2008: en el archivo relativo a ese año hay 37 observaciones, pero 11
→ de ellas no tienen la fecha de inicio, por lo que no se han podido
→ utilizar

# Día de la semana
g_dia <- datos %>%
  st_drop_geometry() %>%
  # filter(fire==1) %>%
  count(weekdays(date),fire) %>%
  rename("dia" = "weekdays(date)") %>%
  ggplot(aes(x = dia,y = n,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  scale_x_discrete(limits=c("lunes", "martes", "miércoles", "jueves",
  → "viernes", "sábado", "domingo")) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=0.5)) +
  # geom_smooth(se=F) +
  scale_fill_hue(direction = -1) +
  theme_minimal() +
  xlab("Día") +
  ylab("Número de observaciones")
g_dia

```

```
# Para mostrar los tres gráficos juntos
ggarrange(g_dia,g_mes,g_year,ncol=1,common.legend = T,legend = "bottom")

# Distribución temporal -----
# Se cargan los polígonos de las provincias
and <- esp_get_ccaa(ccaa = "Andalucía") %>% st_transform(st_crs(datos))
prov <- esp_get_prov() %>% filter(nuts2.name=="Andalucía") %>%
  st_transform(st_crs(datos))

# Casos positivos
g1 = ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos %>% filter(fire==1),size = 1, alpha =
    ~ 0.4,col="red")+
  ggtitle("Distribución espacial de casos positivos") +
  theme_minimal()

# Casos negativos
g0 = ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos %>% filter(fire==0),size = 1, alpha =
    ~ 0.4,col="blue") +
  ggtitle("Distribución espacial de casos negativos") +
  theme_minimal()

# Ambos gráficos juntos:
ggarrange(g1,g0,nrow=2)
```

B.4.2. Análisis univariantes

T2M

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=T2M),size = 1.2, alpha = 0.6) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de T2M por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

Las temperaturas más elevadas se encuentran en el interior.

```
datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_T2M = mean(T2M),
            Mes = `month(date, label = T)` %>%
    ggplot(aes(x=Mes,y=AVG_T2M,fill=fire)) +
    geom_col(position="dodge",alpha=0.8) +
    scale_fill_hue(direction = -1) +
    theme_minimal()
  # + theme(axis.text.x = element_text(angle = 45, vjust = 0.5,
  #   hjust=0.5))
```

En prácticamente todos los meses, la media de temperaturas en las observaciones positivas es mayor que en las observaciones negativas. Las temperaturas son más altas en los meses de verano, como cabía esperar.

RH2M

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=RH2M),size = 1.2, alpha = 0.6) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=F)) +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de RH2M por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

La humedad del aire es mayor en las de costa. Las zonas más secas se encuentran en las zonas de interior del norte de Andalucía.

```
datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_RH2M = mean(RH2M),
            Mes = `month(date, label = T)` %>%
```

```
ggplot(aes(x=Mes,y=AVG_RH2M,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  scale_fill_hue(direction = -1) +
  theme_minimal()
# + theme(axis.text.x = element_text(angle = 45, vjust = 0.5,
#   hjust=0.5))
```

En prácticamente todos los meses, la humedad del aire media entre las observaciones positivas es menor que en las observaciones negativas. La humedad del aire disminuye significativamente en verano.

GWETTOP

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=GWETTOP),size = 1, alpha = 0.6) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=F)) +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de GWETTOP por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

Las zonas con mayor humedad superficial de suelo son las que se encuentran en la costa mediterránea. Las zonas con un suelo más seco son las que se encuentran al norte de la cuenca el Guadalquivir, destacando la provincia de Huelva (hay que tener en cuenta que también es una zona en la que hay muchos incendios, la mayoría de los cuales se produce en verano, por lo que hay una gran concentración de observaciones durante el periodo estival, lo que podría influir en el hecho de que se observen tantas observaciones con valores tan bajos de la humedad superficial de suelo).

```
datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_GWETTOP = mean(GWETTOP),
            Mes = `month(date, label = T)` %>%
    ggplot(aes(x=Mes,y=AVG_GWETTOP,fill=fire)) +
    geom_col(position="dodge",alpha=0.8) +
    scale_fill_hue(direction = -1) +
    theme_minimal()
```

En general, la humedad superficial de suelo media entre los casos positivos es inferior que entre los casos negativos. Esto es especialmente evidente entre los meses de primavera y otoño. En los meses de invierno esto no está tan claro, incluso parece revertirse la tendencia.

WS10M

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos,
    ↪ aes(color=WS10M, alpha=WS10M, size=as.numeric(ifelse(WS10M<7.5,1,1.2))))
  ↪ +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  facet_wrap(~month(date,label=TRUE))+
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de WS10M por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  guides(size="none",alpha="none")
```

Los valores más elevados de la velocidad del viento a 10m de altura se dan en las zonas costeras.

```
datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_WS10M = mean(WS10M),
            Mes = `month(date, label = T)` %>%
  ggplot(aes(x=Mes,y=AVG_WS10M,fill=fire)) +
  geom_col(position="dodge",alpha=0.8)+ 
  scale_fill_hue(direction = -1) +
  theme_minimal()
```

En todos los meses, los valores medios de la velocidad del viento a 10m sobre la superficie son significativamente mayores en las observaciones positivas.

WD10M

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=WD10M),size = 1.2, alpha = 0.6) +
  facet_wrap(~month(date,label=TRUE))+
  # scale_color_stepsn(colours = rainbowrainbpw(8,rev=T)) +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de WD10M por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  guides(color = guide_legend(override.aes = list(size = 5))) +
  scale_color_viridis_d(option="turbo")

# Quisiera tener la dirección mayoritaria por mes y si hay incendio o no
←
mode <- function(x) { names(which.max(table(x))) }

datos_mode_WD10M = datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(mode_WD10M = mode(WD10M))

tibble(mes = datos_mode_WD10M[datos_mode_WD10M$fire==1,1],
       fire1 = datos_mode_WD10M[datos_mode_WD10M$fire==1,3],
       fire0 = datos_mode_WD10M[datos_mode_WD10M$fire==0,3])

# No parece aportar ninguna información relevante
```

PRECTOTCORR

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos,
  →   aes(color=PRECTOTCORR,size=PRECTOTCORR,alpha=PRECTOTCORR)) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=F)) +
  guides(size="none") +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de PRECTOTCORR por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
```

```

axis.ticks.x=element_blank(),
axis.text.y=element_blank(),
axis.ticks.y=element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank()) +
guides(alpha="none")

```

En la gran mayoría de las observaciones no se han observado precipitaciones. Se observan algunos valores positivos de precipitaciones distribuidos por el territorio Andaluz, alcanzándose los máximos en observaciones localizadas ne las cordilleras béticas.

```

datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_PRECTOTCORR = mean(PRECTOTCORR),
            Mes = `month(date, label = T)` %>%
  ggplot(aes(x=Mes,y=AVG_PRECTOTCORR,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  scale_fill_hue(direction = -1) +
  theme_minimal()

```

Se observa una clara diferencia en la media mensual de precipitaciones en ambas clases en todos los meses, siendo mucho mayores entre las observaciones negativas. En los meses de verano esto sigue siendo cierto, si bien las diferencias se suavizan significativamente, ya que en ambos clases las precipitaciones son reducidas.

NDVI

```

ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=NDVI),size = 1.2, alpha = 0.6) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de NDVI por mes") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

```

Los valores más elevados del NDVI se encuentran en el norte de la comunidad (Sierra Morena y Sierra de Aracena) y en la zona sur-centro. Los valores son especialmente

elevados en la Sierra de Grazalema y en las marismas del Guadalquivir y especialmente bajos en el este de la comunidad (Almería).

```
datos %>%
  st_drop_geometry() %>%
  group_by(month(date,label=T),fire) %>%
  summarize(AVG_NDVI = mean(NDVI),
            Mes = `month(date, label = T)` %>%
  ggplot(aes(x=Mes,y=AVG_NDVI,fill=fire)) +
  geom_col(position="dodge",alpha=0.8) +
  scale_fill_hue(direction = -1) +
  theme_minimal()
# theme(axis.text.x = element_text(angle = 45, vjust = 0.5,
#   hjust=0.5))
```

El valor del NDVI disminuye en los meses de verano y alcanza su máximo en los meses de invierno. No se observa claramente una relación entre el NDVI y la clase de la observación en la muestra.

poblacion

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=poblacion,size =
    ~ poblacion,alpha=poblacion/median(poblacion))) +
  scale_color_gradient(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  # scale_color_gradient(low="blue", high="red")+
  ggtitle("Distribución espacial de poblacion") +
  theme_minimal()

by(datos$poblacion,datos$fire,summary)
```

En la gran mayoría del territorio se observan niveles de población inferiores a 20.000 habitantes por municipio. Destacan las capitales de provincia por tener valores significativamente más elevados.

dens_poblacion

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dens_poblacion, size= dens_poblacion,
    alpha=dens_poblacion)) +
  scale_color_gradient(colours = rainbow(5,rev=T)) +
```

```

guides(size="none",alpha="none") +
# scale_color_gradient(low="blue", high="red")+
ggtitle("Distribución espacial de dens_poblacion") +
theme_minimal()

by(datos$dens_poblacion,datos$fire,summary)

```

Comentario similar a la población. Mayor densidad de población en las zonas de costa y en las capitales de provincia.

pendiente

```

ggplot(data = prov) +
geom_sf() +
geom_sf(data = datos, aes(color=pendiente, size= pendiente,
                           alpha=pendiente)) +
scale_color_gradientn(colours = rainbow(5,rev=T)) +
guides(size="none",alpha="none") +
ggtitle("Distribución espacial de pendiente") +
theme_minimal()

```

elevacion

```

ggplot(data = prov) +
geom_sf() +
geom_sf(data = datos, aes(color=elevacion, size= elevacion,
                           alpha=elevacion)) +
scale_color_gradientn(colours = rainbow(5,rev=T)) +
guides(size="none",alpha="none") +
ggtitle("Distribución espacial de elevacion") +
theme_minimal()

```

curvatura

```

ggplot(data = prov) +
geom_sf() +
geom_sf(data = datos, aes(color=curvatura, alpha=curvatura)) +
scale_color_gradientn(colours = rainbow(5,rev=T)) +
guides(size="none",alpha="none") +
ggtitle("Distribución espacial de curvatura") +
theme_minimal()

```

dist_carretera

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_carretera, alpha=dist_carretera))
  → +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_carretera") +
  theme_minimal()
```

dist_electr

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_electr, alpha=dist_electr)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_electr") +
  theme_minimal()
```

dist_camino

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_camino, alpha=dist_camino)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_camino") +
  theme_minimal()
```

dist_sendero

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_sendero, alpha=dist_sendero)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_sendero") +
  theme_minimal()
```

dist_poblacion

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_poblacion, alpha=dist_poblacion))
  → +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_poblacion") +
  theme_minimal()
```

dist_ferrocarril

```
ggplot(data = prov) +
  geom_sf() +
  geom_sf(data = datos, aes(color=dist_ferrocarril,
  → alpha=dist_ferrocarril)) +
  scale_color_gradientn(colours = rainbow(5,rev=T)) +
  guides(size="none",alpha="none") +
  ggtitle("Distribución espacial de dist_ferrocarril") +
  theme_minimal()
```

B.4.3. Análisis multivariantes

Variables numéricas

```
datos_numeric = datos %>%
  select(where(is.numeric)) %>%
  st_drop_geometry()

# Correlaciones -----
R = cor(datos_numeric)
corrplot(R,method = "ellipse",type = "lower")
summary(R-diag(diag(R)))

# Las variables más correlacionadas en la muestra son T2M con RH2M
→ (negativamente, -0.71), T2M con GWETTOP (negativamente, -0.69) y
→ GWETTOP con R2HM (positivamente, 0.68). Tiene sentido que la humedad
→ del aire a dos metros esté correlacionada positivamente con la
→ humedad del suelo y que ambas estén negativamente correlacionadas
→ con la temperatura del aire. También están correladas la población
→ con la densidad de población (positivamente, 0.63).
```

```
# Gráfico de coordenadas paralelas -----
datos |>
  select(fire,where(is.numeric)) |>
  ggparcoord(columns=2:19,alphaLines=0.1,groupColumn = "fire") +
  xlab('') +
  ylab('') +
  scale_x_discrete(labels=colnames(datos_numeric)) +
  scale_color_hue(direction = -1) +
  # guides(color ="none")+
  theme_minimal() +
  ggtitle("Tipificación a normal estándar (por defecto)") +
  theme(plot.title = element_text(size = 9),
        axis.text.x = element_text(angle = 90))

# Boxplots -----
boxplots <- map(names(datos_numeric), ~ ggplot(datos, aes(x = fire, y =
  .data[[.x]],fill=fire)) +
  geom_boxplot() +
  scale_fill_hue(direction = -1) +
  guides(fill="none") +
  labs(title = paste(.x, "~ fire")))
ggarrange(plotlist = boxplots,ncol=6,nrow=3)

# PCA -----
pca <- princomp(datos_numeric,cor=T)
summary(pca)
# Se necesitan al menos 11 componentes principales para explicar el 80%
# de la variabilidad de las 18 variables numéricas de la muestra
# (tipificadas) y al menos 14 para explicar el 90%. Esto refleja, la
# complejidad del conjunto de datos.
```

Variables categóricas

```
datos_categ <- datos %>%
  select(WD10M,orientacion,emp,uso_suelo) %>%
  st_drop_geometry()

# Histogramas -----
histogramas <- map(names(datos_categ),
  ~ ggplot(datos, aes(x = .data[[.x]],
    fill = fire)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  scale_fill_hue(direction = - 1) +
  theme_minimal() +
  labs(title = .x)+

  theme(axis.text.x = element_text(size=7)))
```

```
ggarrange(plotlist = histogramas, ncol = 2, nrow =
  ceiling(length(histogramas)/2))
```

B.5. Modelos

```
# Librerías -----
# Se cargan las librerías que se usarán en esta sección
library(tidyverse) # Manipulación de datos
library(sf) # Vector data
library(tidymodels) # Ecosistema para la construcción de modelos
library(akima) # Función interp
library(magrittr) # Operador %>%>%
library(ggpubr) # Función ggarrange
library(knitr) # Función kable

# Carga de datos -----
load("salidas_intermedias/datos_strat_depurados_geom_2024-05-03.RData")

# Agrupación clases uso_suelo -----
# Nos quedamos con los 7 niveles del factor más frecuentes (clases 2 y
# 3)
datos <- datos |>
  mutate(uso_suelo = fct_lump(uso_suelo,
    n = 7,
    other_level= "Otro"))

# Funciones para la evaluación de modelos -----
# Función para obtener las medidas de rendimiento de los modelos a
# partir de un objeto predict
get_metrics <- function(pred) {
  list(
    res = tibble(
      roc_auc = pred |> roc_auc(truth = fire, .pred_0) |>
        pull(.estimate),
      accuracy = pred |> accuracy(truth = fire, .pred_class) |>
        pull(.estimate),
      recall = pred |> sensitivity(truth = fire,
        .pred_class, event_level="second") |> pull(.estimate),
      specificity = pred |> spec(truth = fire,
        .pred_class, event_level="second") |> pull(.estimate),
      precision = pred |> precision(truth = fire,
        .pred_class, event_level="second") |> pull(.estimate)),
    conf_mat = pred |> conf_mat(truth = fire, .pred_class)))
}
```

```
# Función para mostrar gráficamente los resultados del tuning de un
→ modelo con dos parámetros
tuning_plot = function(mod_res) {
  datos_metrics = mod_res %>%
    collect_metrics()

  plots = list()

  for (metric in unique(datos_metrics$metric)) {

    datos = datos_metrics %>%
      filter(.metric==metric)

    # Interpolar los datos faltantes
    datos_interp <- interp(datos[[1]], datos[[2]], datos$mean)

    # Crear un nuevo dataframe con los datos interpolados
    datos_interp_df <- data.frame(
      expand.grid(x = datos_interp$x, y = datos_interp$y), z =
      → as.vector(datos_interp$z))

    # Crear el gráfico de mapa de calor con interpolación
    p = ggplot(datos_interp_df, aes(x = x, y = y, fill = z)) +
      geom_tile() +
      scale_fill_viridis_c(option = "turbo", name = NULL, na.value =
      → "transparent")+
      labs(title = "",
           x = colnames(datos)[1],
           y = colnames(datos)[2],
           fill = metric) +
      theme_minimal()

    plots[[metric]] = p
  }
  ggarrange(plotlist = plots,
            labels=c("Accuracy", "Specificity", "ROC-AUC", "Recall"),
            align = "hv")
}
```

B.5.1. Partición temporal entrenamiento / validación / test

```
set.seed(123)

splits = initial_validation_time_split(datos,
```

```

prop=c(0.6,0.2))

training <- training(splits) %>% st_drop_geometry()
val_set <- validation_set(splits) %>% st_drop_geometry()
test <- testing(splits) %>% st_drop_geometry()

```

B.5.2. Regresión logística con penalización

```

# 1º Definimos el modelo:
lr_mod <-
  logistic_reg(penalty = tune(), mixture = tune()) %>%
  set_engine("glmnet")

# 2º Creamos la receta
lr_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date, features = c("dow", "month")) %>%
  # step_holiday(date, holidays = holidays) %>%
  step_rm(date, cod_municipio, municipio) %>% # Se eliminan variables
  → identificadoras
  step_dummy(all_nominal_predictors()) %>% # Se crean variables dummy
  → para los factores
  step_lincomb() %>% # Elimina variables con dependencia lineal exacta
  step_corr() %>% # Elimina variables con correlación superior a 0.9
  step_zv(all_predictors()) %>% # Eliminar variables con varianza nula
  step_normalize(all_predictors()) # Se normalizan todos los predictores

# 3º Creamos el workflow
lr_workflow <-
  workflow() %>%
  add_model(lr_mod) %>%
  add_recipe(lr_recipe)

# 4º Creamos el grid para los parámetros
lr_reg_grid <- expand_grid(penalty = 10^seq(-4, -1, length.out = 10),
                           mixture = seq(0,1,length.out=10))

# 5º Ajustamos el modelo
lr_res <-
  lr_workflow %>%
  tune_grid(val_set,
            grid = lr_reg_grid,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy, roc_auc, recall, spec))

```

```
# 6º Evaluación de modelos
tuning_plot(lr_res)

lr_res |>
  collect_metrics() |>
  group_by(.metric) |>
  mutate(.metric = ifelse(.metric == "recall","spec",
                        ifelse(.metric == "spec","recall",
                               .metric))) |>
  summarise(max = max(mean),min=min(mean))

# 7º Selección del mejor modelo
lr_best <-
  lr_res %>%
  select_best(metric="accuracy")
lr_best

# Extraer coeficientes
lr_workflow %>%
  finalize_workflow(lr_best) %>%
  fit(training) %>%
  extract_fit_parsnip() %>%
  tidy() %>%
  print(n=100)
```

B.5.3. Regresión logística con penalización + PCA

```
# 1º Creamos el modelo
lr_pca_mod <-
  logistic_reg(penalty = tune(), mixture = tune()) %>%
  set_engine("glmnet")

# 2º Creamos la receta
lr_pca_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date,features = c("dow","month")) %>%
  # step_holiday(date, holidays = holidays) %>%
  step_rm(date,cod_municipio,municipio) %>% # Se eliminan variables
  ~ identificadoras
  step_dummy(all_nominal_predictors()) %>% # Se crean variables dummy
  ~ para los factores
  step_lincomb() %>% # Elimina variables con dependencia lineal exacta
  step_corr() %>% # Elimina variables con correlación superior a 0.9
```

```

step_zv(all_predictors()) %>% # Eliminar variables con varianza nula
step_normalize(all_predictors()) %>% # Se normalizan todos los
  → predictores
step_pca(all_numeric_predictors(), num_comp = tune())

# 3º Creamos el workflow
lr_pca_workflow <-
  workflow() %>%
  add_model(lr_pca_mod) %>%
  add_recipe(lr_pca_recipe)

# 4º Creamos el grid para los parámetros
lr_pca_reg_grid <- expand_grid(penalty = 10^seq(-4, -1, length.out = 10),
                                mixture = seq(0,1,length.out=10),
                                num_comp = c(20,25,30,35,40,45,50))

# 5º Ajustamos el modelo
lr_pca_res <-
  lr_pca_workflow %>%
  tune_grid(val_set,
            grid = lr_pca_reg_grid,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy,roc_auc,recall,spec))

# 6º Evaluación modelos
lr_pca_tuning = lr_pca_res |>
  collect_metrics() |>
  group_by(.metric) |>
  summarise(max = max(mean),min=min(mean))

# 7º Selección del mejor modelo
lr_pca_best <-
  lr_pca_res %>%
  select_best(metric="accuracy")
lr_pca_best

```

B.5.4. Árboles de decisión

```

# 1º Creamos el modelo
dt_mod <-
  decision_tree(cost_complexity = tune()) %>%
  set_engine("rpart") %>%
  set_mode("classification")

# 2º Creamos la receta

```

```
dt_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date, features = c("dow", "month")) %>%
  # step_holiday(date) %>%
  step_rm(date, cod_municipio, municipio)

# 3º Creamos el workflow
dt_workflow <-
  workflow() %>%
  add_model(dt_mod) %>%
  add_recipe(dt_recipe)

# 4º Se ajusta el modelo
set.seed(345)
dt_res <-
  dt_workflow %>%
  tune_grid(val_set,
            grid = 10,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy,roc_auc,recall,spec))

# 5º Se evalúan los modelos obtenidos
dt_res |>
  collect_metrics() |>
  group_by(.metric) |>
  mutate(.metric = ifelse(.metric == "recall","spec",
                        ifelse(.metric == "spec","recall",
                               .metric))) |>
  summarise(max = max(mean),min=min(mean))

# Gráfico del ajuste
dt_res %>%
  collect_metrics() %>%
  mutate(.metric = ifelse(.metric == "recall","spec",
                        ifelse(.metric == "spec","recall",
                               .metric))) |>
  ggplot(aes(x = cost_complexity, y = mean,col=.metric)) +
  geom_point() +
  geom_line() +
  ylab("") +
  scale_x_log10(labels = scales::label_number())+
  theme_minimal()

# 6º Selección del mejor modelo
dt_best <- dt_res |>
  select_best(metric = "accuracy")
dt_best
```

B.5.5. Bosques aleatorios

```

# Detectar el número de núcleos para trabajar en paralelo
cores <- parallel::detectCores()
cores

# Construimos el modelo, especificando el número de núcleos a usar en la
→ computación en paralelo de forma que la computación sea más
→ eficiente

# ETAPA 1: fijado mtry=4, se ajusta min_n
# ----

# 1º Construir el modelo
rf_mod1 <-
  rand_forest(mtry = 4, min_n = tune(), trees = 1000) %>%
  set_engine("ranger", num.threads = cores) %>%
  set_mode("classification")

# 2º Construir la receta con el preprocesamiento
rf_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date, features = c("dow", "month")) %>%
  # step_holiday(date) %>%
  step_rm(date, cod_municipio, municipio)
# No normalizamos en este caso pues no es necesario

# 3º Ensamblar todo con workflow
rf_workflow1 <-
  workflow() %>%
  add_model(rf_mod1) %>%
  add_recipe(rf_recipe)

# 4º Train and tune
set.seed(345)

rf_res1 <-
  rf_workflow %>%
  tune_grid(val_set,
            grid = expand_grid(min_n = seq(1000, 2500, 100)),
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy, roc_auc, recall, spec))

# Resultados del tuning

rf_tuning1 <- rf_res1 |>
  collect_metrics() |>

```

```
group_by(.metric) |>
  summarise(max = max(mean), min=min(mean))

rf_tuning1

# plot

rf_plot1 <-
  rf_res1 %>%
  collect_metrics() %>%
  mutate(.metric = ifelse(.metric == "recall", "spec",
                        ifelse(.metric == "spec", "recall",
                               .metric))) %>%
  ggplot(aes(x = min_n, y = mean, col=.metric)) +
  geom_point() +
  geom_line() +
  ylab("") +
  theme_minimal() +
  labs(title = "Etapa 1\nFijado mtry = 4, se ajusta min_n")

# Mejor modelo
rf_best1 <-
  rf_res1 %>%
  select_best(metric = "spec")

rf_best1

rf_metrics1 <- rf_res1 |>
  collect_predictions(parameters = rf_best1) |>
  get_metrics()

rf_metrics1

# ETAPA 2: fijando min_n de la etapa anterior, se ajusta mtry
# ----

# 1º Se construye el modelo
rf_mod2 <-
  rand_forest(mtry = tune(), min_n = rf_best1$min_n, trees = 1000) %>%
  set_engine("ranger", num.threads = cores) %>%
  set_mode("classification")

# 2º Se usa la misma receta que antes

# 3º Ensamblar todo con workflow
```

```

rf_workflow2 <-
  workflow() %>%
  add_model(rf_mod2) %>%
  add_recipe(rf_recipe)

# 4º Train and tune
set.seed(345)

rf_res2 <-
  rf_workflow2 %>%
  tune_grid(val_set,
            grid = expand_grid(mtry = 1:10),
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy,roc_auc,recall,spec))

# Resultados del tuning

rf_tuning2 <- rf_res2 |>
  collect_metrics() |>
  group_by(.metric) |>
  summarise(max = max(mean),min=min(mean))

rf_tuning2

#plot
rf_plot2 <-
  rf_res2 %>%
  collect_metrics() %>%
  mutate(.metric = ifelse(.metric == "recall","spec",
                         ifelse(.metric == "spec","recall",
                                .metric))) %>%
  ggplot(aes(x = mtry, y = mean,col=.metric)) +
  geom_point() +
  geom_line() +
  ylab("") +
  theme_minimal()+
  labs(title = paste0("Etapa 2\nFijado min_n = ", rf_best1$min_n, " se
                     → ajusta mtry"))

# Mejor modelo
rf_best2 <-
  rf_res2 %>%
  select_best(metric = "spec")
rf_best2

rf_metrics2 <- rf_res2 |>

```

```
collect_predictions(parameters = rf_best2) |>
get_metrics()

rf_metrics2

# -----

# Plots
ggarrange(rf_plot1,rf_plot2,nrow=1,common.legend = T,legend = "bottom")
```

B.5.6. k-Nearest Neighbours

```
# 1º Definimos el modelo:
knn_mod <-
  nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>%
  set_mode("classification")

# 2º Creamos la receta
knn_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date,features = c("dow","month")) %>%
  # step_holiday(date, holidays = holidays) %>%
  step_rm(date,cod_municipio,municipio) %>% # Se eliminan variables
  → identificadoras
  step_dummy(all_nominal_predictors()) %>% # Se crean variables dummy
  → para los factores
  step_lincomb() %>% # Elimina variables con dependencia lineal exacta
  step_corr() %>% # Elimina variables con correlación superior a 0.9
  step_zv(all_predictors()) %>% # Eliminar variables con varianza nula
  step_normalize(all_predictors()) # Se normalizan todos los predictores

# 3º Creamos el workflow
knn_workflow <-
  workflow() %>%
  add_model(knn_mod) %>%
  add_recipe(knn_recipe)

# 4º Train and tune
set.seed(345)
knn_res <-
  knn_workflow %>%
  # fit(training)
  tune_grid(val_set,
```

```

grid = expand_grid(neighbors = c(1,10,25,seq(25,400,25))),  

control = control_grid(save_pred = TRUE),  

metrics = metric_set(accuracy,roc_auc,recall,spec))

# 5º Evaluació de los modelos
knn_res |>  

collect_metrics() |>  

group_by(.metric) |>  

summarise(max = max(mean),min=min(mean))

knn_res %>%
  collect_metrics() %>%
  mutate(.metric = ifelse(.metric == "recall","spec",
                         ifelse(.metric == "spec","recall",
                                .metric))) %>%
# filter(.metric == "accuracy") %>%
ggplot(aes(x = neighbors, y = mean,col=.metric)) +  

geom_point() +  

geom_line() +  

ylab("") +  

theme_minimal()

# 6º Selección del mejor modelo
knn_best <- knn_res |>  

select_best(metric = "accuracy")
knn_best

```

B.5.7. SVM lineal

```

# 1º Construir el modelo
svm_mod <-
  svm_linear(cost = tune()) %>%
  set_engine("kernlab") %>%
  set_mode("classification")

# 2º Construir la receta con el preprocesamiento
svm_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date,features = c("dow", "month")) %>%
  step_rm(date, cod_municipio, municipio) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())

# 3º Ensamblar todo con workflow

```

```
svm_workflow <-
  workflow() %>%
  add_model(svm_mod) %>%
  add_recipe(svm_recipe)

# 4º Train and tune
set.seed(345)
svm_res <-
  svm_workflow %>%
  tune_grid(val_set,
            grid = 15,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy,roc_auc,recall,spec))

# 5º Evaluación de resultados del tuning
svm_res |>
  collect_metrics() |>
  group_by(.metric) |>
  summarise(max = max(mean),min=min(mean))

svm_plot <-
  svm_res %>%
  collect_metrics() %>%
  mutate(.metric = ifelse(.metric == "recall","spec",
                        ifelse(.metric == "spec","recall",
                               .metric))) %>%
  # filter(.metric == "accuracy")%>%
  ggplot(aes(x = cost, y = mean,col=.metric)) +
  geom_point() +
  geom_line() +
  ylab("") +
  theme_minimal()
svm_plot

# 6º Selección del mejor modelo
svm_best <-
  svm_res %>%
  select_best(metric="accuracy")
svm_best
```

B.5.8. SVM radial

```
# 1º Construir el modelo
svm_rbf_mod <-
  svm_rbf(cost = tune(),rbf_sigma = tune()) %>%
```

```

set_engine("kernlab") %>%
set_mode("classification")

# 2º Construir la receta con el preprocesamiento
svm_rbf_recipe <-
  recipe(fire ~ ., data = training) %>%
  step_date(date, features = c("dow", "month")) %>%
  step_rm(date, cod_municipio, municipio) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_lincomb() %>% # Elimina variables con dependencia lineal exacta
  step_corr() %>% # Elimina variables con correlación superior a 0.9
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())

# 3º Ensamblar todo con workflow
svm_rbf_workflow <-
  workflow() %>%
  add_model(svm_rbf_mod) %>%
  add_recipe(svm_rbf_recipe)

# 4º Train and tune
set.seed(345)
svm_rbf_res <-
  svm_rbf_workflow %>%
  tune_grid(val_set,
            grid = 8,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(accuracy,roc_auc,recall,spec))

# 5º Evaluación de resultados del tuning
svm_rbf_res |>
  collect_metrics() |>
  group_by(.metric) |>
  summarise(max = max(mean),min=min(mean))

# 6º Selección del mejor modelo
svm_rbf_best <-
  svm_rbf_res %>%
  select_best(metric="accuracy")

svm_rbf_best

```

B.5.9. Comparación

```
models = tibble(model_name =
  c("lr","lr_pca","dt","rf","svm_linear","svm_rbf","knn"),
  models_tune = list(lr_res, lr_pca_res, dt_res, rf_res2,
  ↵  svm_res, svm_rbf_res, knn_res),
  models_workflow = list(lr_workflow, lr_pca_workflow,
  ↵  dt_workflow, rf_workflow2, svm_workflow,
  ↵  svm_rbf_workflow, knn_workflow))

# save(models, file="salidas_intermedias/all_models.RData")

load("salidas_intermedias/all_models.RData")
models = models %>%
  mutate(best_tuning = map(models_tune,
    function(x) select_best(x, metric =
      "accuracy")),
  best_metrics = map2(models_tune,
    best_tuning,
  ~ collect_predictions(.x,parameters = .y) %>%
  get_metrics() %>%
  extract2(1)), # Para extraer solo las medidas y no la matriz de
  ~ confusion
  roc = map2(models_tune,
    best_tuning,
  ~ collect_predictions(.x,parameters = .y) %>%
  ~ roc_curve(fire, .pred_0)))
)

# metricas
metrics = models %>%
  select(model_name,best_metrics) %>%
  unnest(best_metrics)
kable(metrics,digits=3)

# curva roc
metrics %>%
  pivot_longer(cols = c(roc_auc, accuracy, recall, specificity,
  ↵  precision),
  names_to = "metric") %>%
  ggplot(aes(x = metric, y = value, group = model_name)) +
  geom_line(aes(col = model_name),size=1) +
  geom_point(aes(col = model_name),size=2.3) +
  scale_color_viridis_d(option="turbo") +
  geom_vline(xintercept=1:5, linetype="dotted") +
  labs(col = "Modelo", title = "Métricas sobre validación") +
  theme_minimal() +
  theme(axis.line.x = element_line(color="black", size = 1),
  axis.line.y = element_line(color="black", size = 1))
```

```
# plot medidas
models %>% select(model_name,roc) %>% unnest(roc) %>%
  ggplot(aes(x = 1 - specificity, y = sensitivity, col = model_name)) +
  geom_path(lwd = 1, alpha = 0.7) +
  geom_abline(lty = 3) +
  coord_equal() +
  scale_color_viridis_d(option="turbo") +
  labs(color="Modelo")+
  # scale_color_viridis_d(option = "turbo",name="Modelo") +
  theme_minimal() +
  theme(axis.line.x = element_line(color="black", size = 1),
        axis.line.y = element_line(color="black", size = 1))+ 
  ggtitle("Curva ROC en validación")
```

B.5.10. Test

Se unen los conjuntos training y validation para entrenar el modelo final

```
set.seed(345)
models <- models %>%
  mutate(final_workflow = map2(models_workflow, best_tuning,
    ~ finalize_workflow),
    last_fit = map(final_workflow, function(x)
      ~ last_fit(x,splits,add_validation_set=T))) 

models = models %>%
  mutate(test_metrics = map(last_fit,
    ~collect_predictions(.x) %>%
      ~ get_metrics() %>%
      extract2(1)), # Para extraer solo las
      ~ medidas)

test_roc = map(last_fit,
  ~collect_predictions(.x) %>%
    ~ roc_curve(fire, .pred_0))
  ~ )

# save(models, file="salidas_intermedias/all_models_test.RData")

# metricas en test
test_metrics = models %>% select(model_name, test_metrics) %>%
  ~ unnest(test_metrics)
kable(test_metrics,digits=3)

# plot
test_metrics %>%
```

```
pivot_longer(cols = c(roc_auc, accuracy, recall, specificity,
→ precision),
             names_to = "metric") %>%
ggplot(aes(x = metric, y = value, group = model_name)) +
geom_line(aes(col = model_name), size=1) +
geom_point(aes(col = model_name), size=2.3) +
scale_color_viridis_d(option="turbo") +
geom_vline(xintercept=1:5, linetype="dotted") +
labs(col = "Modelo", title = "Métricas sobre test") +
theme_minimal() +
theme(axis.line.x = element_line(color="black", size = 1),
      axis.line.y = element_line(color="black", size = 1))

# roc
models %>% select(model_name,test_roc) %>% unnest(test_roc) %>%
  ggplot(aes(x = 1 - specificity, y = sensitivity, col = model_name)) +
  geom_path(lwd = 1, alpha = 0.7) +
  geom_abline(lty = 3) +
  coord_equal() +
  scale_color_viridis_d(option="turbo") +
  labs(color="Modelo")+
  # scale_color_viridis_d(option = "turbo",name="Modelo") +
  theme_minimal() +
  theme(axis.line.x = element_line(color="black", size = 1),
        axis.line.y = element_line(color="black", size = 1))+
  ggtile("Curva ROC en test")
```

Para calcular la medida de importancia de las variables en bosque aleatorio. No se hace desde el principio para que la computación sea más rápida.

```
# Se hace el last_fit manualmente:
cores = 8

# last model
last_rf_mod <- rand_forest(mtry = rf_best2$mtry, min_n = rf_best1$min_n,
→ trees = 1000) %>%
  set_engine("ranger", num.threads = cores, importance="impurity") %>%
  set_mode("classification")

# last workflow
last_rf_workflow <-
  rf_workflow2 %>%
  update_model(last_rf_mod)

# last fit
set.seed(345)
last_rf_fit <-
```

```

last_rf_workflow %>%
  last_fit(splits,
            add_validation_set = T)
# VIP
last_rf_fit %>%
  extract_fit_parsnip() %>%
  vip(num_features = 50,aesthetics = list(fill="lightblue")) +
  theme_minimal()

```

B.6. Aplicación de los modelos

```

# Librerías -----
# Se cargan las librerías que se usarán en esta sección
library(tidyverse) # Manipulación de datos
library(sf) # Vector data
library(tidymodels) # Ecosistema para la construcción de modelos
library(ggpubr) # Función ggarrange
library(knitr) # Función kable
library(skimr) # Función skim
library(terra)

# Carga de datos -----
load("salidas_intermedias/datos_strat_depurados_geom_2024-04-27.RData")

# Polígono de Andalucía -----
and <- esp_get_ccaa(ccaa = "Andalucía") %>%
  st_transform(st_crs(datos))

```

B.6.1. Visión general del desempeño del modelo

Se construye una rejilla de puntos con una separación de 10km entre ellos (en la dirección Norte-Sur y Este-Oeste). En cada uno de los puntos se predice la probabilidad de incendio el día 15 de cada mes usando los mejores modelos construidos. Para ello, primero hay que asignar a cada punto los valores correspondientes de las variables predictoras consideradas.

```

# grid de puntos 10km x 10km de andalucía
grid = st_make_grid(and,
                    cellsize = c(10000,10000), # 10000
                    what = "centers")[and]
g = and %>%
  ggplot() +
  geom_sf() +
  theme_minimal()

```

```
g +
  geom_sf(data = grid,color="red",size=0.6) +
  labs(title = "Malla de puntos con una resolución de 10km x 10km")
```

Se va a analizar cada observación el 15 de cada mes.

```
sample = NULL

for (m in 1:12) {
  if (is.null(sample)){
    sample <- tibble(date = rep(ymd(paste("2022",m,"15",sep="/"))),
                     geometry = grid) %>% st_sf()
  } else
    sample <- sample %>%
      bind_rows(tibble(date = rep(ymd(paste("2022",m,"15",sep="/"))),
                     geometry = grid))
}
```

Se le asocian todas las variables correspondientes a cada observación.

```
source("scripts/strat/fun_asignar_variables.R")
full_grid = asignar_variables(sample)
```

Se imputan los valores faltantes de NDVI asignándoles los del año anterior, ya que faltan los archivos de marzo y diciembre de 2022.

```
# La siguiente función lee el archivo con el NDVI correspondiente a un
# mes y a un año dados (si está disponible)

read_NDVI = function(MM,YYYY) {
  MM = str_pad(as.character(MM),2,"left",pad = "0")
  YY = substr(as.character(YYYY),3,4)
  if (as.numeric(YY)<=06) {
    ruta <-
    → paste0("data_raw/vegetacion/",YYYY,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIFF/TI")
  } else if (as.numeric(YY)<=11) {
    ruta <-
    → paste0("data_raw/vegetacion/",YYYY,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIF/TE")
  } else if (as.numeric(YY)<=21){
    ruta <-
    → paste0("data_raw/vegetacion/",YYYY,"TERMODMEDMNDVI/InfGeografica/InfRaster/TIFF/t")
  }else {
    ruta <-
    → paste0("data_raw/vegetacion/",YYYY,"TERMODMEDMNDVI/InfGeografica/InfRaster/COG/te")
  }
}
```

```

if (file.exists(ruta)) {
  NDVI = rast(ruta)
} else
  NDVI = NA
return(NDVI)
}

# Los meses para los cuales no está disponible el archivo de NDVI:
year_month_missing_NDVI = c(
  "2003-01",
  "2003-04",
  "2017-02",
  "2018-11",
  "2020-11",
  "2021-12",
  "2022-03",
  "2022-12")

# Para cada observación para la que no está disponible el NDVI (porque
→ la información de ese mes no está disponible), se obtiene el
→ correspondiente al mismo mes del año anterior, si este está
→ disponible, si no, el del año posterior:

missing_NDVI = full_grid |>
  filter(is.na(NDVI)) |>
  mutate(year = year(date), month = month(date)) |>
  group_by(year,month) |>
  filter(paste(year,str_pad(as.character(month),2,"left"),pad =
  → "0"),sep="-") %in% year_month_missing_NDVI) |> # Se filtra
  → porque también hay observaciones que tienen NA en el NDVI no
  → porque no existe el archivo, si no lo mismo que en ocasiones
  → anteriores, porque discrepancias entre los límites de los
  → polígonos
  nest() |>
  mutate(NDVI_rast = map2(month,year-1,read_NDVI), # Se lee primero el
  → del año anterior
    NDVI_rast =
      → ifelse(is.na(unlist(NDVI_rast)),map2(month,year+1,read_NDVI),NDVI_rast),
      → # Si no está disponible, se toma el del año posterior
    NDVI_nuevo = map2(NDVI_rast,data,~terra::extract(.x,.y)[,2])) |>
  select(-NDVI_rast) |>
  unnest(c(data,NDVI_nuevo)) |>
  mutate(NDVI = NDVI_nuevo,.keep="unused")

ind_modificados = is.na(full_grid$NDVI) &
  (paste(year(full_grid$date),str_pad(as.character(month(full_grid$date)),2,"left",
  → = "0"),sep="-")) %in% year_month_missing_NDVI) # Los elementos que
  → han sido modificados

```

```
# Se asignan los valores imputados del NDVI:  
full_grid[ind_modificados,]$NDVI = missing_NDVI$NDVI  
  
# Resumen  
datos %>% st_drop_geometry %>% skim(.data_name = "datos")
```

Se agrupan los niveles de uso de suelo

```
full_grid <- full_grid |>  
  mutate(uso_suelo = fct_other(uso_suelo,  
    keep =  
      c("21", "22", "23", "24", "31", "32", "33"),  
    other_level = "Otro"))  
  
# save(full_grid, file = full_grid_meses_2022_processed.RData")
```

Se usará el modelo de regresión logística lasso final,

```
load("Private/all_models_test.RData")  
  
model <- models %>% filter(model_name == "lr")  
# model <- models %>% filter(model_name == "sum_linear")  
# model <- models %>% filter(model_name == "rf")  
  
rm(models) # Es un archivo muy pesado, lo eliminamos de la memoria  
  
# Predicciones  
pred_class = model %>%  
  pull(last_fit) %>%  
  .[[1]] %>%  
  extract_workflow() %>%  
  predict(new_data = full_grid)  
pred_probs = model %>%  
  pull(last_fit) %>%  
  .[[1]] %>%  
  extract_workflow() %>%  
  predict(new_data = full_grid, type = "prob")  
pred = cbind(full_grid, pred_class, pred_probs)  
  
# Se cargan los incendios producidos en el año 2022  
incendios22 <-  
  st_read(paste0("./data_raw/incendios_2000-2022/incendios_", 2022, ".shp"))  
  %>%  
  st_transform(st_crs(full_grid)) %>%  
  mutate(date = ymd(fecha_inic))
```

Se muestra el gráfico con la estimación de la probabilidad de incendio el día 15 de cada mes en todos los puntos. Se añaden los incendios producidos en cada mes del año 2022

```
# Gráfico predicciones mes + Incendios producidos
ggplot(data = and) +
  geom_sf() +
  geom_sf(data = pred, aes(color=.pred_1),alpha=0.8,size = 1.5) +
  facet_wrap(~month(date,label=TRUE)) +
  scale_color_gradientn(colours = rainbow(5,rev=T),limits=c(0,1)) +
  # scale_color_gradient(low="blue", high="red")+
  guides(alpha = "none") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  geom_sf(data = incendios22 %>% st_centroid, color = "black", shape
  ← =24, size=1,fill = "red") +
  labs(title = "Probabilidad de incendio estimada el día 15 de cada mes
  ← de 2022",
       subtitle = paste0("Modelo: ",model$model_name),
       color = "Probabilidad\nde incendio\nestimada")
```

B.7. Caso de interés

Se va a usar el modelo para estudiar el incendio del 8 de septiembre de 2021 en Sierra Bermeja por ser de especial relevancia.

```
# Se carga la información del incendio
incendios21 <-
  → st_read(paste0("./data_raw/incendios_2000-2022/incendios_",2021,".shp"))
  → %>%
    st_transform(st_crs(datos)) %>%
    mutate(FECHA_INIC=ymd(FECHA_INIC))

incendio_estudio <- incendios21 %>% filter(month(FECHA_INIC)==9)
rm(incendios21)

# Mapa de Andalucía y de las provincias
and <- esp_get_ccaa(ccaa = "Andalucía") %>% st_transform(st_crs(datos))
prov <- esp_get_prov() %>% filter(nuts2.name=="Andalucía") %>%
  → st_transform(st_crs(datos))

# Gráfico del incendio
```

```
g = prov %>%
  ggplot() +
  geom_sf() +
  geom_sf(data = incendio_estudio, color="red",fill=alpha("red",0.2))+
  theme_minimal()+
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

# Ampliamos la zona del incendio
bbox = st_buffer(incendio_estudio,dist=10000) %>%
  st_bbox()

g1 = prov %>%
  ggplot() +
  geom_sf() +
  geom_sf(data = incendio_estudio, color="red",fill=alpha("red",0.2))+
  theme_minimal() +
  coord_sf(xlim=c(bbox$xmin,bbox$xmax),ylim=c(bbox$ymin,bbox$ymax))

# Gráfico final
gg <- ggarrange(g,g1)
annotate_figure(gg,
               top = text_grob("Caso de estudio: Incendio de Sierra
                           ↳ Bermeja\n8 de septiembre de 2021",size = 14))
```

Se procede de forma similar a la sección anterior. Se crea una rejilla de puntos con una separación de 1km, en la zona que rodea al incendio. En cada punto se predecirá el riesgo de incendio el día de inicio del incendio, 15 y 30 días antes y 15, 30 y 45 días después.

```
# Se crea el grid
grid = st_make_grid(st_buffer(incendio_estudio,dist=10000),
                    cellsize = c(1000,1000),
                    what = "centers") [and]

# Visualización
g +
  geom_sf(data = grid,size=0.7) +
  coord_sf(xlim=c(bbox$xmin,bbox$xmax),ylim=c(bbox$ymin,bbox$ymax))

# Se le asocian las fechas respectivas
sample = tibble(date =
  rep(incendio_estudio$FECHA_INIC,length(grid)),geometry = grid) %>% #
  el día del incendio
```

```

bind_rows(tibble(date =
  ↪ rep(incendio_estudio$FECHA_INIC+15,length(grid)),geometry
  ↪ = grid)) %>% # 15 días después
bind_rows(tibble(date =
  ↪ rep(incendio_estudio$FECHA_INIC+30,length(grid)),geometry
  ↪ = grid)) %>% # el mes después
bind_rows(tibble(date =
  ↪ rep(incendio_estudio$FECHA_INIC+45,length(grid)),geometry
  ↪ = grid)) %>% # 45 días después del incendio
bind_rows(tibble(date =
  ↪ rep(incendio_estudio$FECHA_INIC-15,length(grid)),
  ↪ geometry = grid)) %>% # 15 días antes
bind_rows(tibble(date =
  ↪ rep(incendio_estudio$FECHA_INIC-30,length(grid)),geometry
  ↪ = grid)) %>% # el mes antes
st_sf()

```

Se le asocian a cada registro todas las variables predictoras correspondientes.

```

source("scripts/strat/fun_asignar_variables - copia.R")
full_grid <- asignar_variables(sample)

# Se eliminan las observaciones con valores perdidos:
full_grid <- full_grid %>% drop_na()

# Se agrupan los niveles de uso de suelo
full_grid <- full_grid |>
  mutate(uso_suelo = fct_other(uso_suelo,
    keep =
      ↪ c("21","22","23","24","31","32","33"),
    other_level= "Otro"))

full_grid %>% st_drop_geometry() %>% skim()

# save(full_grid,file =
  ↪ "salidas_intermedias/full_grid_incendio_0921_processed.RData")

```

Se usará el modelo de regresión logística lasso final para estimar la probabilidad de incendio en cada punto.

```

load("Private/all_models_test.RData")

model <- models %>% filter(model_name=="lr")
# model <- models %>% filter(model_name=="sum_rbf")
# model <- models %>% filter(model_name=="sum_linear")
# model <- models %>% filter(model_name=="rf")

```

```
rm(models) # Es un archivo muy pesado, lo eliminamos de la memoria

# Predicciones
pred_class = model %>%
  pull(last_fit) %>%
  .[[1]] %>%
  extract_workflow() %>%
  predict(new_data = full_grid)
pred_probs = model %>%
  pull(last_fit) %>%
  .[[1]] %>%
  extract_workflow() %>%
  predict(new_data = full_grid, type="prob")
pred = cbind(full_grid, pred_class, pred_probs)
```

Se muestra en un gráfico

```
g <- ggplot(data = and) +
  geom_sf() +
  geom_sf(data = pred, aes(color=.pred_1, alpha = .pred_1), size = 1.5) +
  facet_wrap(~date) +
  scale_color_gradientn(colours = rainbow(5, rev=T), limits=c(0,1)) +
  # scale_color_gradient(low="blue", high="red")+
  labs(title="Incendio de Sierra Bermeja",
       subtitle = paste0("Model: ",model$model_name)) +
  guides(alpha = "none") +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  geom_sf(data = incendio_estudio, fill="transparent", col="red") +
  labs(color = "Probabilidad\nestimada de\nincendio") +
  coord_sf(xlim=c(bbox$xmin,bbox$xmax), ylim=c(bbox$ymin,bbox$ymax))
```

g

Bibliografía

- [1] ALLAIRE, JJ; XIE, YIHUI; DERVIEUX, CHRISTOPHE; MCPHERSON, JONATHAN; LURASCHI, JAVIER; USHEY, KEVIN; ATKINS, ARON; WICKHAM, HADLEY; CHENG, JOE; CHANG, WINSTON y IANNONE, RICHARD (2024). *rmarkdown: Dynamic Documents for R*.
<https://github.com/rstudio/rmarkdown>. R package version 2.26,
<https://pkgs.rstudio.com/rmarkdown/>.
- [2] AWATI, KAILASH (2024). «Eight to Late». <https://eight2late.wordpress.com/>.
- [3] BOETTIGER, CARL (2018). *rdflib: A high level wrapper around the redland package for common rdf applications*. doi: 10.5281/zenodo.1098478.
<https://doi.org/10.5281/zenodo.1098478>.
- [4] BREIMAN, L.; FRIEDMAN, J.; STONE, C.J. y OLSHEN, R.A. (1984). *Classification and Regression Trees*. Taylor & Francis. ISBN 9780412048418. doi: <https://doi.org/10.1201/9781315139470>.
- [5] CZERNECKI, BARTOSZ; GŁOGOWSKI, ARKADIUSZ y NOWOSAD, JAKUB (2020). *Climate: An R Package to Access Free In-Situ Meteorological and Hydrological Datasets For Environmental Assessment*. doi: 10.3390/su12010394.
<https://github.com/bczernecki/climate/>. R package version 0.9.1.
- [6] GIMOND, MANUEL (2023). «Intro to GIS and Spatial Analysis». <https://mgimond.github.io/Spatial/>.
- [7] HASTIE, T.; TIBSHIRANI, R. y FRIEDMAN, J.H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer. ISBN 9780387848846.
<https://hastie.su.domains/Papers/ESLII.pdf>.
- [8] HERNANGÓMEZ, DIEGO (2024). *mapSpain: Administrative Boundaries of Spain*. doi: 10.5281/zenodo.5366622.
<https://ropenspain.github.io/mapSpain/>.
- [9] HIJMANS, ROBERT J. (2024). *terra: Spatial Data Analysis*.
<https://CRAN.R-project.org/package=terra>. R package version 1.7-71.
- [10] KUHN, MAX y WICKHAM, HADLEY (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles..*
<https://www.tidymodels.org>.

- [11] LIZ-LÓPEZ, HELENA; HUERTAS-TATO, JAVIER; PÉREZ-ARACIL, JORGE; CASANOVA-MATEO, CARLOS; SANZ-JUSTO, JULIA y CAMACHO, DAVID (2023). «Spain on Fire: A novel wildfire risk assessment model based on image satellite processing and atmospheric information».
- [12] LOVELACE, ROBIN; NOWOSAD, JAKUB y MUENCHOW, JANNE (2019). *Geocomputation with R*. CRC Press. ISBN 1-138-30451-4.
- [13] PEBESMA, EDZER (2018). «Simple Features for R: Standardized Support for Spatial Vector Data». *The R Journal*, **10**(1), pp. 439–446. doi: 10.32614/RJ-2018-009. <https://doi.org/10.32614/RJ-2018-009>.
- [14] PEBESMA, EDZER y BIVAND, ROGER (2023). *Spatial Data Science: With applications in R*. Chapman and Hall/CRC, London. doi: 10.1201/9780429459016. <https://r-spatial.org/book/>.
- [15] R CORE TEAM (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- [16] SHAO, YH y DENG, NY (2015). «The Equivalence Between Principal Component Analysis and Nearest Flat in the Least Square Sense». *Journal of Optimization Theory and Applications*, **166**, pp. 278–284. doi: 10.1007/s10957-014-0647-y. <https://link.springer.com/article/10.1007/s10957-014-0647-y>.
- [17] SPARKS, ADAM H. (2018). «nasapower: A NASA POWER Global Meteorology, Surface Solar Energy and Climatology Data Client for R». *The Journal of Open Source Software*, **3**(30), p. 1035. doi: 10.21105/joss.01035.
- [18] STOJANOVA, DANIELA; PANOV, PANCE; KOBLER, ANDREJ; DŽEROSKI, SAŠO y TAŠKOVA, KATERINA (2006). «Learning to predict forest fires with different data mining techniques».
- [19] SUTHAHARAN, SHAN (2016). *Machine Learning Models and Algorithms for Big Data Classification*, tomo 36 de *Integrated Series in Information Systems*. Springer New York, NY. doi: 10.1007/978-1-4899-7641-3.
- [20] TAY, J. KENNETH; NARASIMHAN, BALASUBRAMANIAN y HASTIE, TREVOR (2023). «Elastic Net Regularization Paths for All Generalized Linear Models». *Journal of Statistical Software*, **106**(1), pp. 1–31. doi: 10.18637/jss.v106.i01.
- [21] THABTAH, FADI; HAMMOUD, SUHEL; KAMALOV, FIRUZ y GONSALVES, AMANDA (2020). «Data imbalance in classification: Experimental evaluation». *Information Sciences*, **513**, pp. 429–441. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2019.11.004>. <https://www.sciencedirect.com/science/article/pii/S0020025519310497>.
- [22] VAN DER LOO, M. y DE JONGE, E. (2018). *Statistical Data Cleaning with Applications in R*. Wiley. ISBN 9781118897157. doi: 10.1002/9781118897126.
- [23] WICKHAM, H. y GROLEMUND, G. (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media. ISBN 9781491910368. <https://r4ds.hadley.nz/>.

- [24] WICKHAM, HADLEY; AVERICK, MARA; BRYAN, JENNIFER; CHANG, WINSTON; McGOWAN, LUCY D'AGOSTINO; FRANÇOIS, ROMAIN; GROLEMUND, GARRETT; HAYES, ALEX; HENRY, LIONEL; HESTER, JIM; KUHN, MAX; PEDERSEN, THOMAS LIN; MILLER, EVAN; BACHE, STEPHAN MILTON; MÜLLER, KIRILL; OOMS, JEROEN; ROBINSON, DAVID; SEIDEL, DANA PAIGE; SPINU, VITALIE; TAKAHASHI, KOHSKE; VAUGHAN, DAVIS; WILKE, CLAUS; WOO, KARA y YUTANI, HIROAKI (2019). «Welcome to the tidyverse». *Journal of Open Source Software*, 4(43), p. 1686. doi: 10.21105/joss.01686.
- [25] XIE, YIHUI (2014). «knitr: A Comprehensive Tool for Reproducible Research in R». En: Victoria Stodden; Friedrich Leisch y Roger D. Peng (Eds.), *Implementing Reproducible Computational Research*, Chapman and Hall/CRC. ISBN 978-1466561595.
- [26] —— (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd.^a edición.
<https://yihui.org/knitr/>. ISBN 978-1498716963.
- [27] —— (2023). *knitr: A General-Purpose Package for Dynamic Report Generation in R*.
<https://yihui.org/knitr/>. R package version 1.45.
- [28] XIE, YIHUI; ALLAIRE, J.J. y GROLEMUND, GARRETT (2018). *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9781138359338.
<https://bookdown.org/yihui/rmarkdown>.
- [29] XIE, YIHUI; DERVIEUX, CHRISTOPHE y RIEDERER, EMILY (2020). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9780367563837.
<https://bookdown.org/yihui/rmarkdown-cookbook>.