Índice general

Capítulo 1

Preliminares

1.1. Datos georreferenciados

Todos los datos empleados en este trabajo son georreferenciados, lo que significa que están asociados a ubicaciones geográficas específicas. Por ello, resulta esencial introducir, aunque sea de forma general, 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 rasters, al ser los tipos de datos fundamentales en este contexto, con características bien diferenciadas entre ellos.

1.1.1. Datos Vectoriales

El modelo de datos vectoriales geográficos 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.

1.1.1.1. Simple features

Las "Simple features" son un estándar abierto ampliamente usado para la representación de datos vectoriales, desarrollado y respaldado por el Open Geospatial Consortium (OGC, por sus siglas en inglés), 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 ingés) 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 geoprocesamiento (GDAL, PROJ, GEOS, S2,...).

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 geometríca, dimensión, "Bounding Box" (límites o extensión geográfica) e información sobre el Sistema de referencia de coordenadas. Y además, presentan una columna de geometrías que tiene el nombre de "geom". Algunas ventajas del uso del modelo de "simple features" 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 and GEOMETRYCOLLECTION.

1.1.2. Datos Raster

El modelo de datos raster representa el espacio con una cuadrícula de celdas (también llamadas píxeles), que generalmente es regular, es decir, con todas las celdas de igual tamaño. Aunque no se tratarán en el presente trabajo, cabe mencionar que existen otros modelos de raster más complejos en los que se usan cuadrículas irregulares (rotadas, truncadas, rectilíneas o curvilíneas) y que pueden manipularse con el paquete de R (stars)[https://cran.r-project.org/web/packages/stars/index.html]. A cada una de estas celdas se le asocia uno (rasters de una sola capa) o varios (rasters multicapa).

Los datos en formato raster constan de una cabecera y una matriz cuyos elementos representan celdas equipespaciadas. En la cabecera del raster 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 usado en este trabajo usa por defecto el de la esquina superior izquierda). La resolución se calcula como:

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

La representación en forma de matriz evita tener que almacenar explícitamente las coordenadas de cada una de las cuatro esquinas de cada píxel, 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 raster 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 rásters regulares con una o varias capas a través de la clase de objetos `SpatRaster`. Sin embargo, existen otras alternativas, como el paquete (stars)[https://cran.r-project.org/web/packages/stars/index.html], que además de ser más potente, permite trabajar con rásters no regulares y ofrece una mejor integración con el paquete sf y el entorno tidyverse.

1.1.3. 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.

1.1.3.1. Sistemas de Coordenadas Geográficas

Los sistemas de coordenadas geográficas (GCS por sus siglas en inglés) 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 longitud 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 el 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 el el centro del elipsoide el que se alinea con el centro de la Tierra. Ejemplos de datums 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.

1.1.3.2. Sistemas de Coordenadas Proyectadas

Un Sistema de Coordenadas Proyectadas (PCS por sus siglas en inglés) 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, 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, 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. La mayoría de las proyecciones también se pueden clasificar en planas, cilíndricas o cónicas en función de cómo se realiza la proyección.

Un caso particular y ampliamente usado de PCS cilíndrico son los Universe Transverse Mercator (UTM), en el 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^{o} 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).

1.2. Análisis exploratorio de datos

El análisis exploratorio de datos (EDA, por sus siglas en inglés), 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 buenos resultados con ellos.

El EDA hace referencia al conjunto de técnicas estadísticas 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.

1.2.1. 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 R4DS, Wickman) 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.

Entendemos por datos técnicamente correcto cuando cada valor pertenece a una variable y está almacenado en el tipo que 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 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.

1.2.2. PCA

1.3. 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.

1.3.1. 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$i = 1, 2, ..., n$, se verifica que las variables $Y|_{\underline{X}=\underline{x}_i}$ y su distribución pertenece a la famila Bernouilli, es decir,

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

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

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

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 $\{(\underline{x}_i, y_i)\}_{i=1,...,n}$ 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 $\underline{\beta} = \left(\beta_0, \underline{\beta}_1\right)$, 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*.

El paquete glmnet implementa una combinación de ambos métodos (llamada *elastic net*), en la que se añade un parámetro de mezcla $\alpha \in [0,1]$ que combina ambos enfoques. El problema de optimización resultante es:

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

1.3.2. Support Vector Machine

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

1.3.2.1. Linear SVM

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

$$h(x) = w_1 x_1 + w_2 x_2 + ... + w_d x_d + b = \underline{w}^t \underline{x} = 0$$

que mejor separe a la muestra.

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

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

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

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.

Dado un punto $\underline{x}_i$ y un hiperplano $h(x) = w_1 x_1 + w_2 x_2 + ... + w_d x_d + b = \underline{w}^t \underline{x} = 0$, la distancia entre ambos viene dada por:

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

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

Dada una muestra linealmente separable $\{(\underline{x}_i, y_i)\}_{i=1,...,n}$ 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

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

O equivalentemente,

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

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

De está ultima expresión se deduce que maximizar el margen τ es equivalente a minimizar la norma euclídea de 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:

$$\begin{array}{ll} \min\limits_{w,b} & \frac{1}{2}w^t w \\ \text{s.t.} & \underline{w}^t \underline{x}_i + b \geq 1, \qquad \forall i \in \{1, ..., n\} \\ & w \in \mathbb{R}^d, b \in \mathbb{R} \end{array} \tag{1.1}$$

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 $\xi_i, \quad i = 1, ..., n$. Se habla entonces de hiperplano de separación *soft margin*. De esta forma, se llega al problema de optimización siguiente:

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2}w^t w + C\sum\limits_{i=1}^{n} \xi_i \\ \text{s.t.} & \underline{w}^t \underline{x}_i + b \geq 1, \qquad \forall i \in \{1, ..., n\} \\ & \xi \geq 0, \qquad \forall i \in \{1, ..., n\} \\ & w \in \mathbb{R}^d, b \in \mathbb{R} \end{array} \tag{1.2}$$

1.3.3. Random Forest

1.3.4. Redes Neuronales

1.4. Validación del ajuste

Partición entrenamiento/ validación / test

1.5. Evaluación 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. Además, 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 a distintas métricas, en función de las características propias de cada problema.

1.5.1. 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 definen las siguientes métricas de rendimiento de un modelo de clasificación binaria:

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

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

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

$$Precisión = \frac{TP}{TP + FP}$$

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

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

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

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

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.

1.6. Herramientas

Toda la parte práctica del presente trabajo se ha llevado a cabo empleado el lenguaje de programación R a través del entorno de desarrollo integrado (IDE) 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. R 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 las que cuenta este lenguaje.

Los paquetes que se han utilizado han sido:

- tidyverse:

  • ggplot2, para la visualización. • dplyr, para la manipulación. • tidyr, para la ordenación. • readr, para la importación. • purrr, para la programación funcional

- tidymodels
- sf
- terra
- nasapower: obtención de información climática satelital
- mapSpain:

. . . (podemos seguir hasta el infinito)