Universidad Nacional de La Plata
Facultad de Ciencias Astronómicas y Geofísicas

Tesis para obtener el título de
Licenciado en Astronomía

Detección y clasificación de señales en series de tiempo de Blazares utilizando algoritmos de aprendizaje automático

Manuel López Vargas

Director: Dr. Daniel D. Carpintero
Codirector: Dr. Tobías Canavesi

La Plata, Argentina
- Febrero de 2025 -

A mi gran amor,
por tu apoyo y amor incondicional.
Gracias, An.

Resumen

El objetivo general de este trabajo es utilizar herramientas de aprendizaje automático (*Machine Learning, ML*) para el estudio de microvariabilidad en AGN y contrastar sus resultados, beneficios y complejidades con las técnicas clásicas como tests estadísticos. A lo largo de este trabajo entrenaremos modelos de ML utilizando 5225 observaciones de AGN simulados en IRAF, de los cuales 5000 son no variables, 225 son variables y entre los cuales 25 son periódicos. Luego de hacer una limpieza y análisis de estos datos intentaremos emplear distintos algoritmos de ML hasta encontrar el óptimo para estos problemas de clasificación. Una vez entrenados los modelos clasificaremos observaciones fotométricas de 18 AGN en distintas bandas. Antes que nada, describiremos el modelo más aceptado para los AGN, sus características y componentes, así como también trataremos de entender por qué es importante estudiar la variabilidad de estos objetos y qué información nos puede dar esto acerca del objeto compacto central. Luego daremos una introducción a la inteligencia artificial y al ML, y describiremos cuantitativamente dos tipos de algoritmos de clasificación basados en árboles de decisión, precisamente *Random Forest* (bosque aleatorio o árboles aleatorios) y *XGBoost* (*eXtreme Gradient Boost*).

Finalmente construiremos un algoritmo *XGBoost* para la clasificación de la variabilidad el cual obtuvo buenas métricas de rendimiento, y lo compararemos con el test estadístico de Fisher (método comúnmente utilizado para detectar variabilidad en objetos astrofísicos), obteniendo mejores resultados que dicho test, habiendo clasificado bien el 99,5 % de los AGN sintéticos para los cuales ya se conocía su variabilidad, contra un 82,7 % por parte del test estadístico.

Abstract

The general objective of this work is to use ML tools for the study of microvariability in AGNs and to contrast their results, benefits and complexities with classical techniques such as statistical tests.

Throughout this work we will train Machine Learning (ML) models using 5225 observations of AGNs simulated in IRAF, of which 5000 are non-variable, 225 are variable and among which 25 are periodic. After cleaning and analyzing these data we will try to use different ML algorithms until we find the optimal one for these classification problems. Once the models are trained we will classify photometric observations of 18 AGNs in different bands.

First of all, we will describe the most accepted model for AGNs, its characteristics and components, as well as we will try to understand why it is important to study the variability of these objects and what information this can give us about the central compact object. Then we will give an introduction to artificial intelligence and ML, and we will describe quantitatively two types of classification algorithms based on decision trees, namely Random Forest and XGBoost (eXtreme Gradient Boost).

Finally we built a *XGBoost* algorithm for variability classification which obtained good performance metrics and compared it with Fisher's statistical test (a method commonly used to detect variability in astrophysical objects) and obtained better results than that test, classifying well 99,5 % of the synthetic AGNs for which their variability was already known, against 82,7 % by the statistical test.

Agradecimientos

A mi familia, mis amigos, todas las personas que me acompañaron durante la carrera de las cuales aprendí mucho. A mi facultad, a los profesores, a mi director Daniel que me tuvo mucha paciencia y a Tobi, mi codirector, quien no solo me ayudo con este trabajo sino que también fue mi mentor en el mundo de la ciencia de datos y a quien le debo mucho.

Gracias.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La presente Tesis se enmarca dentro de un proyecto de investigación que apunta a estudiar la microvariabilidad de los AGN, cuyo comportamiento puede ayudarnos a entender la estructura del BH central y los chorros (*jets*).

El objetivo general de esta Tesis consiste en utilizar algoritmos de ML, en particular algoritmos basados en árboles de decisión como *Random Forest* y *XGBoost*, para clasificar los AGN en variables y no variables.

Nos enfocaremos particularmente en estudiar la radiación de los blazares. Sin embargo, estos algoritmos se pueden generalizar para clasificar, en principio, cualquier serie de tiempo.

1.1. Contexto

Los AGN son una de las fuentes de energía más intensas del Universo. Estos objetos emiten energías del orden de $\sim 10^{40}$ erg s^{-1} en galaxias cercanas, hasta $\sim 10^{47}$ erg s^{-1} en cuásares lejanos. Los AGN emiten en todo el espectro electromagnético, con un pico en el ultravioleta y cantidades significativas en rayos X y en las frecuencias de infrarrojo hasta ondas de radio. En esta última banda se suelen encontrar eyecciones de material colimado relativista, comúnmente conocidas como *jets*.

Por otro lado, estas fuentes pueden presentar características como variabilidad en escalas temporales de años o incluso horas o minutos (microvariabilidad). Esta variabilidad en escalas de tiempo tan cortas solo puede ser explicada si se tiene un objeto compacto. Teniendo en cuenta que, a su vez, las altas luminosidades implican grandes masas, se propone el modelo de agujero negro súper masivo (*Super massive black hole*, SMBH) para explicar la naturaleza de los AGN, con masas $M_\bullet \geq 10^8\ M_\odot$, el cual acrece material a través de un disco de acreción a altas temperaturas y expulsa material por los polos del objeto compacto (*jets*). (Ahondaremos más en la morfología de los AGN en el Capítulo 2.)

Existen diferentes tipos de AGN, según su espectro:

- **Núcleos de galaxias Seyfert:** Son los menos luminosos, con magnitudes $M_B > -23$ mag. Se caracterizan espectroscópicamente por tener una componente estelar tipo G y líneas de emisión intensas y de alta excitación.
  Estos AGN se dividen en dos subtipos. Los Seyfert I poseen líneas de emisión permitidas muy anchas (velocidades Doppler hasta $\sim 10^4$ km s^{-1}), y líneas prohibidas relativamente angostas. Los Seyfert II se caracterizan por tener líneas de emisión relativamente angostas (~ 3000 km s^{-1}), tanto permitidas como prohibidas, y el continuo es relativamente más débil que en las Seyfert I.
  Este tipo de objetos en general se encuentran en galaxias de tipo temprano.

- **Núcleos de radiogalaxias:** Son fuentes de radio muy intensas debido principalmente a los *jets* con una componente en óptico (no siempre) debida al núcleo, y se clasifican según su emisión en el rango visual: radiogalaxias de líneas anchas (*Broad-line radio galaxies*) y radiogalaxias de líneas angostas (*Narrow-line radio galaxies*).

- **Cuásares:** Los cuásares son los AGN más luminosos, con magnitudes $M_B \leq -23$ mag. Estos aparentan ser objetos puntuales, de ahí el nombre objetos casi estelares (*Quasi-stellar objects*, QSO) o cuásares. Se subclasifican segun su emisión en radio como cuásares radiointensos (*Radio loud quasars*, RLQ) y cuásares radiosilenciosos (*Radio quiet quasars*, RQQ).

- **Objetos BL Lac - Blazares:** Los objetos BL Lac están caracterizados por tener espectros ópticos sin líneas de emisión o absorción fuertes. Estos junto con los radio-cuásares de espectro plano (*Flat spectrum radio-quasars*), que se caracterizan por tener variaciones en el visual en escalas de tiempo muy cortas (días), forman una categoría de AGN llamada blazares. Estos objetos poseen un *jet* en la línea de la visual y todos los descubiertos hasta la fecha son fuentes de radio.

- **LINER:** Los LINER (*Low-ionization nuclear emission-line region*) se encuentran en galaxias con baja luminosidad. Son muy comunes y se pueden identificar también con galaxias Seyfert de baja luminosidad.

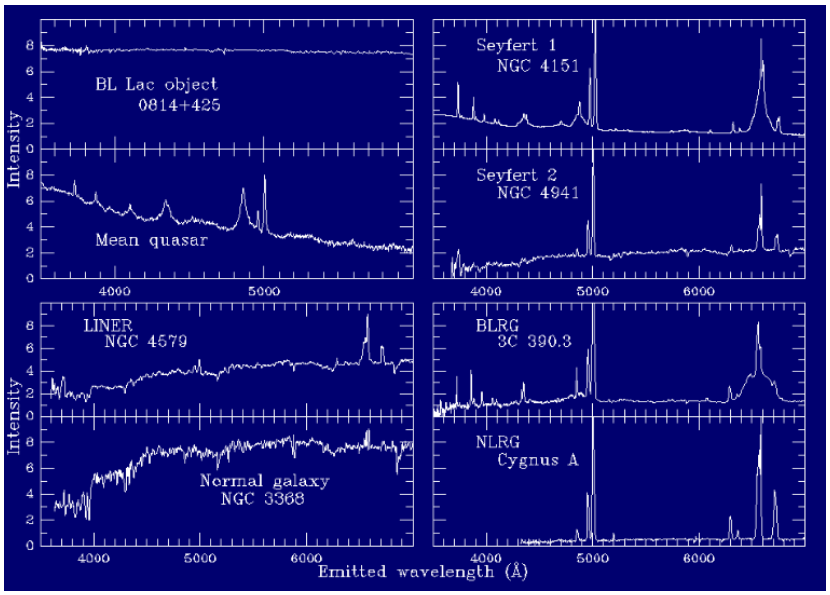

Figura 1.1. Espectros ópticos de distintos AGN y de una galaxia de tipo temprano (Keel 1983, Owen et al. 1990, Lawrence et al. 1996).

Capítulo 2

Modelo de AGN

Las propiedades de las galaxias Seyfert y de las radiogalaxias (RG) son básicamente idénticas a las de los RQQ y RLQ, salvo por un factor de escala en la luminosidad. Esto lleva a enmarcar a todos estos objetos bajo el mismo fenómeno subyacente, el de los AGN. Aunque no hay prueba directa, la evidencia apunta a que se trata de objetos compactos (BH) acreciendo material a través de un disco de acreción.

2.1. Fenomenología

El espectro continuo de los AGN puede ser modelado con una ley de potencia. Así, la luminosidad L en función de la frecuencia ν está dada por:

$$L(\nu) = A\nu^{-\alpha}, \tag{2.1}$$

donde A y el índice espectral α son constantes.

Sobre las bases de las leyes de potencia observadas de polarizaciones en algunas frecuencias y de los tamaños compactos de las regiones de emisión, se encuentra que la radiación del continuo en las RG, galaxias Seyfert y los cuásares es de origen térmico y proviene del disco de acreción, mientras que en los blazares domina la emisión producida por el *jet* y esta es de origen no térmico, principalmente radiación sincrotrón y efecto Compton inverso.

Por su parte, la clasificación de los blazares depende principalmente de su aspecto en la banda óptica, donde presentan una componente no térmica asociada al *jet*, una componente térmica asociada a la acreción hacia el SMBH y desde la región de líneas anchas (la cual definiremos más adelante) y por último la luz proveniente de la galaxia anfitriona (elíptica).

En la Fig. 2.1 se muestran estas componentes en la distribución espectral de energía (*spectral energy distribution*, SED) de cuatro AGN conocidos.

2.2. El modelo estándar de AGN: SMBH

A partir del estudio del cuásar 3C−273 se planteó la posibilidad de la existencia de una gran masa central (v.g., Salpeter 1964; Zwicky 1966). Se requiere una masa para poder mantener ligado el gas de alta velocidad que produce las líneas anchas. Con lo que desde un

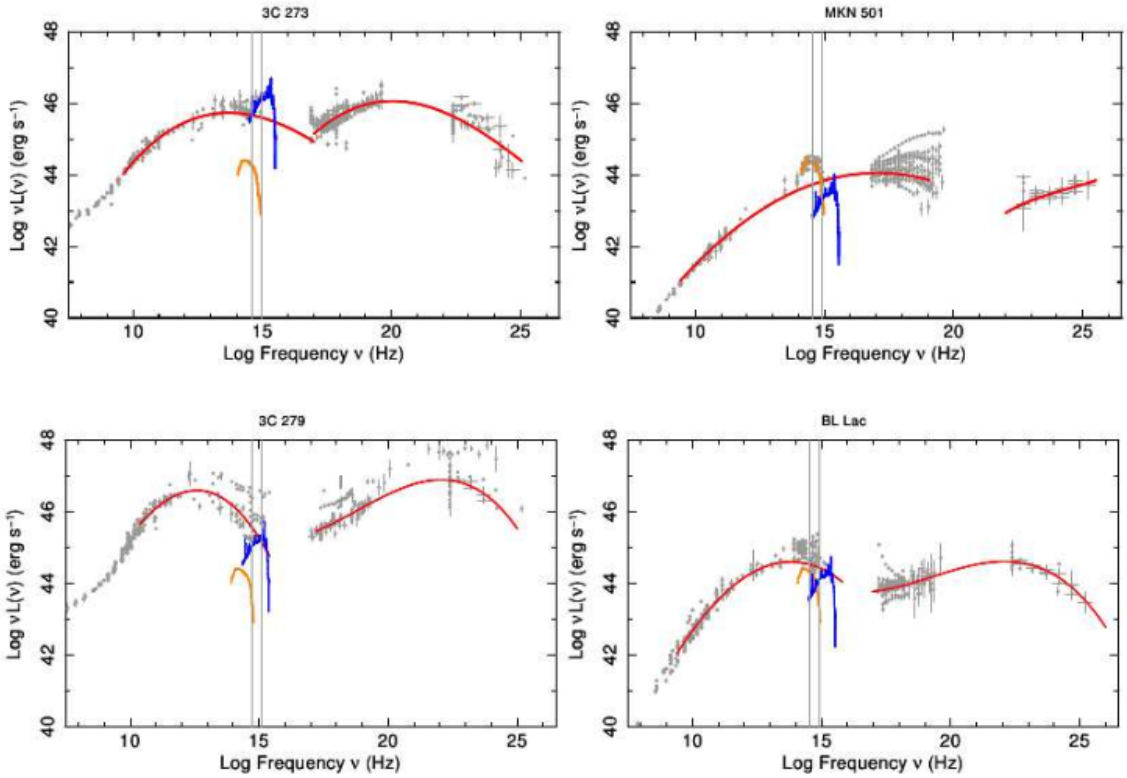

Figura 2.1. SEDs de cuatro blazares: dos FSRQ, 3C 273 y 3C 279, y dos BL Lacs, MKN501 y BL Lac. Las líneas rojas corresponden a la emisión no térmica del *jet*; las líneas azules representan la emisión desde el disco y desde la región de líneas anchas; y las líneas naranjas muestran la luz de la galaxia anfitriona. Las dos líneas verticales indican la ventana óptica observada (Giommi et al. 2012).

principio se sugirió que la fuente de energía de los AGN es la acreción de material hacia un objeto compacto.

El modelo de objeto compacto, en particular el de SMBH se describe cualitativamente en la Fig. 2.2, como así también las subestructuras responsables de las diferentes componentes de la SED.

Un argumento muy convincente de este supuesto es el propuesto por Lynden-Bell (1969), quien midió la energía $E_{\rm RG}$ almacenada en los lóbulos de las radiogalaxias encontrando valores de $E_{\rm RG} \sim 10^{54}$ J. La masa asociada a esta energía es:

$$\mathcal{M}_{\rm rad} = E_{\rm RG}\ c^{-2} \simeq 6 \times 10^6 \mathcal{M}_\odot, \tag{2.2}$$

donde c es la velocidad de la luz. Si (2.1) fuera el resultado de fusión nuclear, requeriría una masa original

$$\mathcal{M} \geq \frac{\mathcal{M}_{\rm rad}}{0.007} \approx 10^9 \mathcal{M}_\odot. \tag{2.3}$$

Por otra parte, las escalas de tiempo de variabilidad implican

$$R \leq c\ \Delta t \Rightarrow R \leq 10\ \text{hora} - \text{luz} = 72\ \text{au}. \tag{2.4}$$

Aquí R es el tamaño de la fuente y Δt la escala de tiempo de variabilidad. La energía de

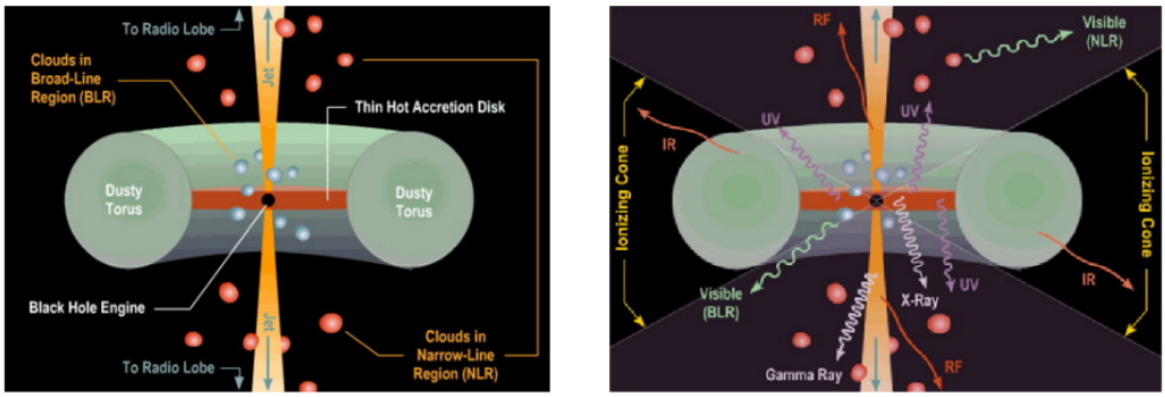

Figura 2.2. El modelo estándar: componentes de un AGN (panel izquierdo) y sus contribuciones a la SED (panel derecho). Imagen original obtenida de: http://astronomyonline.org/Cosmology/Galaxies.asp.

ligadura asociada al sistema es entonces:

$$\frac{G\mathcal{M}^2}{R} \simeq 1.7 \times 10^{55}\ \mathrm{J}, \tag{2.5}$$

siendo G la constante de gravitación universal.

Por lo que, con este argumento, tenemos un modelo que produce energía más que suficiente por contracción gravitatoria para explicar la energía detectada.

Además del argumento propuesto por Lynden-Bell (1969), otro razonamiento que apoya la idea de una gran masa central se basa en que el balance entre la presión de radiación y la gravedad requiere que la luminosidad bolométrica no supere el límite de Eddington.

La aceleración debida a la presión de radiación es:

$$a_{\mathrm{rad}} = \frac{\sigma_{\mathrm{T}}}{\mu_{\mathrm{p}}} \frac{\mathcal{L}}{4\pi c\ r^2}, \tag{2.6}$$

donde σ_{T} es la sección eficaz de Thomson del electrón, $\mathcal{L}$ es la luminosidad, r es la distancia a la fuente y μ_{p} es la masa del protón. El cociente entre las aceleraciónes por presión de radiación y gravitatoria es entonces:

$$\frac{a_{\mathrm{rad}}}{g} = \frac{\sigma_{\mathrm{T}}\mathcal{L}}{4\pi c\ \mu_{\mathrm{p}}\ G\mathcal{M}_\bullet} = \frac{\mathcal{L}}{L_{\mathrm{Edd}}}, \tag{2.7}$$

donde $\mathcal{M}_\bullet$ es la masa del objeto compacto, y la luminosidad de Eddington viene dada por:

$$L_{\mathrm{Edd}} = \frac{4\pi c\ G\ \mathcal{M}_\bullet\ \mu_{\mathrm{p}}}{\sigma_{\mathrm{T}}} = 1,51 \times 10^{31} \frac{\mathcal{M}_\bullet}{\mathcal{M}_\odot} \mathrm{W}. \tag{2.8}$$

Considerando una luminosidad de 10^{40} W

$$\frac{\mathcal{L}}{L_{\mathrm{Edd}}} \simeq \frac{10^{40}\ \mathrm{W}}{1,51 \times 10^{31} \frac{\mathcal{M}_\bullet}{\mathcal{M}_\odot}\ \mathrm{W}} \Rightarrow \mathcal{M}_\bullet \approx 10^9 \mathcal{M}_\odot, \tag{2.9}$$

llegamos a una masa 10^9 veces la masa del Sol.

2.3. Estructura y componentes del AGN

Cualquier modelo de AGN tiene que ser capaz de explicar esta amplia fenomenología:

- Naturaleza de la emisión del continuo
- Naturaleza de la emisión de líneas
- Líneas anchas y líneas delgadas
- RQQ y RLQ
- *Jets* y lóbulos emisores en radio
- Blazares

Para explicar todo esto, el modelo de Urry y Padovani (1995) involucra (ver Fig. 2.2):

- Un SMBH central
- Un disco de acreción rodeando al SMBH
- Un anillo grueso (toro) de gas y polvo
- Dos *jets* de plasma relativista perpendiculares al disco (no siempre presentes).
- Una región cercana del SMBH con nubes de gas moviéndose a altas velocidades: región de líneas anchas
- Una región alejada al SMBH con nubes de gas a bajas densidades y velocidades: región de líneas angostas

Este modelo funciona porque es capaz de explicar cómo las distintas componentes del AGN son responsables de las emisiones en distintas frecuencias, y porque permite unificar los distintos tipos de AGN según su orientación con respecto a la visual (modelo anisótropo).

La anisotropía es la responsable de dos procesos físicos que dependen de la orientación con la visual: extinción y efecto faro (*beaming*), los cuales permiten unificar distintos tipos de AGN en un mismo escenario. Como se muestra en la Fig. 2.3, los distintos tipos de AGN se observan según su orientación con la línea de la visual. En efecto:

- En las galaxias Seyfert I y BLRG, hay una visión directa a la BLR y a las regiones internas del disco de acreción. Es también el caso de los QSO, que en general presentan líneas anchas.
- En las galaxias Seyfert II y NLRG, el toro bloquea la emisión óptica de la BLR y de las regiones internas del disco de acreción. La emisión X blanda es absorbida por el gas.
- En los blazares el *jet* está en la dirección de la visual. Luego, debido al efecto faro relativista, domina el continuo emitido por el *jet*.

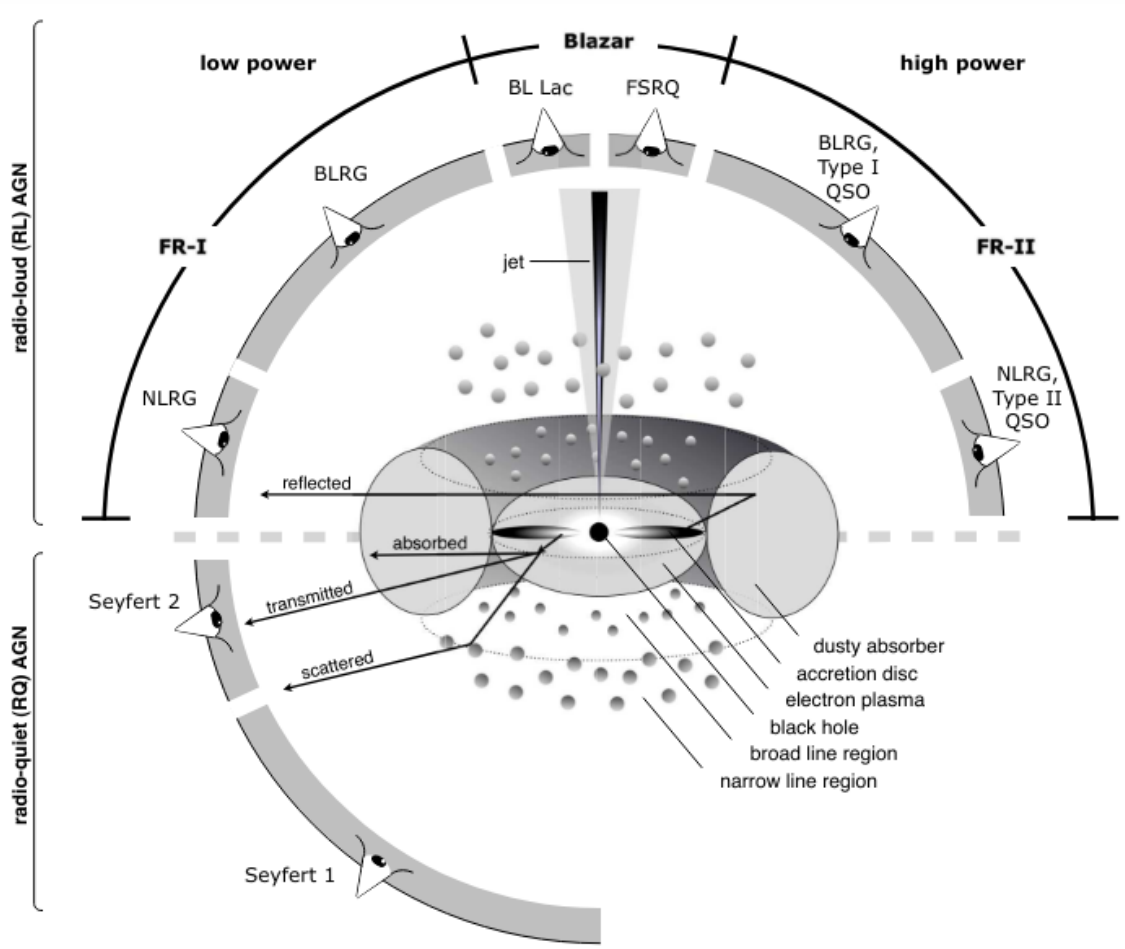

Figura 2.3. Representación esquemática del fenómeno AGN en el esquema unificado. El tipo de objeto que vemos depende de la dirección de la visual, de si el AGN produce o no un *jet* con emisión significativa y de la potencia del objeto central (Beckmann y Shrader 2013).

2.4. *Jet* relativista

2.4.1. Modelo de *Jet*

El mecanismo de formación de los *jets* no está bien entendido. Los modelos más aceptados son los propuestos por Blandford y Rees (1974), y Blandford y Payne (1982), que se tratan de modelos magneto-centrífugos. Estos se basan en el hecho de que los campos magnéticos asociados al disco tienen dos componentes, una a gran escala y otra a menor escala. La segunda es la responsable del calentamiento del disco, mientras que la primera es la encargada de formar el *jet*. Así, una partícula se mueve a lo largo de las líneas de campo magnético, y estas a su vez rotan fijas al disco acompañando su movimiento. Cerca del disco las líneas de campo se mantienen perpendiculares al mismo. Pero cuanto más lejos estén del eje , más rápido rotarán, por lo que también lo harán las partículas que se muevan sobre las mismas. Esto desencadena una eyección colimada de partículas relativistas, en la dirección del eje de rotación del sistema (Fig. 2.4).

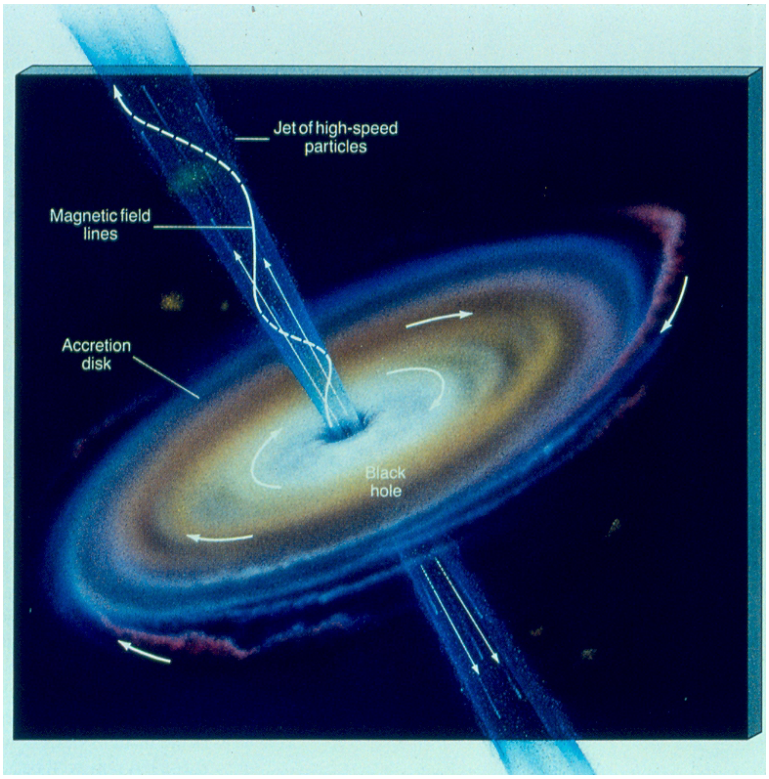

Figura 2.4. Líneas de campo magnético en el *jet* colimado. (Imagen obtenida de hildaand-trojanasteroids.net)

2.4.2. Emisión del jet

Como mencionamos anteriormente, la emisión del *jet* es debida a procesos no térmicos, tales como sincrotrón y efecto Compton inverso. La componente en radio se debe a la emisión sincrotrón, y en altas energías ambos procesos tienen relevancia. A su vez, existen a lo largo del *jet* procesos de pérdidas de energía debido a la expansión adiabática del *jet* (Fig. 2.5), y esto hace que muy pocos sean observables en el óptico.

La potencia en radio del AGN quedaría determinada por la rotación del SMBH, capaz de proveer la energía necesaria para acelerar el plasma de los *jets* hasta velocidades relativistas alcanzando escalas de kiloparsecs, pero sin afectar la emisión entre el infrarrojo y rayos X, que depende de la acreción.

2.5. Blazares

Como se muestra en la Fig. 2.3, un blazar es un AGN cuyo *jet* está orientado en la dirección de la visual, con un ángulo no mayor a 10°. En la banda óptica son objetos puntuales, su continuo proviene del núcleo y presentan un grado de polarización alto. La radiación de estos objetos es de origen no térmico y parece estar relativísticamente amplificada. En general un blazar presenta las siguientes características (Zibecchi 2013):

- Emisión de continuo suave desde el infrarrojo hasta el ultravioleta
- Alta polarización lineal en el óptico

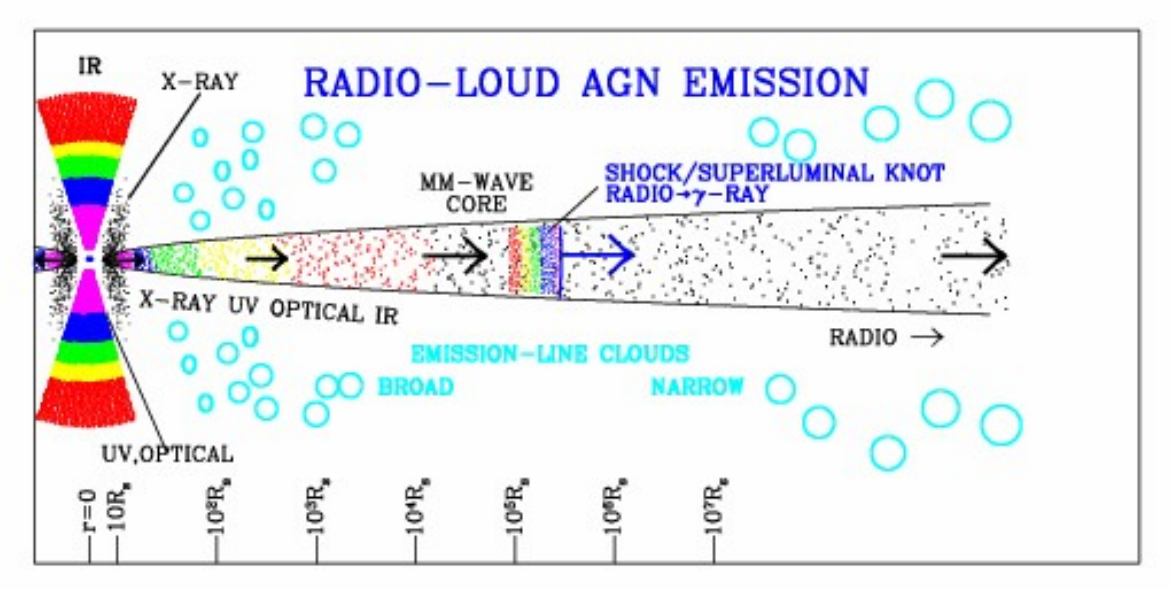

Figura 2.5. Representación de la expansión adiabática de un *jet* y las componentes responsables de las emisiones en las distintas bandas. (Imagen extraída de http://www.bu.edu/blazars/research.html).

- Alta variabilidad en el rango visual en escalas de tiempo menores a 1 día
- Emisión intensa y variable en el continuo de radio
- Fuerte emisión en rayos γ y rayos X

2.5.1. Microvariabilidad

Estudiar este fenómeno nos puede ayudar a entender los procesos físicos involucrados, así como también el tamaño de la región donde es producida la radiación. Si la escala temporal de la microvaribilidad es Δt, luego la región de radiación R estará acotada:

$$R \leq c\Delta t. \tag{2.10}$$

De todas las escalas temporales de variabilidad que se pueden hallar en una misma frecuencia para la fuente, siempre la menor es la asociada al tamaño de la misma. Si la fuente es mayor que esta cota, las distintas partes no puede estar causalmente conectadas.

El patrón de variabilidad en diferentes rangos espectrales puede tener implicaciones importantes sobre los modelos de los procesos radiativos involucrados. Si las variabilidades en dos frecuencias distintas muestran el mismo patrón y están en fase, entonces es probable que el mismo mecanismo esté generando ambas señales y dentro de la misma región física. Por el contrario, si una señal se atrasa sistemáticamente de la otra, se puede suponer que está involucrada la propagación de una región emisora a la otra.

Para longitudes de onda ópticas se han detectado variabilidades en escalas de tiempo de días hasta años, y también en muchos AGN puede haber microvariabilidad en diferentes longitudes de onda. En los años 90' se comenzó a estudiar este fenómeno de microvariabilidad en el óptico (Miller et al., 1989; Carini et al., 1990, 1991, 1992). A inicios del actual siglo, comenzaron a realizarse estudios de carácter sistemático sobre la incidencia de la microvariabilidad en los blazares (Romero et al., 1999; Stalin et al., 2004, Andruchow et al., 2005; Andruchow,

2006). Se piensa que este fenómeno está asociado al *jet*, ya que se detecta microvariabilidad si la emisión del *jet* domina en los blazares o es significativa en los RLQ.

Capítulo 3

Obtención y descripción de los datos

En el presente trabajo utilizamos dos conjuntos de datos diferentes. El primero se trata de curvas de luz diferenciales (*Differential light curves*, DLC) sintéticas simuladas utilizando IRAF (*Image reduction and analysis facility*) (Zibecchi et al. 2020). Estas se realizaron simulando todas las condiciones de observación tales como *seeing* variable, masa de aire, nubosidad y luz lunar. Estos son los datos que utilizaremos para entrenar nuestro algoritmo de ML.

El segundo conjunto se trata de observaciones ya publicadas anteriormente por Romero et al. (1999) y Romero et al. (2002), ambas obtenidas en el telescopio óptico Jorge Sahade en CASLEO. Con estas observaciones clasificaremos los AGN en variables y no variables.

3.1. DLC sintéticas

Zibecchi et al. (2020) simularon DLC de AGN variables y no variables en diferentes marcos o *frames*. Para ello, se colocó un conjunto de objetos puntuales en cada fotograma. En la Fig. 3.1 se muestran ejemplos de estos objetos artificiales. En la parte superior hay 200 objetos que representan AGN con magnitudes entre 16 y 17 mag. En la parte inferior hay 63 estrellas de campo, con magnitudes que cubren un rango de 15 a 17 mag, utilizadas como estrellas de comparación y control. El rango de magnitudes corresponde a valores cercanos a los de estrellas estándar en campos AGN (González-Pérez et al. 2001).

Las magnitudes estándar fueron transformadas a cuentas (ADU) en las imágenes CCD simuladas teniendo en cuenta el telescopio y la configuración instrumental utilizados para obtener los datos analizados en Andruchow et al. (2003). El telescopio Jorge Sahade tiene un tamaño de espejo de 2,15 m, más grande que la mayoría de los telescopios utilizados en estudios de variabilidad de AGN.

Utilizamos en total 25 marcos, los cuales cuentan cada uno con 200 AGN no variables, 9 AGN variables y 63 estrellas de control y comparación, y las DLC sintéticas de cada objeto están compuestas de 40 puntos o "medidas". El procedimiento realizado para obtener las DLC de todos los AGN y estrellas de control y comparación en diferentes marcos es el siguiente. En primer lugar, se toma el valor más bajo del *seeing*, sin incluir nubes ni Luna, y se agrega este *seeing* a la primera imagen de los 200 AGN. Luego se hace variar la masa de aire a lo largo de 40 valores diferentes, a efectos de simular el cambio en altura durante el transcurso de una noche. Así se obtiene la primera curva de luz de un AGN no variable, ya que la imagen fue siempre la misma. Se repite este procedimiento para los 199 AGN restantes y las

63 estrellas de control y comparación. Luego, se repite lo mencionado pero en vez de utilizar una misma imagen de AGN para toda la noche, se eligen distintas imágenes de entre las 200 originales, y así obtenemos un AGN variable. Esto se hace con 9 combinaciones distintas de imágenes de AGN para una misma noche, obteniendo así 9 AGN variables (Fig. 3.2). Se repite todo lo descripto para otros 4 valores de *seeing* y obtenemos finalmente los primeros 5 marcos, llamados casos de control (CTR). El siguiente marco se obtiene repitiendo todo el procedimiento mencionado pero en lugar de dejar fijo el *seeing* se hace variable, de 5 maneras distintas. Estos son los llamados calidad de imagen (IQ) o *seeing* variable. Luego, se toman los 5 CTR y se les agrega a cada uno a lo largo de la noche valores de nubosidades combinados de 5 maneras distintas. De estos marcos, en este trabajo tomamos los 5 generados con el valor más pequeño del *seeing* con 5 cambios de nubosidad. Otros 5 marcos se generan análogamente a los últimos pero con la iluminación lunar (sin nubosidad). Finalmente, los últimos 5 marcos se obtienen combinando el *seeing* más pequeño con el i-ésimo valor de nubosidad e iluminación lunar, con i tomando valores de 1 a 5. Se obtienen así los 25 marcos mencionados.

3.2. Observaciones de AGN

Los datos utilizados en este trabajo ya fueron publicados y estudiados anteriormente. Estos corresponden a curvas de luz de 58 objetos, de los cuales 23 son AGN del hemisferio sur (Romero et al. 1999), 20 son blazares EGRET (Romero et al. 2002), (3 de estos AGN están en ambos trabajos) y otros 18 objetos estudiados en Zibecchi et al. (2024). De estos AGN tomamos una muestra de 18, teniendo un total de 106 sesiones de observación, en los filtros rojo (R) y visual (V). Se entiende por sesión a una noche de observación de un dado AGN. La mayoría de estas imágenes fueron tomadas a fines de la década de 1990, mientras

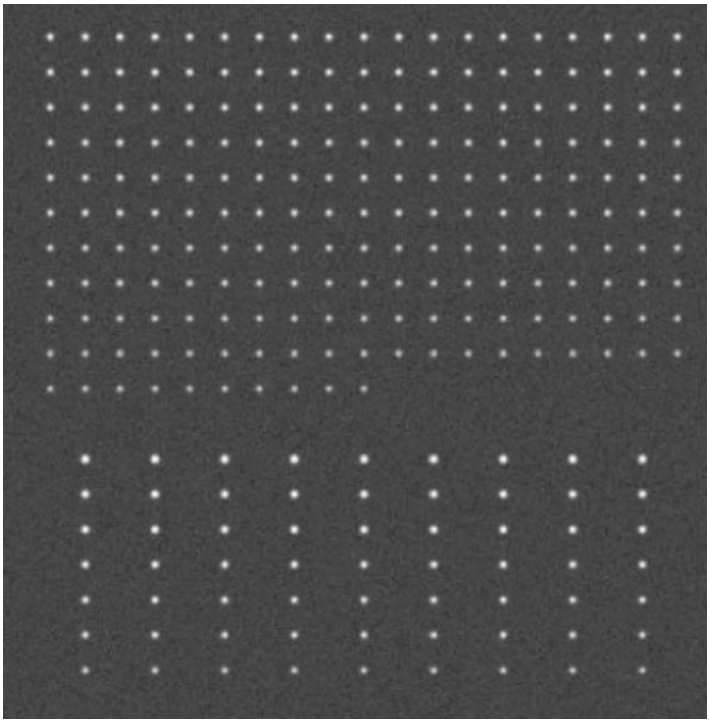

Figura 3.1. Fotograma CCD simulado. Conjunto superior: AGN con diferentes magnitudes. Conjunto inferior: candidatos a estrellas de comparación y de control (Andruchow et al. 2003).

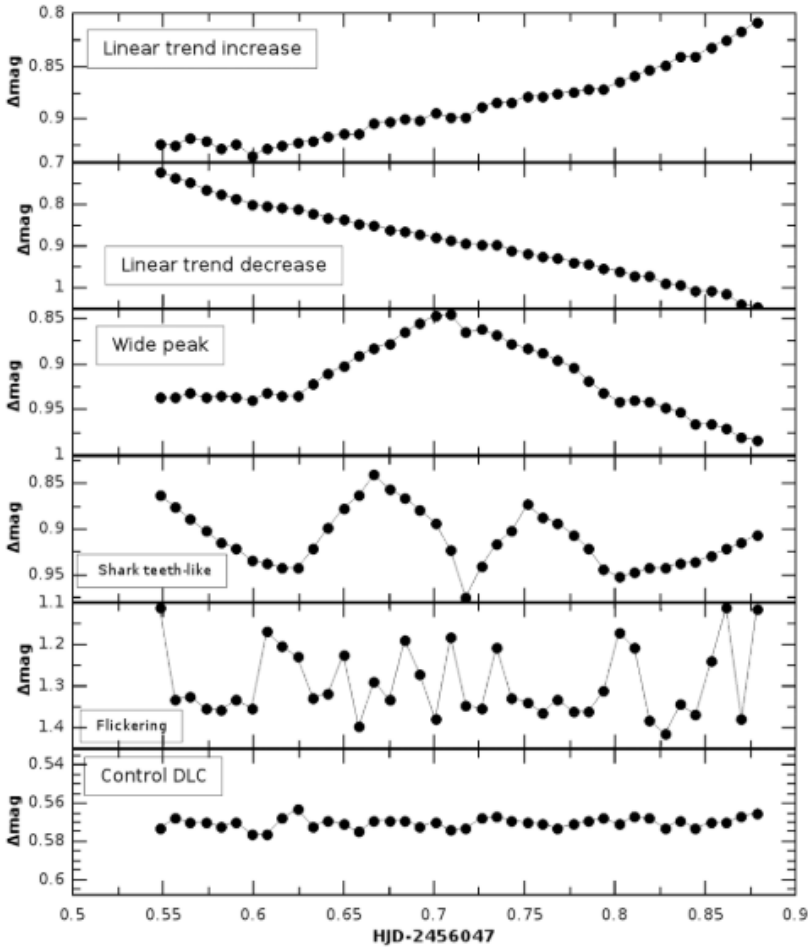

Figura 3.2. Diferentes tipos de variaciones, junto con la DLC de control. Desde el panel superior al inferior: variabilidad de tendencia lineal creciente, variabilidad de tendencia lineal decreciente, variabilidad de pico ancho, variabilidad en forma de dientes de tiburón, variabilidad parpadeante y DLC de control (Andruchow et al. 2003).

que otras fueron tomadas en 2015.

En el momento de las observaciones estos objetos se clasificaban como QSO, tanto RLQ como RQQ, y objetos BL Lac, que a su vez habían sido detectados en radiofrecuencias (*radio-selected*, RBL) y en rayos X (*X-ray-selected*, XBL). Luego se descubrió que este tipo de clasificación era producto de un efecto de selección, lo que producía una mala interpretación de lo observado. Posteriormente, la clasificación anterior fue reemplazada por una nueva basada en los índices espectrales radio−óptico y radio−rayos X, que separa aquellos objetos que presentan su máximo de radiación hacia el óptico (BL Lac con corte en bajas frecuencias) y los que presentan su máximo a frecuencias de rayos X (BL Lac con corte en altas frecuencias) (Padovani y Giommi 1995). Luego, con la publicación del primer catálogo del instrumento Fermi LAT, se corrigió la clasificación a partir de la frecuencia en la que cae el pico sincrotrón de la distribución espectral de energía. Así, se clasifican en los que presentan un bajo pico sincrotrón (*Low synchroton peak*, LSP), los que presentan un pico sincrotrón intermedio (*Intermediate synchrotron peak*, ISP) y los que presentan un pico sincrotrón alto (*High synchrotron peak*, HSP) (Abdo et al. 2010).

Las características de cada uno de estos objetos se encuentran detalladas en la Tabla 3.1, especificándose el nombre de la fuente, tipo de objeto, ascensión recta (α), declinación (δ),

corrimiento al rojo z y magnitud visual ($m_{\rm v}$). Los mismos fueron extraídos de la base de datos para fuentes extragalácticas de la NASA (NASA/IPAC *Extragalactic Database, NED*).

Objeto	Tipo	α (J2000.0) h:m:s	δ (J2000.0) °:':"	z	$m_{\rm v}$
0208-512	BLL/LSP$^{\bullet}$	02 : 10 : 46	-51 : 01 : 02	1.003	16.9
0521-365	BLL/LSP$^{\bullet}$	05 : 22 : 58	-36 : 27 : 31	0.55	14.5
1127-145	FSRQ/LSP$^{\bullet}$	11 : 30 : 07	-14 : 49 : 27	1.187	16.9
1229-021	QSO*	12 : 32 : 00	-02 : 24 : 05	1.045	17.7
1256-229	QSO*	12 : 59 : 08	-23 : 10 : 39	0.481	17.3
1424-418	FSRQ/LSP$^{\bullet}$	14 : 27 : 56	-42 : 06 : 19	1.522	17.7
1510-229	FSRQ/LSP$^{\bullet}$	15 : 12 : 50	-09 : 06 : 00	0.361	16.5
2005-489	BLL/HSP$^{\bullet}$	20 : 09 : 25	-48 : 49 : 54	0.071	13.4
2155-304	BLL/HSP$^{\bullet}$	21 : 58 : 52	-30 : 13 : 32	0.116	13.1
0414+009	BLL/HSP^{+}	04 : 16 : 52:2	01 : 05 : 23:9	0.287	15.86(R)
0846.9-2336	BZU^{+}	08 : 47 : 01:5	-23 : 37 : 01:6	0.061	13 (R)
1116-46	FSRQ/LSP^{+}	11 : 18 : 26:9	-46 : 34 : 15	0.713	17.02
1443-389	BLL/HSP^{+}	14 : 43 : 57:2	-39 : 08 : 39:7	0.139	14.81(R)
1917.7-1921	BLL/HSP^{+}	19 : 17 : 44:8	-19 : 21 : 31:6	0.137	15.24(R)
1958.2-3011	BLL/HSP^{+}	19 : 58 : 14:9	-30 : 11 : 11:8	0.119	13.97(R)
2126-158	FSRQ/LSP^{+}	21 : 29 : 12:1	-15 : 38 : 41	3.268	16.43(R)
2149-306	FSRQ/LSP^{+}	21 : 51 : 55:5	-30 : 27 : 53:6	2.340	17.48(R)
2310-4374	BLL^{+}	23 : 10 : 41:7	-43 : 47 : 34:1	0.088	15.92

Tabla 3.1. Datos de los AGN observados. $\bullet$: Abdo et al. (2010); $\star$: Veron y Cetty (2006); $+$: Zibecchi et al. (2024).

3.2.1. Fotometría diferencial

La fotometría diferencial fue desarrollada por Howell y Jacobi (1986) y un punto fundamental es tomar pequeñas exposiciones repetidamente del objeto de estudio. Algunas de las estrellas de fondo son utilizadas para comparación durante el proceso de reducción, en el cual se obtienen las magnitudes instrumentales de todos los objetos de interés. Luego, se calculan las magnitudes diferenciales a partir de las estrellas de comparación y control elegidas, como se explica mas adelante.

La principal característica de este método es que puede ser usado para objetos débiles, con magnitudes en el visual V $>$ 15 mag, con tiempos de exposición no menores a 15 s. Esto es posible debido a que las características dimensionales del CCD son usadas para tener una mejor definición del nivel del cielo, de esta manera la apertura elegida puede ser mucho más chica que en fotometría convencional, por lo que esta técnica es ideal para el estudio de objetos variables débiles de corto período. Otra característica importante de este método es que no se necesita una noche perfectamente clara para realizar las observaciones; en caso de no tenerla se agrega más resolución temporal a las mismas.

Designamos las magnitudes instrumentales del objeto de estudio, en nuestro caso los AGN, con la letra V, y las de las estrellas de control y comparación con las letras K y C, respectivamente. Se calculan luego las diferencias V$-$C y C$-$K, donde esta última se utiliza para: (i) detectar alguna variabilidad en las estrellas de control, (ii) medir la precisión

intrínseca del sistema instrumental, y (*iii*) proveer una comparación para determinar si el objeto de estudio es o no variable.

Es de suma importancia la correcta elección de las estrellas de control para obtener resultados confiables. El criterio original con el que han trabajado Romero et al. (1999, 2002) fue elegir dos grupos diferentes de estrellas en cada imagen y luego realizar un promedio de cada grupo para así obtener una media de estrellas de control y otra para estrellas de comparación. Sin embargo, en este trabajo decidimos utilizar las recomendaciones dadas por Howell et al. (1988), que consiste en seleccionar como estrella de control a una con la magnitud lo más similar posible al AGN, mientras que la estrella de comparación se elige de modo que sea un poco más brillante que los otros dos objetos, para que el factor de peso estadístico Γ (Howell et al. 1988) sea lo más cercano posible a 1. Algo de vital importancia es que tanto las estrellas de comparación como de control deben ser no variables.

Capítulo 4

Elementos de aprendizaje automático

El método clásico que se ha aplicado para estudiar la microvariabilidad de los AGN es el de los tests estadísticos, como el test F y el test C, los cuales dependen de algunos factores como el ruido. En general, no se han obtenido buenos resultados (Zibecchi 2013). Es por esto que decidimos utilizar ML como método alternativo a los tests estadísticos.

En esta sección presentamos los conceptos básicos de aprendizaje automático, específicamente aprendizaje automático supervisado que emplearemos a lo largo de la Tesis. En este método, los sistemas de ML aprenden cómo combinar entradas para producir predicciones útiles sobre datos nunca antes vistos por el modelo.

El aprendizaje automático consiste en un paradigma de programación diferente a la programación clásica. En esta última, los humanos introducimos reglas a un programa, los datos son procesados acorde a esas reglas y obtenemos respuestas. Por el contrario, con ML los humanos introducimos datos así como las respuestas que esperamos de esos datos, y obtenemos las reglas. Esas reglas son aplicadas a nuevos datos nunca antes vistos por el modelo para predecir respuestas, como se muestra en la Fig. 4.1.

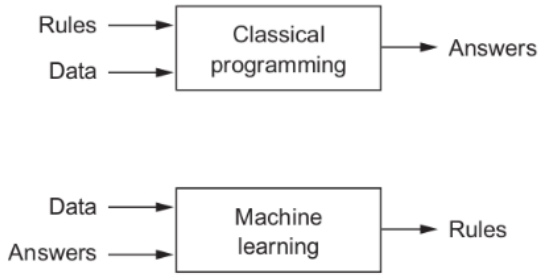

Figura 4.1. Aprendizaje automático: Un nuevo paradigma de programación. Figura tomada de Chollet, 2017.

Algunas definiciones importantes son:

- La etiqueta es el valor que estamos prediciendo (por ejemplo, la variable y de una regresión lineal simple).
- Un atributo (*feature*) es una dato de entrada (la variable x en un modelo de regresión lineal simple).

- Un ejemplo es una instancia de datos en particular $\vec{x}$. Estos se dividen en dos categorías: ejemplos etiquetados y ejemplos sin etiquetas.

Un ejemplo etiquetado incluye tanto atributos como la etiqueta correspondiente (x, y). Los ejemplos etiquetados son los utilizados para entrenar un modelo de aprendizaje automático supervisado. Luego ese modelo se usa para predecir la etiqueta de ejemplos sin etiquetas.

Un modelo define la relación entre los atributos y la etiqueta. Contiene dos fases:

- Entrenamiento: mostrar ejemplos etiquetados al modelo y permitir que este aprenda gradualmente las relaciones entre los atributos y la etiqueta.
- Inferencia: aplicar el modelo entrenado a ejemplos sin etiqueta. Es decir, utilizar el modelo para realizar predicciones.

El modelo más sencillo de aprendizaje supervisado en ML es la regresión lineal simple, que consiste en encontrar la línea recta que mejor se adapta a un conjunto de datos. A partir de este modelo simple podemos entender conceptos elementales. La ecuación para un modelo de este tipo es:

$$\hat{y} = b + w_1 x_1, \tag{4.1}$$

donde $\hat{y}$ es la etiqueta (resultado predicho), w_1 el peso del atributo 1, x_1 es un atributo (una entrada conocida por el modelo), y b es la ordenada al origen. Un modelo más sofisticado, que se base en tres atributos, usaría la siguiente ecuación:

$$\hat{y} = b + w_1 x_1 + w_2 x_2 + w_3 x_3. \tag{4.2}$$

Entrenar un modelo significa alimentarlo con ejemplos etiquetados para que pueda aprender los valores óptimos de los pesos y la ordenada al origen. En aprendizaje automático supervisado el algoritmo se optimiza al examinar varios ejemplos e intentar encontrar un modelo que minimice la pérdida (que definiremos a continuación).

La pérdida es un número que indica qué tan incorrecta fue la predicción del modelo en un solo ejemplo. Si la predicción del modelo es perfecta, la pérdida es cero, de lo contrario la pérdida es mayor. El objetivo de entrenar un modelo es encontrar un conjunto de pesos y ordenada al origen que, en promedio, tengan pérdidas bajas en todos los ejemplos.

Este es el objetivo de la función de pérdida, la cual toma las predicciones del modelo de ML y las etiquetas verdaderas (que es lo que deseamos que el modelo reproduzca) y calcula el apartamiento entre ambas. El uso de esta función nos ayuda a ajustar el valor de los pesos en la dirección que minimice la pérdida. Este ajuste es realizado por el optimizador, mecanismo por el cual el modelo se ajusta basado en los datos y la pérdida.

Un ejemplo de pérdida lo da el error cuadrático medio (ECM), el cual es el promedio de la pérdida al cuadrado de cada ejemplo.

$$ECM = \frac{1}{N} \sum_{(x,y)\in D} (y - \hat{y})^2, \tag{4.3}$$

donde D es el conjunto de ejemplos etiquetados, que son los pares (x, y), N es la cantidad de ejemplos y $\hat{y}$ es la salida del modelo.

La Fig. 4.2 contiene un diagrama del mecanismo de aprendizaje de ML. Se puede observar que el modelo toma uno o más atributos como entrada y devuelve una predicción $\hat{y}$ como resultado. Por ejemplo, en un modelo de regresión simple, con un solo atributo, debemos tomar valores iniciales para b y w_1 elegidos al azar. La parte "calcular pérdida" del diagrama es la función de pérdida que usará el modelo. Esta incorpora dos valores de entrada, la

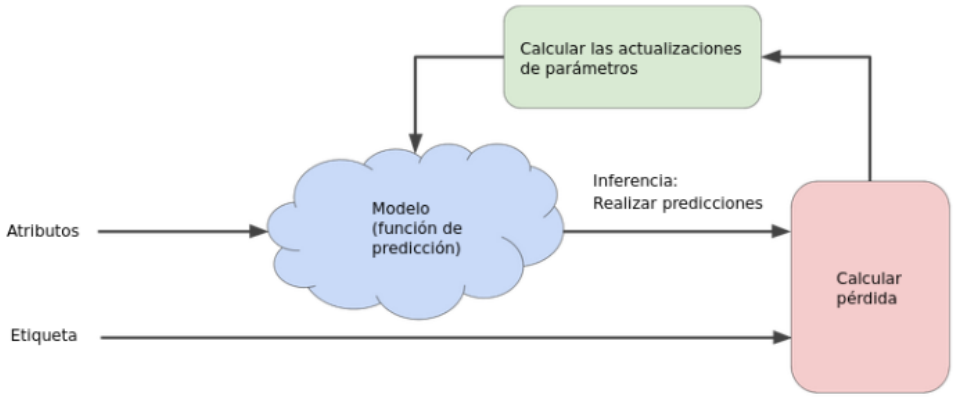

Figura 4.2. Un enfoque iterativo para entrenar un modelo.

predicción $\hat{y}$ para los atributos x y la etiqueta correcta y correspondiente a los atributos x. Finalmente llegamos a la parte de "actualizar parámetros" del diagrama, donde el sistema de aprendizaje automático examina el valor de la función de pérdida y genera valores nuevos para b y w_1.

En los problemas de regresión, la pérdida EMC con respecto a la ponderación w_1 es una función convexa, por lo tanto posee un solo mínimo, donde la pendiente es nula. Ese mínimo es hacia donde converge la función de pérdida.

El aprendizaje continúa iterando hasta que el algoritmo descubre los parámetros del modelo con la pérdida más baja posible. En general, itera hasta que la pérdida general deja de cambiar o cambia muy lentamente. Cuando eso ocurre decimos que el modelo ha convergido. En la sección de "actualizar parámetros" de la Fig. 4.2 es donde entra el juego el optimizador elegido. Un optimizador básico es el mecanismo iterativo llamado "descenso de gradientes".

La primera etapa en el descenso de gradientes es elegir un valor inicial para el peso w_1, el cual será al azar. Luego, el algoritmo calcula el gradiente de la curva de pérdida en el punto de partida, que para el caso de un solo peso es equivalente a la derivada. El algoritmo toma un paso en la dirección para la cual la derivada indica que la función decrece para reducir la pérdida, para determinar el siguiente punto a lo largo de la curva de la función de pérdida. Se repite este proceso y se acerca cada vez más al mínimo, como puede notarse en la Fig. 4.3.

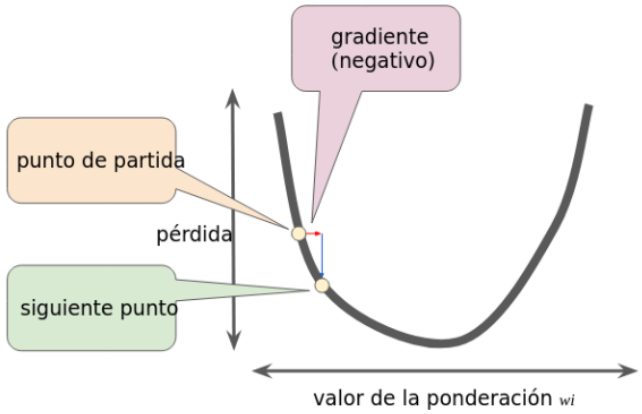

Figura 4.3. Un paso del gradiente nos mueve al siguiente punto en la curva de pérdida.

Un hiperparámetro es todo aquello que podemos ajustar, sintonizar o modificar para al-

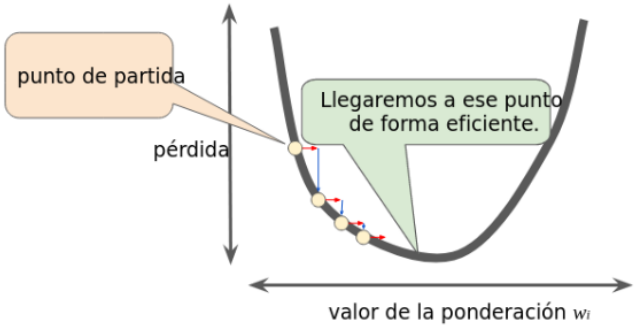

Figura 4.4. La tasa de aprendizaje es la correcta.

terar el desempeño del modelo. Un ejemplo de ellos es la tasa de aprendizaje, la cual debe ser elegida de modo tal que no sea ni muy pequeña (el aprendizaje llevaría mucho tiempo) ni muy grande (el siguiente punto rebotará del otro lado del mínimo y puede diverger). Los algoritmos de descenso de gradientes multiplican el gradiente por un escalar η, conocido como tasa de aprendizaje (*learning rate*), para determinar el siguiente punto. En la Fig. 4.4 puede observarse cómo los puntos van descendiendo lentamente hacia el mínimo de la función, lo cual es un indicio de que es una correcta elección de la tasa de aprendizaje para el descenso de gradientes.

Los pesos se actualizan luego de cada paso:

$$\begin{aligned} w_i' &= w_i - \eta \frac{\partial C}{\partial w_i}, \\ b' &= b - \eta \frac{\partial C}{\partial b_l}, \end{aligned} \tag{4.4}$$

donde C es la función de pérdida.

En el descenso de gradientes un lote es la cantidad total de ejemplos N que se usan para calcular el gradiente en una sola iteración. Sin embargo, los conjuntos de datos pueden tener miles de millones, o incluso cientos de millones, de ejemplos. En consecuencia, un lote puede ser enorme lo que puede causar que incluso una sola iteración tome un tiempo muy prolongado para calcularse.

La función de pérdida es la suma de las funciones de pérdida en cada ejemplo:

$$C = \frac{1}{N} \sum_x C_x. \tag{4.5}$$

Supongamos un caso de función de pérdida tridimensional. Para calcular el gradiente de la pérdida, obtenemos el gradiente en cada uno de los ejemplos y promediamos:

$$\nabla C = \frac{1}{N} \sum_x \nabla C_x. \tag{4.6}$$

donde ∇ indica el gradiente de la función.

El descenso de gradientes estocástico es un método para obtener el gradiente en mucho menos tiempo de cómputo que en el caso donde se utiliza un lote. La idea es elegir ejemplos al azar de nuestro conjunto de datos (minilote) para estimar un promedio general a partir

de otros mucho más pequeños. El término estocástico indica que los ejemplos que componen cada minilote son elegidos al azar.

Supongamos un minilote compuesto de m ejemplos $X_1, X_2, ..., X_m$. Se aproxima el gradiente de la función de pérdida por el gradiente en el minilote:

$$\frac{\sum_{j=1}^{m} \nabla C_{X_j}}{m} \approx \frac{\sum_X \nabla C_X}{N} = \nabla C. \tag{4.7}$$

Dado un minilote, se actualizan los pesos:

$$\begin{aligned} w_i' &= w_i - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_i}, \\ b' &= b - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l}, \end{aligned} \tag{4.8}$$

donde las sumas son realizadas sobre los ejemplos que se encuentran en el minilote.

Luego se elige otro conjunto de ejemplos aleatorios para conformar un nuevo minilote y se entrena con ellos. Finalmente, cuando el algoritmo ya examinó los datos de entrenamiento del conjunto de datos completo (compuesto por minilotes) decimos que se cumple una época. El gradiente de la función de pérdida C luego de una época se calcula mediante la media de los gradientes de cada minilote que se procesa durante la época. El número de épocas del modelo es otro de los hiperparámetros a ajustar del modelo.

La generalización hace referencia a la capacidad del modelo para adaptarse de manera adecuada a datos nuevos. Sin embargo, un modelo puede sobreajustarse a los datos de entrenamiento y no ajustarse a los datos nuevos, es decir, el modelo obtiene una pérdida baja durante el entrenamiento pero no se desempeña bien al predecir datos nuevos.

Por lo tanto, es conveniente dividir el conjunto de datos en dos subconjuntos: conjunto de entrenamiento y conjunto de prueba. Así un buen rendimiento en el conjunto de prueba es un indicador útil de un buen rendimiento en los datos nuevos.

Se puede reducir aún más la posibilidad de sobreajuste al particionar el conjunto de datos en 3 subconjuntos. Se añade el conjunto de validación, con el cual se evalúan los resultados del conjunto de entrenamiento en cada época. Finalizado el entrenamiento, se usa el conjunto de prueba para verificar la evaluación después de que el modelo haya pasado el conjunto de validación.

4.1. Árboles de decisión

Los árboles de decisión (*decision trees*, DT) son uno de los algoritmos comúnmente utilizados en diversos campos, como el ML, procesamiento de imágenes e identificación de patrones. Los DT son un modelo de clasificación utilizado habitualmente en Minería de Datos. Los nodos y ramas componen cada árbol (Fig. 4.5), cada nodo representa características de una categoría a clasificar y cada subconjunto de datos define un valor que puede tomar el nodo. Debido a su análisis sencillo y su precisión en múltiples formas de datos, los árboles de decisión han encontrado muchos campos de aplicación. Estos forman parte de la familia de algoritmos de aprendizaje supervisado dentro del ML, y pueden utilizarse para resolver problemas de regresión y clasificación.

Esta es una técnica en la que cualquier camino que comience desde la raíz se describe mediante una secuencia de división de datos según el valor del nodo, hasta que se alcanza un resultado booleano en cada nodo hoja (Fig. 4.6).

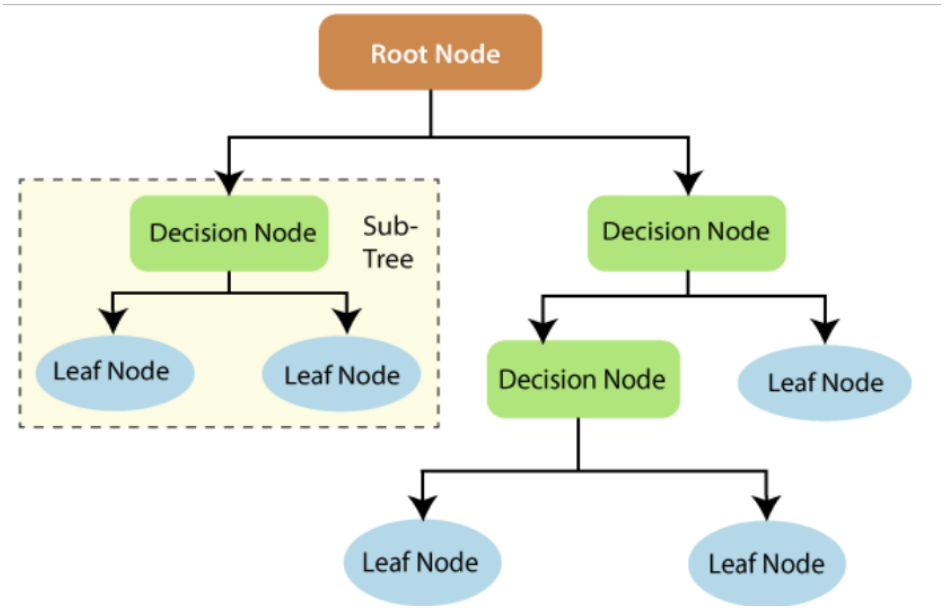

Figura 4.5. Estructura de un árbol de decisión. Imagen extraída de Jijo y Abdulazeez (2021).

El nodo situado en la parte superior del árbol se denomina nodo raíz. Este nodo tiene ramificaciones hacia los nodos situados por debajo. Los nodos del árbol que tienen ramas por debajo se denominan nodos internos o divisiones. Los nodos de la parte inferior del árbol se denominan nodos terminales u hojas. Nos centraremos en los árboles binarios, en los que cada nodo interno sólo tiene dos ramas. Por desgracia, los modelos de árbol suelen tener un poder predictivo limitado. Sin embargo, cuando se combinan varios modelos de árbol, como en los árboles empaquetados (Breiman, 1996), los bosques aleatorios (Breiman, 2001) o los algoritmos de refuerzo, que veremos en el capítulo siguiente, suelen tener una capacidad predictiva muy buena.

Algunas de las ventajas e inconvenientes que presentan este tipo de algoritmos se resumen a continuación (Nielsen, 2016).

Ventajas:

- Sencillos de interpretar
- Son relativamente rápidos de construir
- Pueden tratar de forma natural tanto datos continuos como categóricos
- Pueden tratar de forma natural los datos que faltan
- Son resistentes a los valores atípicos (*outliers*) en los datos de entrada
- Son invariantes bajo transformaciones monótonas de las entradas
- Realizan una selección implícita de variables
- Rueden capturar relaciones no lineales en los datos
- Pueden capturar interacciones de alto orden (relaciones entre tres o más variables) entre las variables de entrada
- Se adaptan bien a grandes conjuntos de datos

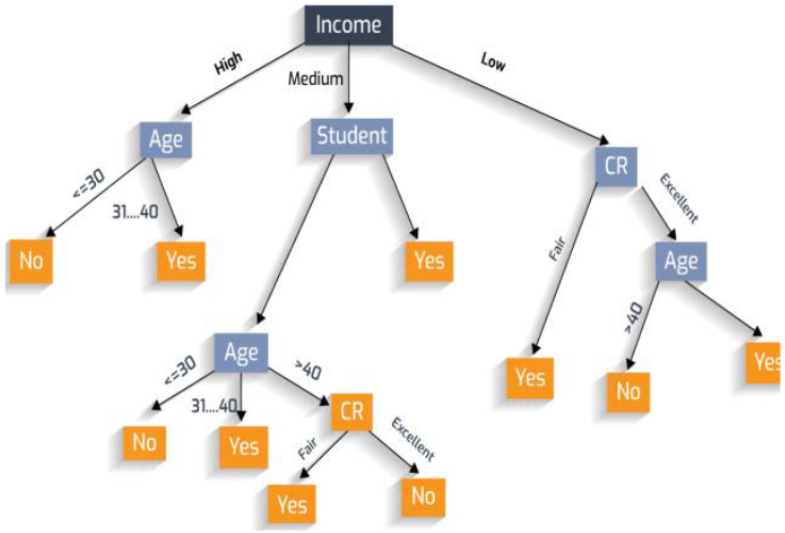

Figura 4.6. Estructura de un árbol de decisión.Imagen extraída de Jijo y Abdulazeez (2021).

Desventajas:

- Tienden a seleccionar predictores con un mayor número de valores distintos
- Pueden sobreajustarse cuando se enfrentan a predictores con muchas categorías
- Son inestables y tienen una varianza elevada
- Carecen de suavidad
- Tienen dificultades para captar la estructura aditiva, es decir, la contribución lineal de la variables
- Tienden a tener un rendimiento predictivo limitado

En el capítulo siguiente describiremos dos algoritmos de ML que utilizamos para este trabajo, daremos una descripción cualitativa de cómo funcionan y luego abordaremos algunos detalles matemáticos.

Capítulo 5

Algoritmos basados en árboles de decisión: *XGBoost* y *Random Forest*

5.1. Como funciona *XGBoost* para clasificación?

Antes de adentrarnos en los aspectos matemáticos más complejos de este algoritmo de ML, comenzaremos con una descripción cualitativa de su funcionamiento.

Como menciona el título de este capítulo, *XGBoost* está basado en árboles de decisión, específicamente en árboles de decisión reforzados (*boosting tree methods*). Los algoritmos de árboles de decisión reforzados son técnicas de aprendizaje automático que combinan varios árboles de decisión débiles (o de bajo rendimiento) para crear un modelo más robusto y preciso. El refuerzo o *boosting* es una técnica de ensamblaje de modelos que mejora la precisión al entrenar secuencialmente una serie de modelos débiles. Cada modelo se enfoca en corregir los errores cometidos por el modelo anterior.

Ahora aprenderemos cómo *XGBoost* construye estos árboles, para hacer esto utilizaremos un ejemplo muy sencillo. Imaginemos que tenemos 4 mediciones de dosis de medicamentos, que varían entre 0 y 20 mg (miligramos), las cuales están etiquetadas de manera binaria según si hicieron efecto, tomando un valor igual a 1, o si no hicieron efecto, tomando un valor igual a 0. El primer paso para ajustar *XGBoost* a los datos de entrenamiento es realizar una predicción inicial, que por defecto esta toma un valor de 0,5. Con esto se calculan los residuos de cada medida, que es la diferencia de las medidas con la predicción inicial. Para ajustar un árbol a los residuos se calcula el puntaje de similitud (*similarity score*) que se utiliza para decidir cómo dividir los datos en un nodo, eligiendo el abtributo que maximiza la ganancia en base al puntaje de similitud. Esta se calcula según:

$$similarity\ score = \frac{(\sum Residuo_i)^2}{\sum[probabilidad\ previa_i\ (1 - probabilidad\ previa_i)] + \lambda}, \tag{5.1}$$

donde λ es el parámetro de regularización (Nielsen 2016).

Se calcula el puntaje de similitud para la primera hoja llamada raíz. Suponiendo que las mediciones son las de la Tabla 5.1 este es igual a 0. Ahora debemos decidir si podemos hacer un mejor trabajo agrupando residuos similares si los dividimos en dos grupos. Elegimos

como primera aproximación un umbral igual a 15, ya que es el promedio de las últimas dos observaciones; por lo tanto, los tres residuos con dosis menores a 15 van a la hoja de la izquierda y el residuo con dosis mayor a 15 va a la hoja de la derecha (Fig. 5.2). Con esto, calculamos nuevamente los puntajes de similitud para cada hoja y luego la ganancia:

$$Ganancia = izquierda_{similitud} + derecha_{similitud} - raiz_{similitud}. \tag{5.2}$$

A la hora de elegir el umbral para dividir nuestro árbol se busca maximizar la ganancia. En nuestro ejemplo, considerando todas las divisiones posibles, resulta que el umbral de dosis 15 es el que maximiza la ganancia. Por lo tanto, Dosis < 15 será la primera rama de nuestro árbol. Ahora dividiremos esta rama en dos hojas. El umbral que maximizará la ganancia en este caso será Dosis < 5. Por lo tanto, nuestro primer árbol quedará como el de la Fig. 5.3.

Dejamos de hacer crecer este árbol porque limitamos la cantidad de niveles a 2. Sin embargo, *XGBoost* tiene un umbral para la cantidad mínima de residuos en cada hoja. Este umbral se determina calculando un parámetro llamado cobertura, que se define como el denominador del puntaje de similitud menos λ.

Medición	Probabilidad de efectividad
5 mg	0
8 mg	1
12 mg	1
18 mg	0

Tabla 5.1. Datos de entrenamiento para nuestro ejemplo.

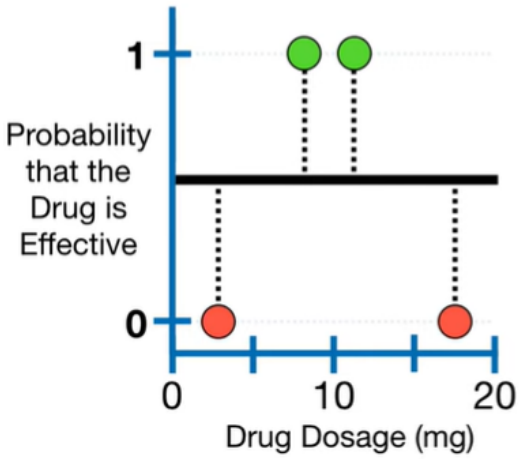

Figura 5.1. Gráfico de los datos de entrenamiento con sus etiquetas, en verde los casos positivos y en rojo los casos negativos. Las lineas punteadas muestran los residuos de cada observación.

Ahora veremos cómo podar este árbol, basado en los valores de las ganancias. Se calcula la diferencia entre la ganancia asociada con la rama más baja del árbol y el valor de un parámetro γ. Si esta diferencia es menor a cero, se elimina o poda la rama, si es mayor a cero no la podamos. Ahora bien, el valor de la ganancia depende de λ y valores de λ mayores que cero reducen la sensibilidad del árbol a las observaciones individuales al podarlos y combinarlas con otras observaciones.

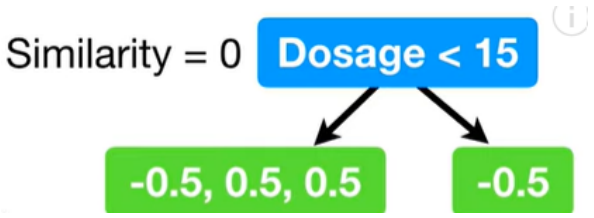

Figura 5.2. Calculamos la primera división para construir nuestro árbol.

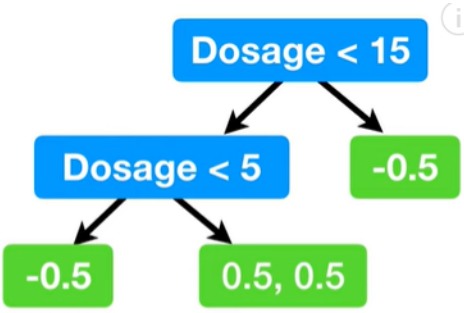

Figura 5.3. Primer árbol construido por *XGBoost* para nuestro ejemplo.

Teniendo nuestro árbol definitivo de la Fig. 5.3 determinemos los valores de salida para las hojas. Estos se calculan según la Ec. 5.1. Tomando γ y λ igual a cero obtenemos los valores de salida de cada hoja (Fig. 5.4).

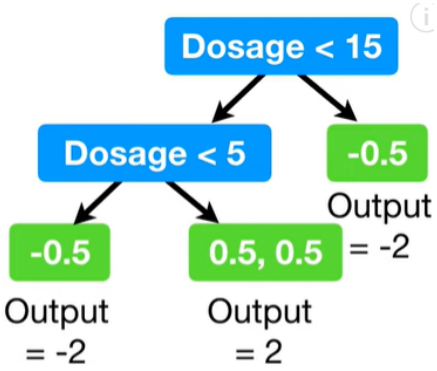

Figura 5.4. Primer árbol construido por *XGBoost* con los valores de salida para cada hoja.

Por último, hay que calcular las predicciones para cada hoja. *XGBoost* realiza nuevas predicciones comenzando con la predicción inicial. Luego, tomando esta predicción, la convertimos a un valor de probabilidades logarítmicas con la siguiente transformación:

$$\ln(\frac{p}{1-p}) = \ln(odds), \tag{5.3}$$

donde *odds* es el cociente de probabilidades y p es la probabilidad, le sumamos esto a la salida de nuestro primer árbol multiplicado por la tasa de aprendizaje η. Hacemos esto para cada salida y transformamos el logaritmo del cociente de probabilidades a probabilidades utilizando la función logística:

$$Probabilidad = \frac{e^{\ln(odds)}}{1 + e^{\ln(odds)}}. \tag{5.4}$$

Una vez calculadas estas probabilidades, a medida que vamos iterando y sumando árboles, los residuos se hacen cada vez más pequeños. Con estos nuevos residuos se construye el siguiente árbol, hasta que los residuos sean tan pequeños como queramos o se haya alcanzado el número máximo de árboles, que es un hiperparámetro del modelo.

En resumen, al construir árboles con *XGBoost* para clasificación calculamos puntajes de similitud y las ganancias para determinar cómo dividir los datos y así podar el árbol calculando la diferencia entre la ganancia y γ.

5.2. Métodos de refuerzo de árboles

Los algoritmos de árboles de decisión reforzados (*boosting tree methods*) son técnicas de aprendizaje automático que combinan varios árboles de decisión débiles (o de bajo rendimiento) para crear un modelo más robusto y preciso. El refuerzo o *boosting* es una técnica de ensamblaje de modelos que mejora la precisión al entrenar secuencialmente una serie de modelos débiles. Cada modelo se enfoca en corregir los errores cometidos por el modelo anterior.

Utilizar árboles como modelos de base para el refuerzo es una opción muy utilizada, teniendo en cuenta que los árboles tienen muchas ventajas que los árboles reforzados heredan y que la capacidad de predicción aumenta considerablemente con el refuerzo. El principal inconveniente de los modelos de árboles reforzados en comparación con los modelos de árbol único es que se pierde la mayor parte de la interpretabilidad (Nielsen, 2016).

5.2.1. Refuerzo de árbol de Newton

El refuerzo de árbol de Newton (*Newton tree boosting*, NTB) es un método de refuerzo para ajustar modelos de árbol aditivos utilizando el algoritmo de reforzamiento de Newton (Nielsen, 2016). En cada iteración m, este algoritmo trata de minimizar el criterio de la Ec. 5.4 (Nielsen, 2016)

$$J_m(\phi_m) = \sum_{i=1}^{n} L(y_i, \hat{f}^{(m-1)}(x_i) + \phi_m(x_i)), \tag{5.5}$$

donde L es la función de pérdida, $\hat{f}$ es el algoritmo de árboles, n la cantidad de nodos terminales y $\phi_m(x)$ es la función base que indica si la observación x_i pertenece al nodo terminal i.

Para los algoritmos de refuerzo de árboles, las funciones base son árboles

$$\phi_m = \sum_{j=1}^{T} w_{jm} I(x \in R_{jm}) \tag{5.6}$$

donde w_{jm} (los pesos) es el ajuste constante en cada nodo terminal de la región R_{jm} (Nielsen 2016), $I(x \in R_{jm})$ es la función que indica si la observación x_j pertenece al nodo terminal j y T es la cantidad de regiones rectangulares no superpuestas R_j.

El algoritmo de refuerzo de árbol de Newton es simplemente el refuerzo de Newton (Nielsen, 2016), en el que las funciones base son modelos de árbol. El refuerzo de Newton aproxima el criterio de la Ec. 5.4 mediante

$$\tilde{J}_m(\phi_m) = \sum_{i=1}^{n} \left[\hat{g}_m(x_i)\phi_m(x_i) + \frac{1}{2}\hat{h}_m(x_i)\phi_m(x_i)^2 \right], \tag{5.7}$$

que es una aproximación de segundo orden. Aquí $\hat{h}_m(x_i)$ es el hessiano empírico y $\hat{g}_m(x_i)$ es el gradiente. Tanto el hessiano como el gradiente son de la función de pérdida del modelo $\mathcal{R}(f)$ (Nielsen, 2016). Para una estructura de árbol fija, los o pesos vienen dados por

$$\tilde{w}_{jm} = -\frac{G_{jm}}{H_{jm}}, \quad j = 1, ..., T. \tag{5.8}$$

donde $G_{jm} = \sum_{i \in I_{jm}} \hat{g}_m(x_i)$ y $H_{jm} = \sum_{i \in I_{jm}} \hat{h}_m(x_i)$.

Aprender la estructura de un árbol equivale a buscar divisiones o nodos. Para cada división, se propone una serie de divisiones candidatas y se elige la maximiza la ganancia. Para el refuerzo del árbol de Newton, buscamos la división que minimice el criterio de la Ec. 5.6. Introduciendo los pesos de la ecuación anterior en la Ec. (5.6) y haciendo un pcoo de álgebra obtenemos

$$\tilde{J}_m(\tilde{\phi}_m) = -\frac{1}{2}\sum_{i=1}^{T} \frac{G_{jm}^2}{H_{jm}}. \tag{5.9}$$

Así, la ganancia utilizada para determinar las divisiones viene dada por

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G_{jm}^2}{H_{jm}} \right], \tag{5.10}$$

donde los subíndices L y R hacen referencia a las ramas izquierda y derecha respectivamente de la división.

Otro algoritmo similar es el refuerzo de árbol de gradientes (*gradient tree boosting*, GTB), en donde en este caso el método para la optimización numérica del espacio de funciones es el descenso de gradientes, en lugar del método de Newton (Nielsen, 2016).

5.2.2. *XGBoost*

El algoritmo de ML *XGBoost* esta basado en los algoritmos de refuerzos de árboles, con algunas características adicionales que lo hacen más robusto y efectivo. Mencionaremos algunos puntos importantes de mejoras de *XGBoost* con respecto a NTB y GTB.

Una de las principales ventajas es su capacidad para manejar la compensación (*trade-off*) entre sesgo y varianza de manera más efectiva. Esto se logra a través de varias mejoras, incluyendo:

- Regularización: *XGBoost* incorpora nuevos términos de regularización, lo que ayuda a evitar el sobreajuste del modelo. Esto es algo que GTB y NTB no manejan tan bien, permitiendo que *XGBoost* generalice mejor a nuevos datos.

- *Boosting* secuencial: A diferencia de otros métodos, *XGBoos*t implementa un algoritmo de *boosting* más eficiente y rápido, lo que resulta en una mejor optimización del modelo. Esto incluye una técnica llamada "*shrinkage*"(reducción) que ajusta el peso de cada árbol nuevo, mejorando la precisión y reduciendo la probabilidad de errores.

Algorithm 3: Newton tree boosting

Input : Data set $\mathcal{D}$.
A loss function L.
The number of iterations M.
The learning rate η.
The number of terminal nodes T

Initialize $\hat{f}^{(0)}(x) = \hat{f}_0(x) = \hat{\theta}_0 = \arg\min_{\theta} \sum_{i=1}^{n} L(y_i, \theta)$;

for m = *1,2,..,M* **do**

  $\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}^{(m-1)}(x)}$;

  $\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}^{(m-1)}(x)}$;

  Determine the structure $\{\hat{R}_{jm}\}_{j=1}^{T}$ by selecting splits which maximize $Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G_{jm}^2}{H_{jm}}\right]$;

  Determine the leaf weights $\{\hat{w}_{jm}\}_{j=1}^{T}$ for the learnt structure by $\hat{w}_{jm} = -\frac{G_{jm}}{H_{jm}}$;

  $\hat{f}_m(x) = \eta \sum_{j=1}^{T} \hat{w}_{jm} \mathrm{I}(x \in \hat{R}_{jm})$;

  $\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \hat{f}_m(x)$;

end

Output: $\hat{f}(x) \equiv \hat{f}^{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x)$

Figura 5.5. Algoritmo de refuerzo de árbol de Newton. Imagen extraída de Nielsen, 2016.

- Capacidad de manejo de datos faltantes: *XGBoost* maneja de manera nativa los datos faltantes, lo que mejora la robustez del modelo. GTB y NTB, en cambio, pueden requerir preprocesamiento adicional para tratar con los datos faltantes.

- Computación distribuida: *XGBoost* está diseñado para ser eficiente en términos de paralelización y computación distribuida. Esto permite que el entrenamiento del modelo sea más rápido y escalable en grandes conjuntos de datos, algo que GTB y NTB no logran con la misma eficiencia.

- Algoritmo de *boosting*: *XGBoost* utiliza una versión mejorada del algoritmo de *boosting* basado en gradientes, incluyendo la técnica de optimización "*second-order*"que utiliza la información de la segunda derivada, resultando en una convergencia más rápida y precisa del modelo.

Estos factores combinados hacen que *XGBoost* sea una opción superior en la mayoría de las competiciones de aprendizaje automático, al balancear mejor la precisión y la eficiencia en comparación con GTB y NTB.

5.3. *Random Forest*

El algoritmo de Bosque Aleatorio (*Random Forest*) se ha consolidado como una de las técnicas más populares en el campo del aprendizaje automático. Este método, desarrollado por Leo Breiman en 2001, se basa en la teoría del aprendizaje por conjuntos (*ensemble learning*), que consiste en combinar varios modelos de predicción para mejorar la precisión y robustez de los resultados (Breiman, 2001). *Random Forest* es particularmente eficaz para tareas de clasificación y regresión, gestionando tanto datos numéricos como categóricos. El funcionamiento del algoritmo se basa en dos conceptos principales: los árboles de decisión y el empaquetado (*bagging*).

Veremos ahora con un ejemplo cómo funciona este algoritmo. Supongamos que nuestro conjunto de datos de entrenamiento para este ejemplo es el de la Fig. 5.6, lo primero que hay que hacer es tomar una submuestra aleatoria de este conjunto de datos llamada submuestra *bootstrap*, que tendrá el mismo tamaño que el conjunto de datos original. Una submuestra *bootstrap* es un subconjunto de datos que se generan mediante un proceso conocido como muestreo de reemplazo, donde se seleccionan aleatoriamente observaciones del conjunto de datos original y una misma instancia puede ser seleccionada más de una vez. Con esto, construimos nuestra submuestra *bootstrap* que se muestra en la Fig. 5.7.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Figura 5.6. Conjunto de datos de entrenamiento para nuestro ejemplo.

Luego se crea un árbol con nuestra submuestra *bootstrap* pero solo utilizando un subconjunto m aleatorio de atributos en cada paso. Generalmente m es igual a la raíz cuadrada del número total de atributos en el conjunto de datos. Esta elección empírica ha demostrado ser efectiva en la práctica. Este enfoque reduce la varianza y mejora la generalización del modelo. Elijamos por ahora 2 atributos aleatoriamente para cada paso. De estas dos nos quedamos con una para nuestro nodo raíz, supongamos que elegimos la variable *Good Blood circ.* Luego hay que elegir nuevamente cómo dividir uno de los dos nodos resultantes, que al igual que para el nodo raíz, elegimos aleatoriamente dos atributos candidatos y nos quedamos con uno. Así vamos construyendo nuestro árbol pero en cada paso consideramos un subconjunto aleatorio de atributos como candidatos.

Una vez construido nuestro primer árbol, volvemos al primer paso y repetimos, construyendo una nueva submuestra *bootstrap* y construyendo el árbol considerando un subconjunto de atributos en cada paso o nodo. Hacemos esto cientos de veces (para el caso de un conjunto

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Figura 5.7. Submuestra aleatoria *bootstrap* tomada del conjunto de datos original.

de datos de entrenamiento grande). Con esto obtenemos muchos tipos de árboles distintos. Esta diversidad hace que los bosques aleatorios sean más efectivos que un solo árbol de decisión.

Ahora, hagamos una predicción. Suponemos que tenemos un nuevo paciente con determinados valores para los atributos de entrada. Seleccionamos el primer árbol y seguimos las ramas correspondientes según estos valores de entrada hasta llegar al final de la hoja, las cuales tienen los valores de salida o predicciones, que toman los valores 0 si es NO y 1 si es SI. Hacemos este procedimiento para todos los árboles creados y la predicción final se obtiene a través de la votación mayoritaria. Ahora hay que ver si estos árboles son útiles o no.

Cuando creamos la submuestra *bootstrap* en nuestro ejemplo, hubo una observación que no se utilizó. Estas se llaman datos fuera de la mochila (*out-of-bag*). Si el conjunto de datos de entrenamiento fuese más grande tendríamos más de una observación *out-of-bag*. Dado que nuestras observaciones *out-of-bag* no se utilizaron para entrenar los árboles, podemos usarlas para ver si los árboles clasifican con éxito o no. Hacemos esto para todos los árboles construidos utilizando todas nuestras observaciones *out-of-bag* y la etiqueta más votada será la predicción de nuestro bosque aleatorio y así podemos calcular la proporción de observaciones *out-of-bag* correctamente clasificadas, que es conocida como la precisión del bosque aleatorio. Las proporción de etiquetas clasificadas de manera errónea se conocen como error fuera de la bolsa (*out-of-bag-error*)

Ahora que sabemos medir la precisión de nuestro bosque aleatorio, podemos profundizar en cómo construir dicho bosque. Cuando creamos nuestro primer árbol utilizamos solamente dos atributos al seleccionar los nodos en cada paso, ahora podemos comparar el *out-of-bag-error* de un bosque aleatorio utilizando dos atributos en cada paso con un bosque aleatorio utilizando tres atributos en cada paso. Luego probamos con diferentes configuraciones y seleccionamos el bosque aleatorio más preciso.

Una de las principales ventajas de *Random Forest* es su capacidad para manejar grandes conjuntos de datos con alta dimensionalidad sin requerir una normalización previa. Además, es robusto ante datos ruidosos y ausentes, y puede encontrar interacciones no lineales entre las variables. Sin embargo, el entrenamiento de un algoritmo *Random Forest* puede ser computacionalmente costoso, especialmente para conjuntos de datos muy grandes (que no es

el caso de este trabajo). Asimismo, la interpretabilidad de los modelos individuales puede ser limitada, ya que cada árbol puede ser complejo y difícil de visualizar.

Finalmente, resumiendo en algunos simples pasos:

1) Creamos nuesta submuestra *bootstrap*.
2) Creamos nuestro bosque aleatorio.
3) Estimamos la precisión del bosque aleatorio creado.
4) Cambiamos el número de atributos utilizados en cada paso y creamos un nuevo bosque aleatorio.
5) Repetimos muchas veces esto y elegimos el bosque aleatorio que tenga la mejor precisión.

Capítulo 6

Resultados

6.1. Estudio de variabilidad

Utilizando los datos descritos anteriormente en la Sección 3.1 para el entrenamiento de los algoritmos, lo primero fue realizar un análisis exploratorio de datos (AED) y el resultado más relevante es el desbalance de los datos, de los 5225 AGN tenemos un 95.5 % no variables y un 4.5 % variables. Esto es un problema para los algoritmos de clasificación que usaremos que hay que tener en cuenta a la hora de entrenar los modelos (Ver Apéndice A). Para poder utilizar algoritmos de clasificación de ML necesitamos descartar la variable temporal, ya que es un problema de clasificación y no de *forecasting*. Para esto procedimos a realizar una ingeniería de atributos (*feature engineering*) calculando de cada serie de tiempo los siguientes estadísticos:

- μ
- σ
- β
- κ

donde μ es la media, σ la varianza, β el sesgo y κ la curtosis.

Esto lo hicimos para los datos sin los errores observacionales y para los datos con lodichoss errores restados o sumados, según una variable aleatoria. Es decir,

$$m_{\text{err}} = m + \lambda\delta \tag{6.1}$$

donde m es la magnitud observada, m_{err} es la magnitud con error incluido, δ es el error de la medición y $\lambda = \pm 1$ se elige aleatoriamente. Con esto tenemos entonces los siguientes atributos para nuestro modelo:

- μ
- σ
- β
- κ
- μ_{err}

- σ_{err}
- β_{err}
- κ_{err}
- Var

Aquí los subíndices err indican que están agregados los errores y Var es la variabilidad que es la variable objetivo o *target* que trataremos de predecir.

En la Fig. 6.1 se muestra la matriz de correlación lineal de estos atributos, que calcula la correlación de Pearson entre cada par de variables.

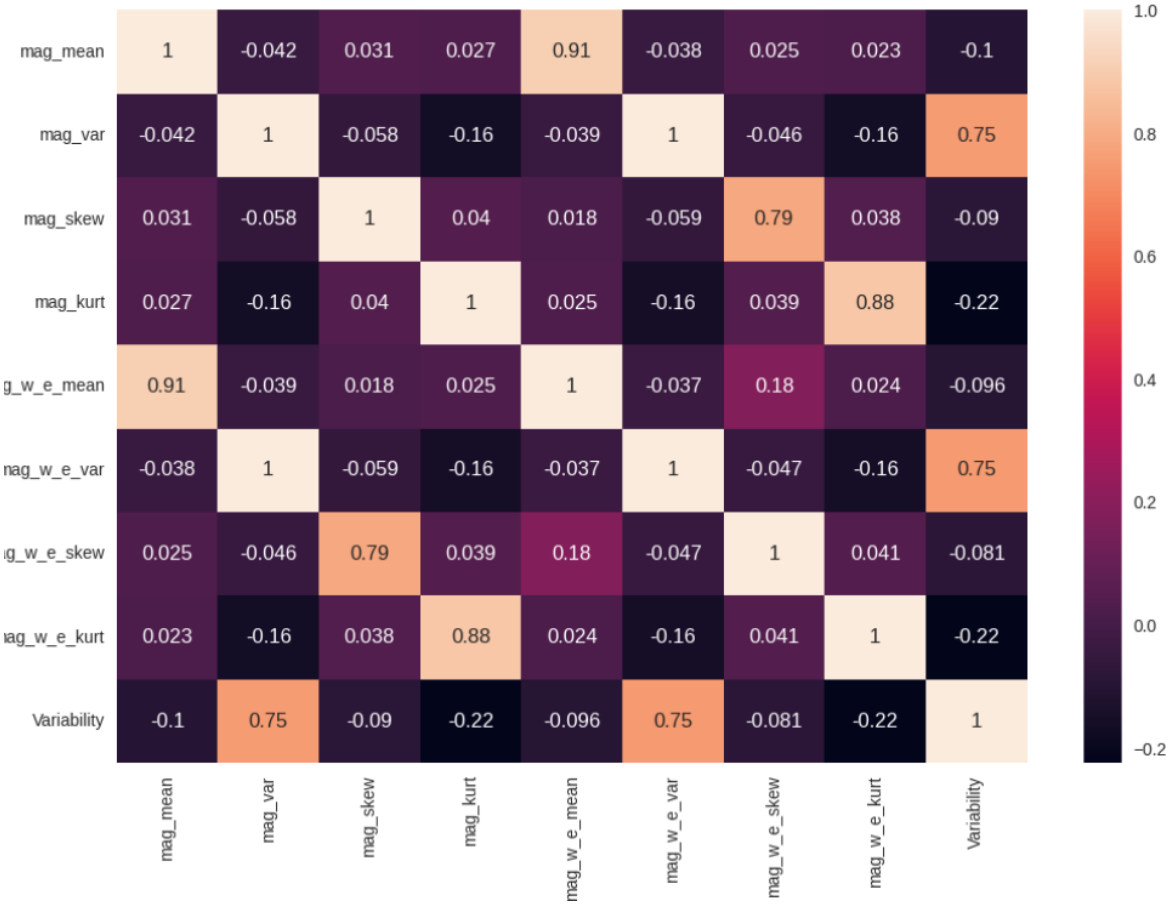

Figura 6.1. Matriz de correlación lineal de todos los atributos

6.1.1. Entrenamiento y selección del algoritmo de aprendizaje automático

Para seleccionar el mejor algoritmo de clasificación utilizamos el método AutoML. Este método se encarga de entrenar varios algoritmos diferentes simultáneamente y ver cuál es el que tiene mejores métricas de rendimiento, es decir, cuál es el que *a priori* clasificará mejor nuestras observaciones. Para nuestro caso, utilizamos únicamente los atributos sin los errores y le dimos al método AutoML algunos de los algoritmos de ML para clasificación más conocidos, entre los cuales se encuentran *XGBoost* y *Random Forest*.

Para hacer la elección del modelo, tuvimos en cuenta las métricas '*Accuracy*', '*Recall*' y 'F1' ya que lo que queremos es minimizar la cantidad de falsos positivos, las cuales tienen en cuenta lo siguiente:

- *Accuracy* (Precisión): Es la proporción de predicciones correctas realizadas por el modelo con respecto al total de predicciones. Se calcula como el número de predicciones correctas dividido por el número total de casos evaluados. Aunque es una métrica útil, puede ser engañosa si las clases están desbalanceadas.
- *Recall* (sensibilidad o tasa de verdaderos positivos): Mide la capacidad del modelo para identificar correctamente todas las muestras positivas. Se calcula como el número de verdaderos positivos (TP) dividido por la suma de los verdaderos positivos y los falsos negativos (FN). Es especialmente importante en situaciones donde detectar la clase positiva es crucial.
- F1 *Score* (Puntuación F1): Es la media armónica del *Precision* (Precisión) y *Recall* (Sensibilidad). La fórmula es:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

  El F1 *Score* es útil cuando se necesita un balance entre *Precision* y *Recall*, especialmente si los datos están desbalanceados.

Todas estas métricas toman valores entre 0 y 1, y valores cercanos a 1 quieren decir que el algoritmo está prediciendo bien.

Los resultados se muestran en la Fig. 6.2.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.9909	0.9935	0.9333	0.8770	0.9033	0.8986	0.8996	0.0410
rf	Random Forest Classifier	0.9903	0.9953	0.9333	0.8733	0.8998	0.8948	0.8967	0.2300
dt	Decision Tree Classifier	0.9882	0.9621	0.9333	0.8312	0.8778	0.8716	0.8741	0.0290
knn	K Neighbors Classifier	0.8868	0.8674	0.7600	0.2555	0.3805	0.3354	0.3973	0.0360
lr	Logistic Regression	0.8158	0.9168	0.8467	0.1781	0.2941	0.2370	0.3344	0.0230

Figura 6.2. Resultados del método AutoML. El resto son métricas que arroja el método por defecto que no tendremos en cuenta.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Extreme Gradient Boosting	0.9951	0.9992	0.9692	0.9265	0.9474	0.9448	0.9450

Figura 6.3. Métricas de rendimiento para el algoritmo de clasificación *XGBoost* luego de la búsqueda de hiperparámetros para su optimización.

Por último, seleccionamos el modelo XGBoost entrenado por el método AutoML e hicimos una búsqueda de hiperparámetros para optimizar el rendimiento del mismo, utilizando

el método *tune_model*, el cual se encarga de buscar los hiperparámetros que den las mejores métricas de rendimiento. Finalmente obtenemos las siguientes métricas de rendimiento para el modelo XGBoost optimizado (Figura 6.3). Aquí podemos ver que nuestro modelo entrenado funciona bien, es decir, clasifica correctamente la gran mayoría de los AGN sintéticos del conjunto de validación. Esto nos dice que el algoritmo es fiable y que podemos realizar predicciones sobre nuestras observaciones de AGN con seguridad.

También obtenemos la matriz de confusión del modelo (Fig. 6.4), la cual es una herramienta utilizada para evaluar el rendimiento de un modelo de clasificación. Esta matriz permite visualizar las predicciones correctas e incorrectas realizadas por el modelo y ayuda a entender qué tipos de errores está cometiendo.

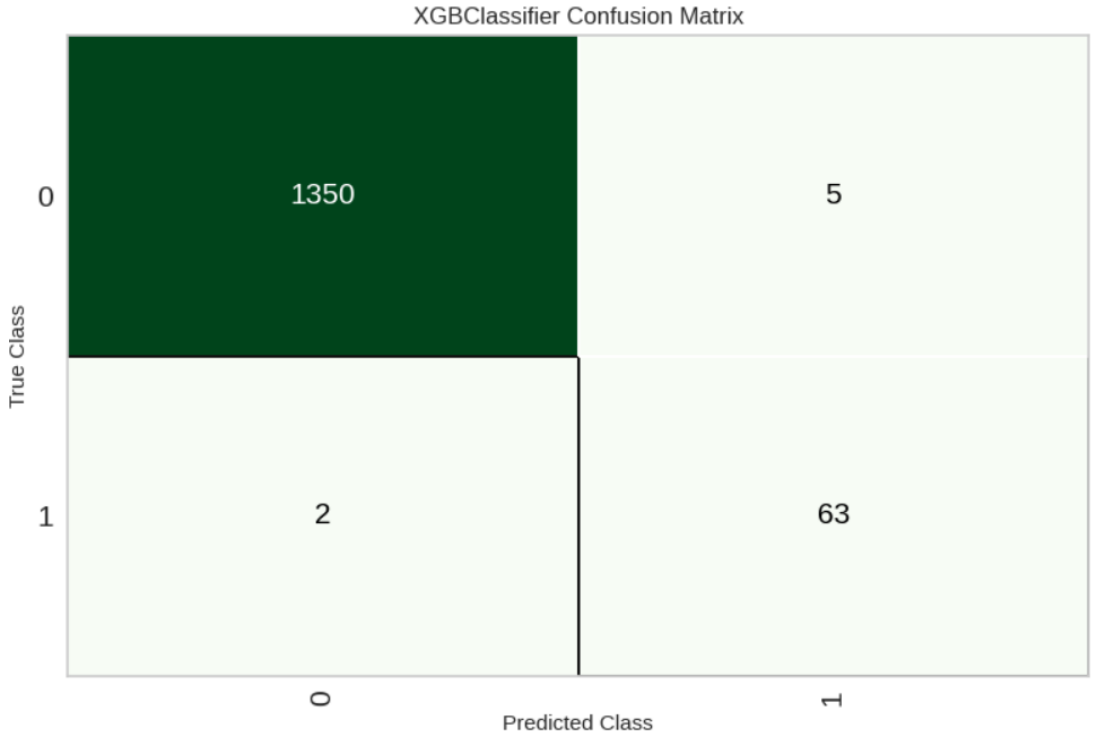

Figura 6.4. Matriz de confusión del modelo *XGBoost*.

Aquí se muestra una representación esquemática de una matriz de confusión:

TN	FN
FP	TP

En este cuadro TN (*True negatives*) representan los datos que fueron clasificados como falsos y que eran falsos, FN (*False negatives*) son los datos que fueron clasificados como falsos y eran positivos, FP (*False positives*) son los que fueron clasificados como falsos y eran positivos y TP (*True positives*) son los que fueron clasificados como positivos y eran positivos. Esta matriz da un resumen de la asertividad a la hora de clasificar los datos del conjunto de validación.

Por otro lado, tenemos la importancia de cada atributo (Fig. 6.5) a la hora de entrenar el modelo. Esto calcula la contribución marginal de cada atributo a la predicción para cada observación, proporcionando una explicación de la importancia de los atributos.

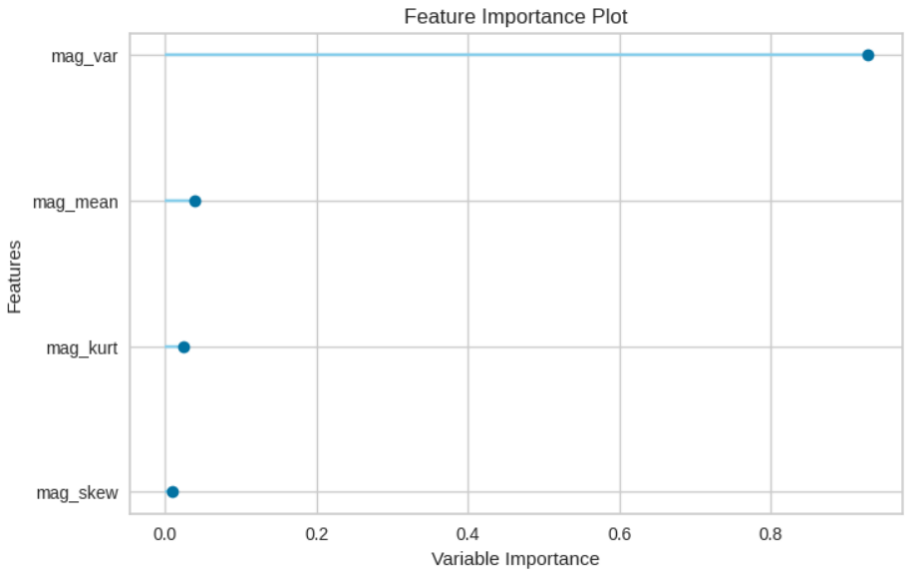

Figura 6.5. Nivel de importancia de cada uno de los atributos a la hora de entrenar el modelo *XGBoost*.

6.1.2. Clasificación de las observaciones de AGN

Para las observaciones de AGN detalladas en la Sección 3.2, calculamos los mismos estadísticos utilizados para entrenar el algoritmo de aprendizaje automático. De aquí obtenemos los atributos mencionados al principio de este Capítulo, con y sin errores. En un primer paso, nos quedamos únicamente con los atributos sin errores, al igual que en el entrenamiento, y con esto generamos las predicciones. Los resultados se muestran en la Tabla 6.1, y viendo las métricas de rendimiento y la matriz de confusión del modelo podemos asegurar que estos resultados son confiables.

6.1.2.1. Errores

Un método comúnmente utilizado en aprendizaje automático para tratar los errores a la hora de utilizar un modelo de clasificación consiste en realizar una nueva predicción teniendo

en cuenta los errores de cada uno de los atributos utilizados por el modelo, es decir, realizando una predicción con las magnitudes con errores, calculados según la Ec. (6.1). Con esto, comparamos los resultados con respecto a la predicción de las magnitudes sin errores y vemos si cambian o no. Si no cambian las predicciones, podemos decir que nuestro modelo de aprendizaje automático no es susceptible a errores de ese orden. Entonces, con los siguientes atributos

- μ_{err}
- σ_{err}
- β_{err}
- κ_{err}

realizamos una nueva predicción y obtuvimos exactamente los mismos resultados que los que se muestran en la Tabla 6.1. Con esto, podemos decir que nuestro modelo de ML no es susceptible a dichos errores observacionales.

6.2. Estudio de periodicidad

Análogamente a lo realizado en la sección anterior, entrenamos un algoritmo de aprendizaje automático para clasificar la periodicidad de los AGN variables. Al igual que antes tenemos un desbalanceo en los datos en cuanto a la cantidad de AGN periódicos y no periódicos, siendo solo el 0,5 % periódicos. Esto es un problema mucho más grave que el caso anterior, ya que aquí la muestra de AGN periódicos es demasiado pequeña y puede no ser suficiente para generalizar el problema.

Siguiendo los mismos pasos mencionados en la Sección anterior, tenemos los siguientes atributos para entrenar el modelo:

- μ
- σ
- β
- κ

y calculamos los atributos con errores nuevamente según la Ec. (6.1).

6.2.1. Entrenamiento y selección del modelo de aprendizaje automático

Nuevamente utilizamos el método AutoML para escoger el mejor algoritmo para nuestro problema, seleccionando una vez más entre algunos de los algoritmos de ML para clasificación más conocidos, entre los cuales se encuentran *XGBoost* y *Random Forest*.

Los resultados del método AutoML se muestran en la Fig. 6.6. En este caso el mejor modelo es un *Random Forest* y nuevamente optimizamos este modelo que seleccionó el método AutoML y realizamos una búsqueda de hiperparámetros utilizando el método *tune_model*. Finalmente obtenemos las métricas de rendimiendo para el modelo *Random Forest* optimizado que se muestran en la Fig. 6.7. Una vez más, podemos ver que nuestro modelo entrenado funciona bien, y que clasifica correctamente la mayoría de los AGN sintéticos del conjunto de validación.

Así también, en la Fig. 6.8 se muestra el nivel de importancia de cada atributo a la hora de entrenar el modelo y la matriz de confusión en la Fig. 6.9.

6.2.2. Clasificación de las observaciones de AGN

Siguiendo los mismos pasos descritos en la Sección 6.1.2, utilizando fotometría diferencial y los atributos sin errores, realizamos una predicción sobre las observaciones de AGN donde los resultados se muestran en la Tabla 6.2. Si bien las métricas de rendimiento para este algoritmo son buenas, la cantidad de datos de AGN periódicos es insuficiente para generalizar el problema, por lo que estos resultados son poco confiables.

6.2.2.1. Errores

Utilizando el mismo método mencionado en la Sección 6.1.2.1, nos quedamos con los atributos con errores

- μ_{err}
- σ_{err}
- β_{err}
- κ_{err}

para ver si nuestro algoritmo entrenado es susceptible a los mismos. Los resultados fueron los mismos a los de la Tabla 6.2.

El código con los detalles de este trabajo se encuentra en el siguiente repositorio.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9989	0.9997	0.7500	0.8000	0.7667	0.7663	0.7703	0.1970
xgboost	Extreme Gradient Boosting	0.9986	0.9997	0.7500	0.7667	0.7467	0.7461	0.7519	0.0480
dt	Decision Tree Classifier	0.9981	0.8496	0.7000	0.7167	0.6800	0.6792	0.6931	0.0250
knn	K Neighbors Classifier	0.9967	0.9492	0.9000	0.5833	0.6900	0.6885	0.7136	0.0370
lr	Logistic Regression	0.9836	0.9948	0.9000	0.2321	0.3597	0.3551	0.4440	0.2560

Figura 6.6. Resultados del método AutoML para la clasificación de periodicidad.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Random Forest Classifier	0.9981	0.9990	0.6250	1.0000	0.7692	0.7683	0.7898

Figura 6.7. Métricas de rendimiento para el algoritmo de clasificación *Random Forest* luego de la búsqueda de hiperparámetros para su optimización.

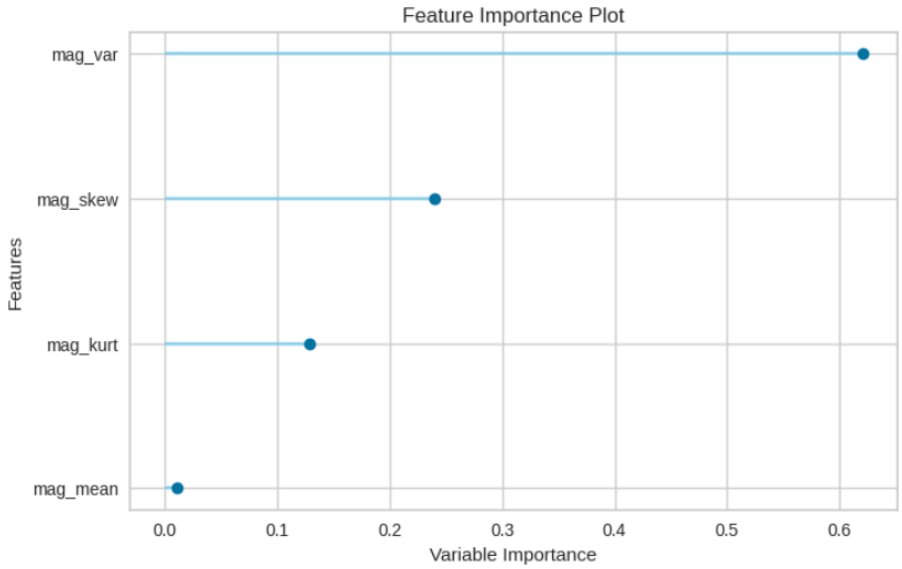

Figura 6.8. Nivel de importancia de cada una de los atributos a la hora de entrenar el modelo Random Forest.

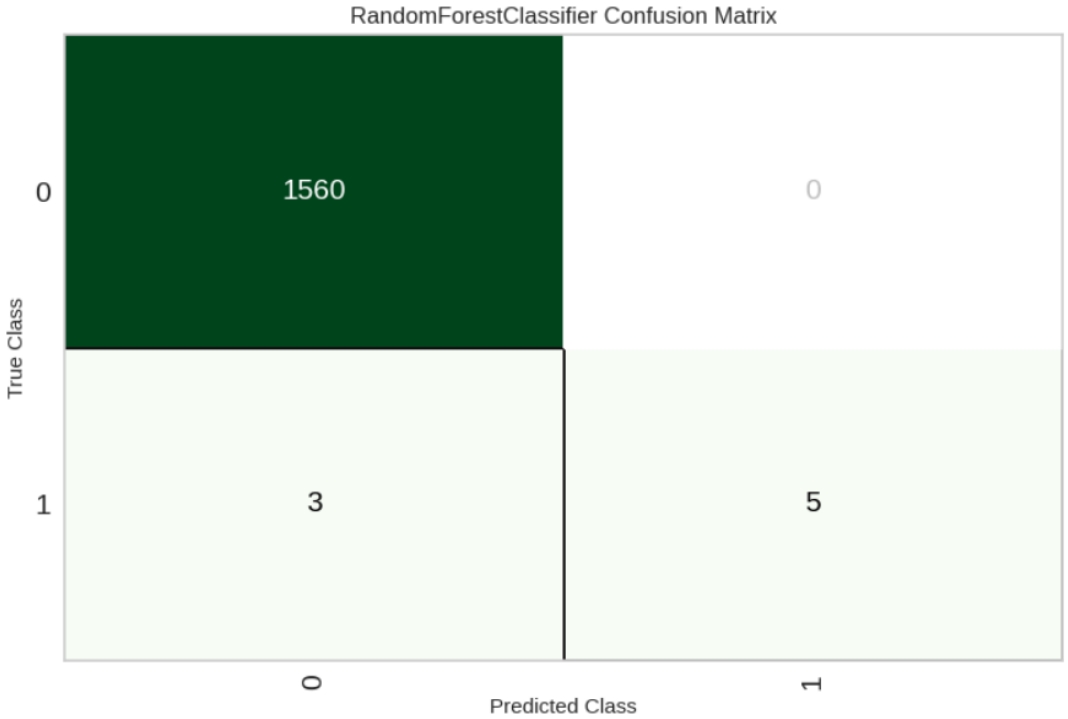

Figura 6.9. Matriz de confusión del modelo *Random Forest*.

AGN/Observación	Cant. observaciones	Predicción
1424R-140415	9	no variable
1424V-140415	9	no variable
0521R-111215	20	no variable
1116R-130415	15	no variable
1116R-140415	15	variable
1116V-130415	15	variable
1116V-140415	12	variable
1229V-120415	13	no variable
1229R-120415	13	no variable
1958V-230414	11	no variable
1958R-240414	10	no variable
1958V-240414	10	no variable
1958R-230414	11	no variable
1510V-120415	15	no variable
1510R-080519	15	no variable
1510V-080519	15	no variable
1510V-040419	8	no variable
1510V-130415	12	no variable
1510R-070519	17	no variable
1510R-090519	13	no variable
1510V-090519	13	no variable
1510V-060419	8	no variable
1510R-130415	13	no variable
1510R-050419	15	no variable
1510V-100519	13	no variable
1510V-050419	15	no variable
1510R-100519	14	no variable
1510V-070519	17	no variable
1510R-060419	8	no variable
1510R-070419	12	no variable
1510V-070419	12	no variable
1510R-140415	12	no variable
1510R-120415	15	no variable
1510V-140415	12	no variable
1510R-040419	7	no variable
1256R-230414	7	no variable
1256V-230414	7	no variable
0208R-130815	9	variable
0208R-150915	6	no variable
0208V-170915	8	no variable
0208V-150915	6	variable
0208R-170915	8	no variable
0208V-130815	9	variable
2005R-250819	25	no variable
2005R-100519	8	no variable
2005R-240819	26	variable
2005V-230819	21	no variable
2005R-230819	22	no variable
2005R-010919	22	no variable
2005R-260819	26	no variable
2005V-010919	22	variable
2005V-120815	24	no variable

AGN/Observación	Cant. observaciones	Predicción
2005R-020919	23	no variable
2005R-030919	24	no variable
2005V-090519	7	no variable
2005V-030919	23	no variable
2005R-310819	22	variable
2005V-310819	22	no variable
2005V-100519	8	no variable
2005V-260819	26	no variable
2005R-120815	25	no variable
2005V-250819	24	variable
2005R-090519	7	no variable
2005V-020919	23	no variable
2005V-240819	26	variable
1917R-250414	13	no variable
1917V-260414	10	no variable
1917R-260414	11	no variable
1917V-250414	13	no variable
2310V-170915	11	no variable
2310R-170915	11	no variable
0847R-230414	3	no variable
0847R-270414	7	no variable
0847V-220414	12	no variable
0847V-260414	3	no variable
0847R-250414	5	no variable
0847V-270414	7	no variable
0847R-220414	12	no variable
0847R-260414	3	no variable
0847V-250414	5	no variable
0847V-240414	5	no variable
0847R-240414	5	no variable
0847V-230414	3	no variable
1127V-100415	15	no variable
1127R-120415	9	no variable
1127R-100415	15	no variable
1127V-120415	9	no variable
0414V-281116	12	no variable
0414R-271116	13	no variable
0414V-271116	13	no variable
0414R-281116	11	no variable
1443V-250414	25	no variable
1443R-270414	14	no variable
1443V-240414	43	no variable
1443R-250414	25	no variable
1443R-240414	43	no variable
1443V-270414	15	no variable
2149V-120815	4	variable
2149V-170915	5	no variable
2149R-120815	3	no variable
2149R-170915	5	no variable

AGN/Observación	Cant. observaciones	Predicción
2126R-170915	12	no variable
2126V-170915	12	no variable
2126R-150915	14	no variable
2126V-150915	14	no variable
2155R-150915	14	no variable
2155V-130815	19	no variable
2155R-130815	19	no variable
2155V-150915	14	no variable

Tabla 6.1. Clasificación del algoritmo de aprondizaje automático *XGBoost* sobre las observaciones de AGN, con respecto a su variabilidad.

AGN/Observación	Cant. observaciones	Predicción
1424R-140415	9	no periódico
1424V-140415	9	no periódico
0521R-111215	20	no periódico
1116R-130415	15	no periódico
1116R-140415	15	no periódico
1116V-130415	15	no periódico
1116V-140415	12	no periódico
1229V-120415	13	no periódico
1229R-120415	13	no periódico
1958V-230414	11	no periódico
1958R-240414	10	no periódico
1958V-240414	10	no periódico
1958R-230414	11	no periódico
1510V-120415	15	no periódico
1510R-080519	15	no periódico
1510V-080519	15	no periódico
1510V-040419	8	no periódico
1510V-130415	12	no periódico
1510R-070519	17	no periódico
1510R-090519	13	no periódico
1510V-090519	13	no periódico
1510V-060419	8	no periódico
1510R-130415	13	no periódico
1510R-050419	15	no periódico
1510V-100519	13	no periódico
1510V-050419	15	no periódico
1510R-100519	14	no periódico
1510V-070519	17	no periódico
1510R-060419	8	no periódico
1510R-070419	12	no periódico
1510V-070419	12	no periódico
1510R-140415	12	no periódico
1510R-120415	15	no periódico
1510V-140415	12	no periódico
1510R-040419	7	no periódico
1256R-230414	7	no periódico
1256V-230414	7	no periódico
0208R-130815	9	no periódico
0208R-150915	6	no periódico
0208V-170915	8	no periódico
0208V-150915	6	no periódico
0208R-170915	8	no periódico
0208V-130815	9	no periódico
2005R-250819	25	no periódico
2005R-100519	8	no periódico
2005R-240819	26	no periódico
2005V-230819	21	no periódico
2005R-230819	22	no periódico
2005R-010919	22	no periódico
2005R-260819	26	no periódico
2005V-010919	22	no periódico
2005V-120815	24	no periódico

AGN/Observación	Cant. observaciones	Predicción
2005R-020919	23	no periódico
2005R-030919	24	no periódico
2005V-090519	7	no periódico
2005V-030919	23	no periódico
2005R-310819	22	no periódico
2005V-310819	22	no periódico
2005V-100519	8	no periódico
2005V-260819	26	no periódico
2005R-120815	25	no periódico
2005V-250819	24	no periódico
2005R-090519	7	no periódico
2005V-020919	23	no periódico
2005V-240819	26	no periódico
1917R-250414	13	no periódico
1917V-260414	10	no periódico
1917R-260414	11	no periódico
1917V-250414	13	no periódico
2310V-170915	11	no periódico
2310R-170915	11	no periódico
0847R-230414	3	no periódico
0847R-270414	7	no periódico
0847V-220414	12	no periódico
0847V-260414	3	no periódico
0847R-250414	5	no periódico
0847V-270414	7	no periódico
0847R-220414	12	no periódico
0847R-260414	3	no periódico
0847V-250414	5	no periódico
0847V-240414	5	no periódico
0847R-240414	5	no periódico
0847V-230414	3	no periódico
1127V-100415	15	no periódico
1127R-120415	9	no periódico
1127R-100415	15	no periódico
1127V-120415	9	no periódico
0414V-281116	12	no periódico
0414R-271116	13	no periódico
0414V-271116	13	no periódico
0414R-281116	11	no periódico
1443V-250414	25	no periódico
1443R-270414	14	no periódico
1443V-240414	43	no periódico
1443R-250414	25	no periódico
1443R-240414	43	no periódico
1443V-270414	15	no periódico
2149V-120815	4	no periódico
2149V-170915	5	no periódico
2149R-120815	3	no periódico
2149R-170915	5	no periódico

AGN/Observación	Cant. observaciones	Predicción
2126R-170915	12	no periódico
2126V-170915	12	no periódico
2126R-150915	14	no periódico
2126V-150915	14	no periódico
2155R-150915	14	no periódico
2155V-130815	19	no periódico
2155R-130815	19	no periódico
2155V-150915	14	no periódico

Tabla 6.2. Clasificación del algoritmo de aprendizaje automático *Random Forest* sobre las observaciones de AGN, con respecto a su periodicidad.

Capítulo 7

Conclusiones

Observando la Fig. 6.3 que muestra las métricas de rendimiento del algoritmo XGBoost luego de la búsqueda de hiperparámetros para su optimización podemos decir que este modelo obtuvo un gran desempeño en el conjunto de datos de validación, por lo que podemos asegurar que el modelo obtendrá predicciones confiables. Luego, en la Tabla 6.1 observamos que se clasificaron como variables las siguientes observaciones:

- AGN-Observación: 1116R-140415
- AGN-Observación: 1116R-140415
- AGN-Observación: 1116V-140415
- AGN-Observación: 0208R-130815
- AGN-Observación: 0208V-150915
- AGN-Observación: 0208V-130815
- AGN-Observación: 2005R-240819
- AGN-Observación: 2005V-010919
- AGN-Observación: 2005R-310819
- AGN-Observación: 2005V-240819
- AGN-Observación: 2149V-120815

Luego, viendo los resultados de la Sección 6.1.2.1 haciendo una predicción con los atributos con errores, es decir, utilizando las observaciones con su error sumado, vemos que se obtienen los mismos resultados que en la Tabla 6.1, por lo que podemos concluir aquí que nuestro modelo de ML entrenado no es susceptible a dichos errores observacionales.

Por otro lado, en la Fig. 6.7 vemos que las métricas del modelo *Random Forest* para el problema de clasificación de periodicidad son buenas, e incluso los modelos de regresión logística y KNN tienen un *Recall* mejor que *Random Forest*, sin embargo la cantidad de datos de AGN periódicos es insuficiente para poder generalizar el problema, por lo que estos modelos son poco fiables.

Por último, comparamos nuestro modelo para la clasificación de variabilidad con un test estadístico de Fisher, a fin de ver qué método funciona mejor. En la Fig. 6.4 se muestra que

de los 1420 AGN del conjunto de validación solamente se clasificaron 7 mal, donde 5 AGN se clasificaron variables cuando no lo eran y 2 se clasificaron no variables cuando sí lo eran. Por el otro lado, el test de Fisher realizado sobre todos los AGN sintéticos, utilizando un nivel de significancia del 1 %, clasificó 1129 AGN como variables sobre 5225. De estos 1129, 225 son variables y el resto no. Por otro lado, no clasificó como variable ningún AGN que no lo sea, por lo que este tipo de test no tiene problema para encontrar la variabilidad sino más bien su punto débil está en distinguir los objetos no variables.

Por lo tanto, con estos resultados, podemos concluir que nuestro algoritmo de clasificación de variabilidad obtiene mejores resultados que los test estadísticos clásicos.

7.1. Trabajo a futuro

El objetivo de fondo de esta tesis es introducir las herramientas de ML para problemas de clasificación. Los algoritmos que utilizamos si bien son robustos y confiables tienen también muchos problemas, sobre todo cuando hay un desbalance importante en los datos por lo que un punto muy importante para poder avanzar en el problema de periodicidad es conseguir más datos de AGN periódicos.
Otra manera de hacer este tipo de clasificaciones es utilizar aprendizaje profundo (*Deep learning*) (LeCun et al. 2015), esto es, utilizar redes neuronales (*neural networks*, NN) para analizar imágenes de curvas de luz. Estos algoritmos son mucho más exitosos y ampliamente más utilizados en el ámbito científico y tecnológico para tareas complejas. Si bien aquí proponemos utilizar las NN para clasificar AGN en variables y no variables o periódicos y no periódicos, esto se podría generalizar tomando imágenes de curvas de luz de todo tipo de objetos y realizar tareas de clasificación múltiple para identificar y clasificar objetos astrofísicos. Así, de una manera muy sencilla, a partir de una imagen de una curva de luz que *a priori* no se sabe a qué objeto pertenece, en muy poco tiempo el algoritmo de NN podría clasificarla e identificar qué tipo de objeto es, y si es variable o periódico.

La aplicabilidad del aprendizaje automático, en especial del aprendizaje profundo, es mucho más amplia que para una tarea de clasificación múltiple. Hay muchos escenarios en donde esas técnicas se pueden aplicar y seguro darán buenos resultados.

Apéndice A

Resultados y parámetros de los modelos de ML

En este apéndice mostraremos algunos resultados y parámetros del método AutoML y de los modelos de clasificación usados en el trabajo, de manera que pueda reproducirse el trabajo realizado.

A.1. Parámetros del AutoML

Como mencionamos en el Capítulo 6, el método AutoML se encarga de entrenar simultáneamente diferentes algoritmos de ML dado un conjunto de datos. Este método tiene parámetros que se pueden modificar para el entrenamiento de los modelos, en la Fig. A.1 se muestran los parámetros del método que utilizamos para entrenar nuestro algoritmo *XGBoost* y en la Fig. A.2 los parámetros para el *Random Forest* Forest. Un parámetro es el *Fix imbalance method* que toma el valor *SMOTE (Synthetic Minority Oversampling Technique)*. Este es un método para arreglar el problema del desbalance de datos y consiste en la generación de muestras sintéticas para la clase minoritaria.

A.2. Hiperparámetros de los modelos

La búsqueda de hiperparámetros o *hypertuning* es el proceso de reentrenar repetidamente un modelo con diferentes hiperparámetros para buscar la combinación de los mismos que maximicen las métricas de rendimiento que prioricemos. Como se menciona en el Capitulo 6, las métricas que buscaremos maximizar son la precisión, sensibilidad (*Recall*) y la puntuación F1. Luego de aplicar el método AutoML para encontrar el mejor modelo realizamos una búsqueda de hiperparámetros con el método *tune_model* que hace esto de forma automática, y como resultado obtenemos los hiperparámetros para el modelo *XGBoost* que se muestran en la Fig. A.3, y en la Fig. A.4 se muestran los hiperparámetros encontrados para el modelo *Random Forest*.

Description	Value
Session id	6684
Target	Variability
Target type	Binary
Original data shape	(4732, 5)
Transformed data shape	(7744, 5)
Transformed train set shape	(6324, 5)
Transformed test set shape	(1420, 5)
Numeric features	4
Preprocess	True
Imputation type	simple
Numeric imputation	mean
Categorical imputation	mode
Fix imbalance	True
Fix imbalance method	SMOTE
Fold Generator	StratifiedKFold
Fold Number	10
CPU Jobs	-1
Use GPU	False
Log Experiment	False
Experiment Name	clf-default-name
USI	3456

Figura A.1. Parámetros de configuración del método AutoML para el problema de variabilidad. Aquí se utilizaron los valores por defecto del método, salvo el parámetro *Fix imbalance method*.

Description	Value
Session id	3492
Target	Periodicity
Target type	Binary
Original data shape	(5225, 5)
Transformed data shape	(8848, 5)
Transformed train set shape	(7280, 5)
Transformed test set shape	(1568, 5)
Numeric features	4
Preprocess	True
Imputation type	simple
Numeric imputation	mean
Categorical imputation	mode
Fix imbalance	True
Fix imbalance method	SMOTE
Fold Generator	StratifiedKFold
Fold Number	10
CPU Jobs	-1
Use GPU	False
Log Experiment	False
Experiment Name	clf-default-name
USI	7daa

Figura A.2. Parámetros de configuración del método AutoML para el problema de periodicidad. Aquí se utilizaron los valores por defencto del método, salvo el parámetro *Fix imbalance method*.

```
XGBClassifier
XGBClassifier(base_score=None, booster='gbtree', callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device='cpu', early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=-1,
              num_parallel_tree=None, objective='binary:logistic', ...)
```

Figura A.3. Hiperparámetros del modelo *XGBoost* optimizado.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_samples_leaf=1,
                       min_samples_split=2, min_weight_fraction_leaf=0.0,
                       monotonic_cst=None, n_estimators=100, n_jobs=-1,
                       oob_score=False, random_state=3492, verbose=0,
                       warm_start=False)
```

Figura A.4. Hiperparámetros del modelo *Random Forest* optimizado.

Bibliografía

- ABDO, A. A., Ackermann, M., Ajello, M., Allafort, A., Antolini, E., Atwood, W. B., ... y Pepe, M. The first catalog of active galactic nuclei detected by the Fermi large area telescope. The Astrophysical Journal, 2010, vol. 715, no 1, p. 429.
- ANDRUCHOW, I.; CELLONE, S. A.; ROMERO, G. E. Incidence of the host galaxy on the measuresments of the optical linear polarization of blazars. Boletín de la Asociación Argentina de Astronomía La Plata, Argentina, 2005, vol. 48, p. 434-440.
- Andruchow I., 2006, Tesis de Doctorado, UNLP. Tema: Estudios Fotopolarimétricos de la Microvariabilidad en Blazares. Directores: Dres. Gustavo E. Romero y Sergio A. Cellone.
- Andruchow, I., Cellone, S. A., Romero, G. E., Dominici, T. P., y Abraham, Z. (2003) Microvariability in the optical polarization of 3C 279. Astronomy & Astrophysics, 2003, vol. 409, no 3, p. 857-865.
- Beckmann, V., y Shrader, C. R. The AGN phenomenon: open issues. arXiv preprint arXiv:1302.1397, 2013.
- BLANDFORD, R D.; PAYNE, D. G. Hydromagnetic flows from accretion discs and the production of radio jets. Monthly Notices of the Royal Astronomical Society, 1982, vol. 199, no 4, p. 883-903.
- BREIMAN, L. Random forests. Machine learning, 2001, vol. 45, p. 5-32.
- Carini, M. T., Miller, H. R., Noble, J. C., y Sadun, A. C. The timescales of the optical variability of blazars. II-AP Librae. Astronomical Journal (ISSN 0004-6256), vol. 101, April 1991, p. 1196-1201.
- Carini, M. T., Miller, H. R., Noble, J. C., y Goodrich, B. D. The timescales of the optical variability of blazars. III-OJ 287 and BL Lacertae. Astronomical Journal (ISSN 0004-6256), vol. 104, no. 1, July 1992, p. 15-27.
- CHOLLET, F. Deep Learning With Python Manning. Publications Company. 2017.
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 1997, vol. 55, no 1, p. 119-139.
- Giommi, P., Polenta, G., Lähteenmäki, A., Thompson, D. J., Capalbi, M., Cutini, S., ... y Zhou, X. Simultaneous Planck, Swift, and Fermi observations of X-ray and γ-ray selected blazars. Astronomy & Astrophysics, 2012, vol. 541, p. A160.

- GONZÁLEZ-PÉREZ, J. N.; KIDGER, M. R.; MARTÍN-LUIS, F. Optical and near-infrared calibration of AGN field stars: an all-sky network of faint stars calibrated on the Landolt system. The Astronomical Journal, 2001, vol. 122, no 4, p. 2055.
- HOWELL, S. B.; JACOBY, G. H. Time-resolved photometry using a CCD. Publications of the Astronomical Society of the Pacific, 1986, vol. 98, no 606, p. 802.
- HOWELL, S- B.; MITCHELL, K. J.; WARNOCK III, A. Statistical error analysis in CCD time-resolved photometry with applications to variable stars and quasars. Astronomical Journal (ISSN 0004-6256), vol. 95, Jan. 1988, p. 247-256.
- JIJO, B. T.; ABDULAZEEZ, A. M. Classification Based on Decision Tree Algorithm for Machine Learning. 02 (01), 20–28. 2021.
- KEEL, W. C. Spectroscopic evidence for activity in the nuclei of normal spiral galaxies. Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 269, June 15, 1983, p. 466-486.
- LECUN, Y.; BENGIO, Y.; HINTON, G.. Deep learning. nature, 2015, vol. 521, no 7553, p. 436-444.
- LYNDEN-BELL, D. Galactic nuclei as collapsed old quasars. Nature, 1969, vol. 223, no 5207.
- MILLER, H. R.; CARINI, M. T.; GOODRICH, B. D. Detection of microvariability for BL Lacertae objects. Nature, 1989, vol. 337, no 6208, p. 627-629.
- NIELSEN, D. Tree boosting with xgboost-why does xgboost win every machine learning competition? 2016. Tesis de Maestría. Nowegian University.
- PADOVANI, Paolo; GIOMMI, Paolo. The connection between x-ray-and radio-selected BL Lacertae objects. Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 444, no. 2, p. 567-581, 1995, vol. 444, p. 567-581.
- ROMERO, G. E.; CELLONE, S. A.; COMBI, J. A. Optical microvariability of southern AGNs. Astronomy and Astrophysics Supplement Series, 1999, vol. 135, no 3, p. 477-486.
- ROMERO, Gustavo E., et al. Optical microvariability of EGRET blazars. Astronomy & Astrophysics, 2002, vol. 390, no 2, p. 431-438.
- SALPETER, E. E. Accretion of interstellar matter by massive objects. Publications, 1964, vol. 1, p. 165.
- STALIN, C. S.; SAGAR, Ram; WIITA, Paul J. Intranight optical variability of radio-quiet and radio lobe-dominated quasars. Monthly Notices of the Royal Astronomical Society, 2004, vol. 350, no 1, p. 175-188.
- URRY, C. Megan; PADOVANI, Paolo. Unified schemes for radio-loud active galactic nuclei. Publications of the Astronomical Society of the Pacific, 1995, vol. 107, no 715, p. 803.
- VÉRON-CETTY, M.-P.; VÉRON, Philippe. A catalogue of quasars and active nuclei. Astronomy & Astrophysics, 2006, vol. 455, no 2, p. 773-777.
- ZIBECCHI, L. C., 2013, Tesis de Licenciatura, UNLP, Tema: Estudio sobre los métodos estadísticos usados en el análisis de la variabilidad de Núcleos de Galaxias Activos, Directores: Dra. Ileana Andruchow y Dr. Sergio A. Cellone.

- ZIBECCHI, Lorena, et al. Microvariability in AGNs: study of different statistical methods–II. Light curves from simulated images. Monthly Notices of the Royal Astronomical Society, 2020, vol. 498, no 2, p. 3013-3022.
- ZIBECCHI, L., et al. Optical monitoring in southern blazars. Analysis of variability and spectral colour behaviours. Monthly Notices of the Royal Astronomical Society, 2024, vol. 535, no 4, p. 3262-3282.
- ZWICKY, F. Compact galaxies and compact parts of galaxies. II. Astrophysical Journal, vol. 143, p. 192, 1966, vol. 143, p. 192.