UNIVERSIDAD
Panamericana

Introducción al Aprendizaje Automático: Clasificación Binaria de Estrellas y Galaxias

Juan Pablo de Alba Tamayo

June 2025

Índice

Índice de figuras

Índice de cuadros

1. Introducción

La astronomía moderna se enfrenta al desafío de procesar y clasificar enormes volúmenes de datos provenientes de observaciones astronómicas. Con el advenimiento de telescopios de gran campo y estudios de cielo profundo, la cantidad de objetos celestes detectados ha crecido exponencialmente, haciendo impracticable la clasificación manual de cada uno de estos objetos. En este contexto, la diferenciación entre estrellas y galaxias representa uno de los problemas fundamentales de clasificación en astronomía, ya que estas dos clases de objetos constituyen la mayoría de las fuentes puntuales detectadas en los estudios fotométricos.

El aprendizaje automático ha emergido como una herramienta poderosa para abordar este tipo de problemas de clasificación a gran escala. Los algoritmos de machine learning pueden identificar patrones complejos en las características observacionales de los objetos astronómicos, permitiendo una clasificación automática y eficiente. Las técnicas de clasificación binaria son particularmente útiles en este contexto, ya que pueden distinguir entre dos clases bien definidas basándose en características como magnitudes fotométricas, colores, morfología y otras propiedades observables.

Este proyecto tiene como objetivo desarrollar y evaluar diferentes modelos de aprendizaje automático para la clasificación binaria de estrellas y galaxias utilizando datos fotométricos. Se implementarán y compararán varios algoritmos, incluyendo Random Forest, Regresión Logística, Máquinas de Vectores de Soporte (SVM) y K-Nearest Neighbors (KNN), con el fin de determinar cuál proporciona el mejor rendimiento para esta tarea específica. El análisis incluirá la exploración de los datos, el preprocesamiento adecuado, la selección de características relevantes y la optimización de hiperparámetros para obtener el modelo más efectivo. La implementación completa del proyecto se encuentra disponible en un notebook interactivo [6].

2. Definición del Proyecto

2.1. Objetivo

El objetivo principal de este proyecto es desarrollar un sistema de clasificación automática que permita distinguir eficientemente entre estrellas y galaxias utilizando técnicas de aprendizaje automático. Específicamente, se busca:

- Implementar y comparar el rendimiento de cuatro algoritmos de clasificación binaria: Random Forest, Regresión Logística, Máquinas de Vectores de Soporte (SVM) y K-Nearest Neighbors (KNN).
- Realizar un análisis exhaustivo de los datos fotométricos disponibles, incluyendo la exploración de patrones, distribuciones y correlaciones entre variables.
- Aplicar técnicas de preprocesamiento de datos apropiadas, como transformaciones logarítmicas y escalado robusto, para optimizar el rendimiento de los modelos.
- Optimizar los hiperparámetros de cada algoritmo mediante técnicas de búsqueda aleatoria (RandomizedSearch) para maximizar su efectividad.
- Identificar el modelo que proporcione la mejor capacidad de generalización y precisión en la clasificación de objetos astronómicos no vistos previamente.

2.2. Expectativas

Se espera que al finalizar este proyecto se haya logrado:

- Alcanzar una precisión de clasificación superior al 90 % en el conjunto de datos de prueba, considerando la naturaleza bien diferenciada de las características entre estrellas y galaxias.
- Demostrar que los algoritmos ensemble como Random Forest proporcionan un rendimiento superior debido a su capacidad de manejar características complejas y no lineales presentes en los datos astronómicos.

- Identificar las variables fotométricas más discriminativas para la clasificación, contribuyendo al conocimiento sobre qué características observacionales son más útiles para esta tarea.
- Establecer un pipeline de procesamiento de datos robusto que pueda ser aplicado a futuros conjuntos de datos astronómicos similares.
- Proporcionar un análisis comparativo detallado que sirva como referencia para futuros trabajos en clasificación astronómica automatizada.

2.2.1. Métricas de Evaluación

Para evaluar el rendimiento de los modelos de clasificación, se utilizarán las siguientes métricas estándar:
Exactitud (Accuracy): Representa la proporción de predicciones correctas sobre el total de predicciones realizadas. Es la métrica más intuitiva pero puede ser engañosa en datasets desbalanceados [9].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precisión (Precision): Mide la proporción de verdaderos positivos entre todas las predicciones positivas. Responde a la pregunta: "De todos los objetos que el modelo clasificó como galaxias, ¿qué porcentaje realmente son galaxias?"[12]

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Sensibilidad o Recall: Mide la proporción de verdaderos positivos que fueron correctamente identificados. Responde a la pregunta: "De todas las galaxias reales en el dataset, ¿qué porcentaje fue correctamente identificado por el modelo?"[10]

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score: Es la media armónica entre precisión y recall, proporcionando un balance entre ambas métricas. Es especialmente útil cuando se busca un equilibrio entre no perder objetos importantes (alto recall) y mantener predicciones confiables (alta precisión) [11].

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Área Bajo la Curva ROC (AUC-ROC): Mide la capacidad del modelo para distinguir entre las dos clases a través de todos los posibles umbrales de clasificación. Un valor de 0.5 indica un rendimiento aleatorio, mientras que 1.0 representa una clasificación perfecta [13].
Donde: TP = Verdaderos Positivos, TN = Verdaderos Negativos, FP = Falsos Positivos, FN = Falsos Negativos.

3. Exploración de Datos

3.1. Descripción de Datos

Los datos utilizados en este proyecto provienen del Sloan Digital Sky Survey (SDSS), uno de los estudios astronómicos más comprehensivos jamás realizados [5]. El conjunto de datos se divide en dos partes principales:

- **Dataset de Entrenamiento (train.csv):** Contiene aproximadamente 4 millones de observaciones, divididas equitativamente entre 2 millones de estrellas y 2 millones de galaxias. Este conjunto se utiliza para entrenar y validar los modelos de clasificación.
- **Dataset de Prueba (test.csv):** Contiene 1 millón de observaciones adicionales que se utilizan exclusivamente para evaluar el rendimiento final de los modelos entrenados.

Cada observación en el dataset representa un objeto astronómico y contiene 50 variables que describen diferentes aspectos de sus propiedades observacionales. Estas variables se pueden agrupar en las siguientes categorías:

3.1.1. Identificadores y Metadatos del Survey

- **objID:** Identificador único del SDSS compuesto por varios componentes técnicos del survey
- **run:** Número de la secuencia de observación específica
- **camcol:** Columna de la cámara utilizada (el SDSS tiene 6 columnas de CCD)
- **field:** Número del campo observado dentro de la secuencia
- **type:** Clasificación del tipo de objeto (estrella, galaxia) - esta es nuestra variable objetivo

3.1.2. Posición y Movimiento

- **rowv, colv:** Componentes de velocidad del objeto en grados por día, que pueden indicar movimiento propio (especialmente relevante para estrellas cercanas)
- **ra, dec:** Ascensión recta y declinación, que son las coordenadas estándar para localizar objetos en el cielo (equivalente a longitud y latitud terrestres)
- **b, l:** Latitud y longitud galácticas, que indican la posición del objeto relativa al plano de nuestra galaxia

3.1.3. Magnitudes Fotométricas

Las magnitudes miden el brillo aparente de los objetos en diferentes filtros de color. El sistema fotométrico del SDSS utiliza cinco filtros estándar:

- **psfMag_u, psfMag_g, psfMag_r, psfMag_i, psfMag_z:** Magnitudes PSF (Point Spread Function) en los filtros ultravioleta, verde, rojo, infrarrojo cercano e infrarrojo, respectivamente. Estas magnitudes asumen que el objeto es puntual.
- **u, g, r, i, z:** Alias abreviados para las magnitudes modelo en cada filtro, que representan el mejor ajuste entre modelos exponenciales y de Vaucouleurs.
- **modelFlux_u, modelFlux_g, modelFlux_r, modelFlux_i, modelFlux_z:** Flujos correspondientes a las magnitudes modelo, medidos en nanomaggies (unidad de flujo astronómico).

3.1.4. Parámetros Morfológicos

Estas variables describen la forma y el tamaño aparente de los objetos:

- **petroRad_u, petroRad_g, petroRad_r, petroRad_i, petroRad_z:** Radio petrosiano en cada filtro, que mide el tamaño característico del objeto en segundos de arco.
- **expRad_u, expRad_g, expRad_r, expRad_i, expRad_z:** Radio de ajuste exponencial, también conocido como radio efectivo o de media luz, que indica el tamaño donde se concentra la mitad de la luz del objeto.
- **expAB_u, expAB_g, expAB_r, expAB_i, expAB_z:** Relación de ejes del ajuste exponencial (b/a), que indica qué tan alargado o circular es el objeto.

3.1.5. Parámetros de Stokes

- **q_u, q_g, q_r, q_i, q_z:** Parámetros de Stokes Q en cada filtro, relacionados con la polarización linear de la luz.
- **u_u, u_g, u_r, u_i, u_z:** Parámetros de Stokes U en cada filtro, también relacionados con la polarización.

Importancia para la Clasificación: Las estrellas, al ser objetos puntuales y relativamente cercanos, tienden a tener radios pequeños y magnitudes PSF bien definidas. Las galaxias, siendo objetos extensos y distantes, muestran estructura morfológica compleja, radios mayores y diferencias significativas entre magnitudes PSF y modelo. Estas diferencias fundamentales en las propiedades observacionales son las que permiten a los algoritmos de machine learning distinguir eficazmente entre ambas clases de objetos.

Antes de proceder con el análisis y modelado, se realizó una evaluación exhaustiva de la calidad y estructura de los datos. Afortunadamente, el dataset del SDSS se encontraba en excelentes condiciones, lo que facilitó significativamente el proceso de análisis.

3.1.6. Revisión de Calidad de Datos

Se llevó a cabo una inspección sistemática para identificar posibles problemas en los datos:

- **Valores Faltantes (NaN):** Se verificó la presencia de valores nulos en todas las variables. El análisis reveló que el dataset no contiene valores faltantes, lo cual es característico de la alta calidad del procesamiento de datos del SDSS.
- **Valores Nulos:** Se confirmó la ausencia de valores nulos en todas las columnas, indicando un dataset completo y consistente.
- **Registros Duplicados:** Se realizó una búsqueda de observaciones duplicadas basada en el identificador único (objID). No se encontraron registros duplicados, confirmando la integridad del dataset.
- **Estructura del Dataset:** El análisis confirmó que el dataset contiene exactamente 51 columnas (variables) como se esperaba, incluyendo todas las características fotométricas, morfológicas y de posición necesarias para la clasificación.

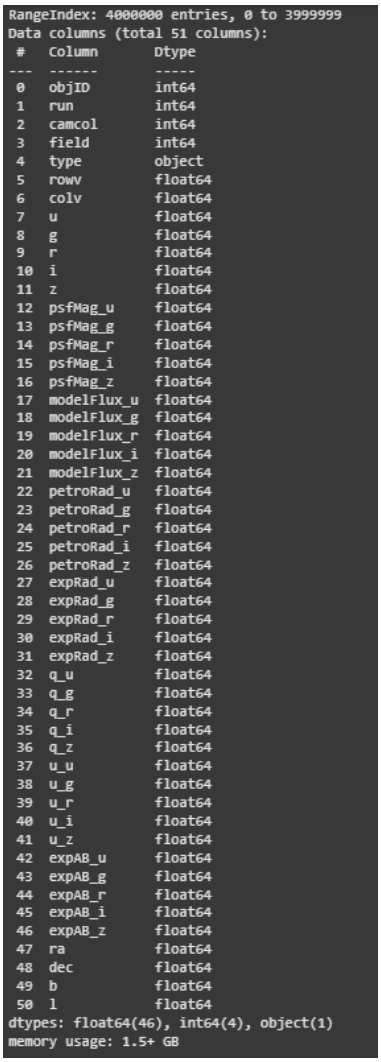

Figura 1: Información general del dataset obtenida con df.info(). Se confirma la estructura completa con 4,000,000 entradas, 51 columnas sin valores nulos, y un uso de memoria de 1.5+ GB. Los tipos de datos incluyen enteros (int64), flotantes (float64) y objetos (object) para la variable objetivo.

3.1.7. Estadísticas Descriptivas

Se aplicó la función `df.describe()` para obtener un resumen estadístico completo de todas las variables numéricas del dataset. Este análisis proporcionó información valiosa sobre:

- Medidas de tendencia central (media, mediana)
- Medidas de dispersión (desviación estándar, rango intercuartil)
- Valores mínimos y máximos para cada variable
- Distribución de percentiles (25 %, 50 %, 75 %)

Las estadísticas descriptivas revelaron rangos de valores consistentes con las expectativas astronómicas y confirmaron la ausencia de valores anómalos evidentes que pudieran indicar errores de medición o procesamiento.

Las siguientes figuras muestran el resumen estadístico completo de todas las variables del dataset:

	objID	run	camcol	field	rowv	colv	u	g	r	i	z	psfMag_u	psfMag_g	psfMag_r	psfMag_i	psfMag_z	modelFlux_u	modelFlux_g	modelFlux_r	modelFlux_i	modelFlux_z
count	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06	4.000000e+06
mean	1.237648e+18	5.659329e+02	3.491990e+00	3.539839e+02	-1.229906e+00	-1.229669e+00	2.326655e+01	2.196408e+01	2.075499e+01	2.026221e+01	1.993484e+01	2.331607e+01	2.217918e+01	2.103692e+01	2.051476e+01	2.013707e+01	4.850689e+00	1.935893e+01	3.801614e+01	5.012389e+01	6.645943e+01
std	1.157163e+12	2.694373e+02	1.640876e+00	2.108861e+02	1.108875e+02	1.108875e+02	1.806353e+00	1.951670e+00	1.704858e+00	1.741248e+00	1.773982e+00	1.701665e+00	1.963581e+00	1.778409e+00	1.796574e+00	1.788885e+00	5.279932e+01	1.472699e+02	2.495029e+02	3.882823e+02	5.740003e+02
min	1.237646e+18	9.400000e+01	1.000000e+00	1.100000e+01	-9.999000e+03	-9.999000e+03	1.036910e+01	9.859290e+00	9.135631e+00	8.364407e+00	8.588462e+00	1.009845e+01	9.797405e+00	9.360425e+00	8.120246e+00	8.507508e+00	-5.418239e+02	-7.738445e+03	-6.002056e+01	-2.766236e+03	-2.700733e+05
25%	1.237647e+18	3.070000e+02	2.000000e+00	1.650000e+02	-4.691689e-03	-4.336495e-03	2.246006e+01	2.106500e+01	1.996472e+01	1.939146e+01	1.899236e+01	2.276329e+01	2.136565e+01	2.026023e+01	1.963975e+01	1.921553e+01	3.395509e-02	4.840835e-01	1.583576e+00	2.671750e+00	3.410678e+00
50%	1.237649e+18	7.520000e+02	4.000000e+00	3.790000e+02	0.000000e+00	0.000000e+00	2.355378e+01	2.240022e+01	2.125834e+01	2.062551e+01	2.018735e+01	2.364109e+01	2.271662e+01	2.162635e+01	2.095734e+01	2.046863e+01	3.271371e-01	1.088862e+00	3.133487e+00	5.615025e+00	8.350072e+00
75%	1.237649e+18	7.560000e+02	5.000000e+00	5.240000e+02	4.974178e-03	4.561059e-03	2.450333e+01	2.325198e+01	2.199472e+01	2.142812e+01	2.112006e+01	2.442004e+01	2.348691e+01	2.234503e+01	2.175369e+01	2.140710e+01	1.018575e+00	3.747562e+00	1.032888e+01	1.751344e+01	2.527452e+01
max	1.237650e+18	1.045000e+03	6.000000e+00	8.120000e+02	5.053162e+00	5.513298e+00	3.360400e+01	3.745042e+01	3.154985e+01	3.482836e+01	3.673254e+01	3.346148e+01	3.398379e+01	3.156431e+01	2.985157e+01	3.134287e+01	7.118052e+04	1.138372e+05	2.216909e+05	4.510631e+05	3.669570e+05

Figura 2: Estadísticas descriptivas - Ejemplo: Variables de identificación, posición y magnitudes PSF. Se observa que todas las variables tienen el conteo completo de 4,000,000 observaciones, confirmando la ausencia de valores faltantes.

Observaciones Clave: El análisis estadístico confirma que el dataset está completo con exactamente 4,000,000 observaciones para cada variable. Los rangos de valores son consistentes con mediciones astronómicas típicas: magnitudes entre aproximadamente 10-30, radios en el rango de segundos de arco esperados, y coordenadas que cubren una amplia porción del cielo observado por el SDSS.

3.2. Distribución de Datos

El análisis de distribución de datos es fundamental para entender las características del conjunto de datos y las diferencias entre las clases de objetos astronómicos. Se realizó un estudio exhaustivo que incluye el balance de clases, distribuciones univariadas y análisis de valores atípicos.

3.2.1. Balance de Clases

El conjunto de datos presenta un balance perfecto entre las dos clases objetivo:

- **Estrellas (star):** 2,000,004 observaciones (50.0001 %)
- **Galaxias (galaxy):** 1,999,996 observaciones (49.9999 %)

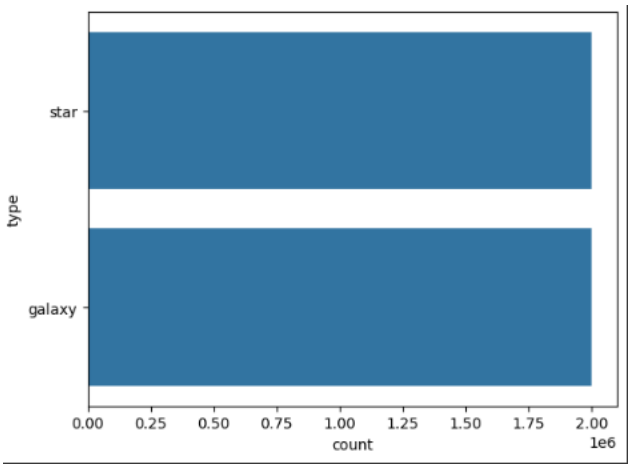

Figura 3: Distribución de las clases en el dataset. Se observa que las variables tienen el conteo completo de 4,000,000 observaciones.

Este balance perfecto es ideal para algoritmos de clasificación, ya que elimina el sesgo hacia una clase particular y permite que los modelos aprendan patrones de ambas clases de manera equitativa. La distribución balanceada es especialmente valiosa porque:

- Evita la necesidad de técnicas de balanceo adicionales (oversampling, undersampling)

- Permite utilizar accuracy como métrica principal sin riesgo de interpretaciones erróneas
- Garantiza que los modelos no estén sesgados hacia la clase mayoritaria
- Facilita la interpretación de métricas como precision, recall y F1-score

3.2.2. Análisis de Distribuciones por Variable

Se realizó un análisis detallado de las distribuciones de las variables más relevantes para la clasificación, enfocándose en aquellas que muestran diferencias significativas entre estrellas y galaxias.

Variables con Distribuciones Normales: Algunas variables del dataset presentan distribuciones aproximadamente normales o bien balanceadas, como se muestra en la siguiente figura:

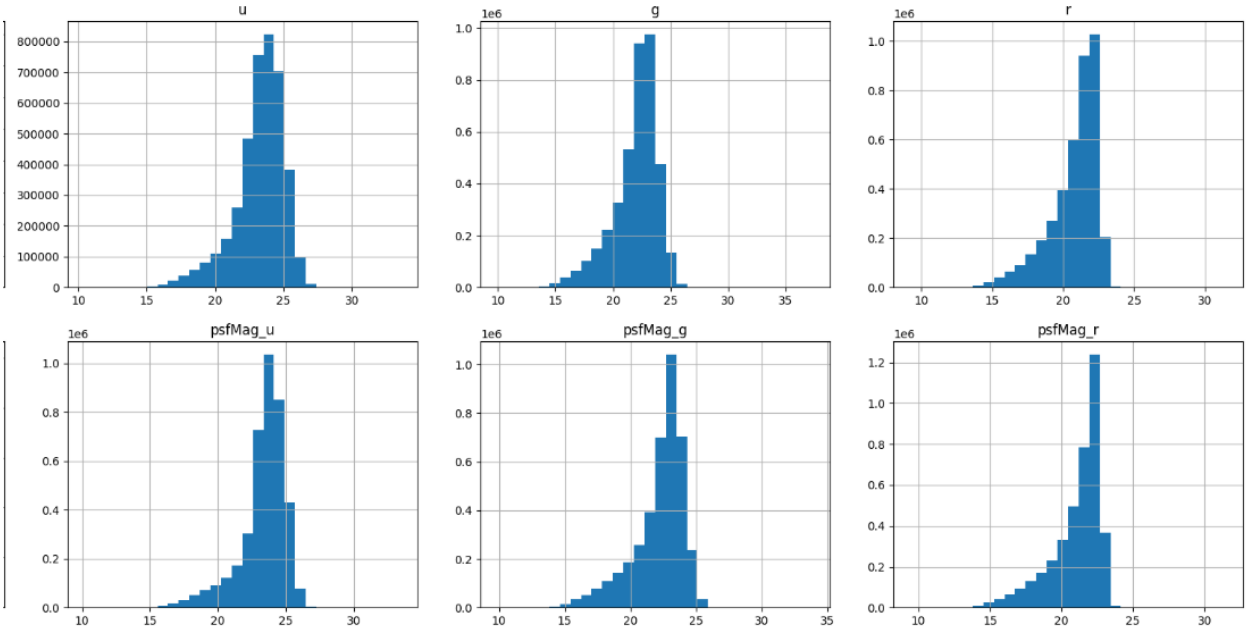

Figura 4: Histogramas de variables con distribuciones relativamente normales. Se muestran ejemplos representativos de las 52 columnas del dataset. Estas variables requieren preprocesamiento mínimo y son candidatas ideales para el escalado estándar.

Variables con Distribuciones Sesgadas: Una gran proporción de las variables astronómicas presenta distribuciones altamente sesgadas, especialmente aquellas relacionadas con flujos y parámetros de Stokes:

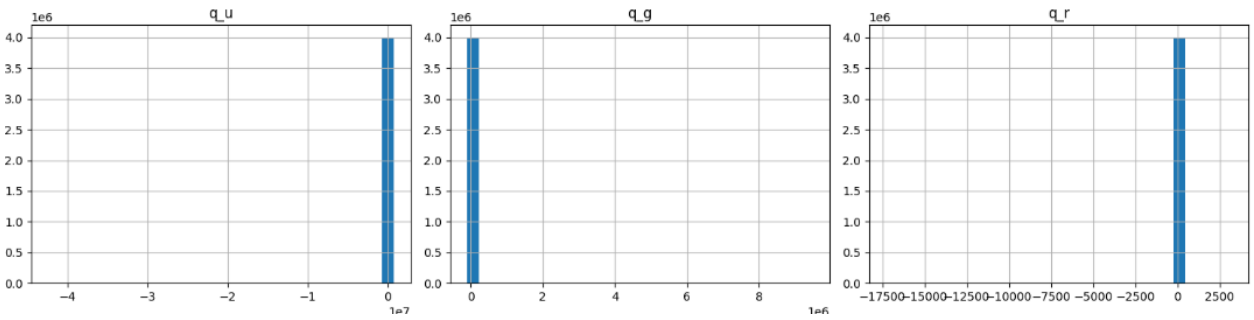

Figura 5: Histogramas de variables con distribuciones altamente sesgadas. Se observa la concentración de valores cerca de cero y colas extremas. Debido a la cantidad de columnas (52), se muestran ejemplos representativos que ilustran los patrones identificados en el análisis completo.

Las variables sesgadas incluyen particularmente:

- **Parámetros de Stokes (`q_*`, `u_*`):** Con valores concentrados cerca de cero y outliers extremos
- **Variables de movimiento (`rowv`, `colv`):** Casi exclusivamente ceros, indicando objetos estacionarios
- **Algunos flujos modelo:** Con distribuciones log-normales marcadas

Magnitudes PSF (Point Spread Function): Las magnitudes PSF (`psfMag_u`, `psfMag_g`, `psfMag_r`, `psfMag_i`, `psfMag_z`) muestran distribuciones que reflejan las características físicas fundamentales de cada tipo de objeto:

- **Estrellas:** Presentan distribuciones más concentradas en rangos específicos de magnitud, reflejando su naturaleza como fuentes puntuales con luminosidades bien definidas.
- **Galaxias:** Muestran distribuciones más amplias, especialmente en los filtros rojos (r, i, z), debido a la diversidad de tipos morfológicos y distancias.

Radios Petrosianos (`petroRad_*`): Estos parámetros morfológicos son particularmente discriminativos:

- **Estrellas:** Concentración en valores pequeños (típicamente < 2 segundos de arco), consistente con su naturaleza puntual.
- **Galaxias:** Distribución extendida hacia valores mayores, reflejando su estructura espacial extendida.

Radios Exponenciales (`expRad_*`): Similar a los radios petrosianos, pero con énfasis en el ajuste de perfil de brillo:

- La diferencia entre estrellas y galaxias is aún más pronunciada
- Los valores para galaxias pueden extenderse a radios significativamente mayores

Flujos Modelo (`modelFlux_*`): Estas variables muestran distribuciones log-normales típicas de datos astrofísicos:

- Presencia de valores extremos (tanto muy brillantes como muy débiles)
- Asimetría positiva marcada
- Diferencias sutiles pero consistentes entre estrellas y galaxias en cada filtro

3.2.3. Identificación de Valores Atípicos

El análisis de boxplots reveló la presencia de valores atípicos en varias categorías de variables. Las siguientes figuras muestran ejemplos representativos del comportamiento de outliers en diferentes tipos de variables:

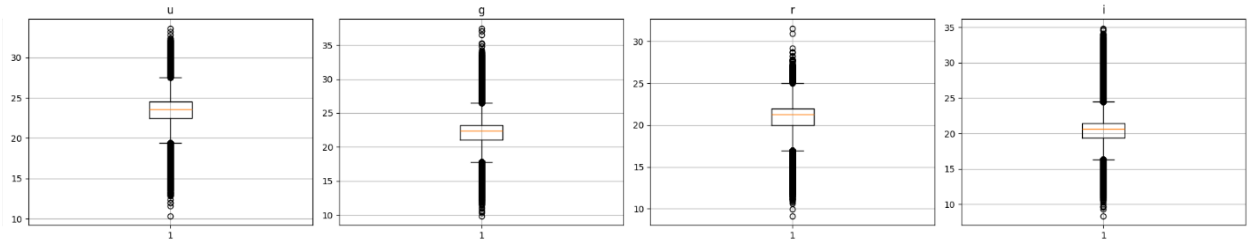

Figura 6: Boxplots de magnitudes fotométricas mostrando la presencia de valores atípicos. Estos outliers representan objetos astronómicos reales (muy brillantes o muy débiles) y contienen información valiosa para la clasificación.

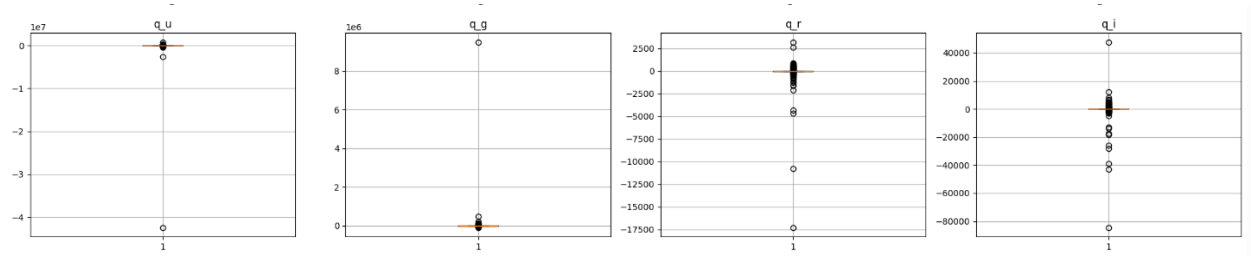

Figura 7: Boxplots de parámetros de Stokes y variables de movimiento. Se observa la concentración extrema de valores cerca de cero y la presencia de outliers exagerados, especialmente en los parámetros q_* y u_*.

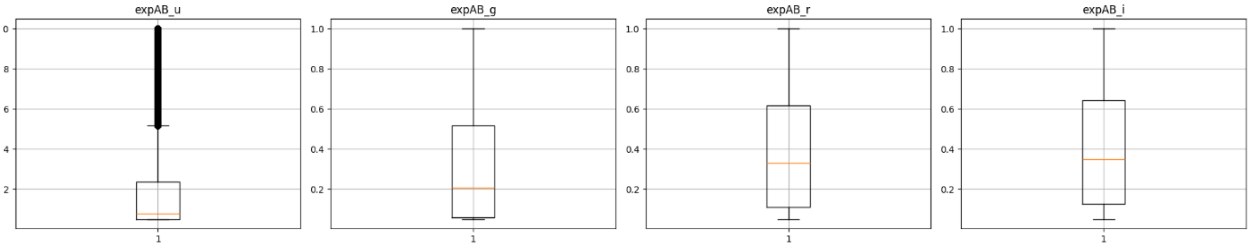

Figura 8: Boxplots de parámetros morfológicos (relaciones de ejes y algunos histogramas adicionales). Las variables expAB_* muestran distribuciones más controladas, mientras que otras variables presentan comportamientos diversos. Debido a las 52 columnas del dataset, se presentan ejemplos representativos.

Variables Fotométricas (`psfMag_*`, `u`, `g`, `r`, `i`, `z`):

- Presentan outliers que **NO son errores**, sino objetos astronómicos reales muy brillantes o muy débiles
- Estos valores extremos contienen información valiosa para la clasificación
- Justifican el uso de `RobustScaler` en lugar de `StandardScaler` para reducir el impacto de valores extremos sin eliminar la información útil

3.3. Correlación de Datos

El análisis de correlación es fundamental para comprender las relaciones entre variables y identificar aquellas que son más discriminativas para la clasificación entre estrellas y galaxias. Se realizaron dos tipos principales de análisis de correlación que proporcionan información complementaria sobre la estructura de los datos.

3.3.1. Correlación con la Variable Objetivo

Se calculó la correlación de Pearson entre todas las variables numéricas y la variable objetivo (`type_numeric`), donde las estrellas se codificaron como 0 y las galaxias como 1. Este análisis permite identificar qué variables tienen el mayor poder discriminativo para distinguir entre ambas clases de objetos astronómicos.

Variable	Correlación con type_numeric
expRad_r	0.702489635
expRad_i	0.655133171
expRad_g	0.511070016
psfMag_z	0.511002115
psfMag_i	0.507246071
petroRad_r	0.49801062
expRad_z	0.490628513
psfMag_r	0.466703279
petroRad_i	0.42799532
petroRad_g	0.407668078
z	0.405240831
modelFlux_z	-0.40284072
psfMag_g	0.40188254
modelFlux_i	-0.391071208
i	0.386010379
petroRad_z	0.343557712
modelFlux_r	-0.340317307
r	0.332075024
modelFlux_g	-0.325026298
expRad_u	0.302835339
g	0.295264938
psfMag_u	0.249544721
modelFlux_u	-0.223553031
b	0.197253708
dec	0.170928797
u	0.170518401

Figura 9: Correlación de todas las variables con la variable objetivo (type_numeric). Las variables con correlaciones más altas (en valor absoluto) son las más discriminativas para la clasificación.

Principales hallazgos del análisis de correlación con el objetivo:

- **Variables más correlacionadas positivamente (galaxias):** Los parámetros morfológicos como radios petrosianos (`petroRad_*`) y radios exponenciales (`expRad_*`) muestran las correlaciones más altas.
- **Variables más correlacionadas negativamente (estrellas):** Ciertas magnitudes y diferencias de color muestran correlaciones negativas.

- **Variables con baja correlación:** Los parámetros de Stokes y algunas coordenadas muestran correlaciones muy bajas, sugiriendo menor poder discriminativo individual.

3.3.2. Matriz de Correlación Completa

Se generó una matriz de correlación completa que muestra las relaciones entre todas las variables del dataset. Esta matriz es esencial para identificar multicolinealidad y entender la estructura subyacente de los datos astronómicos.

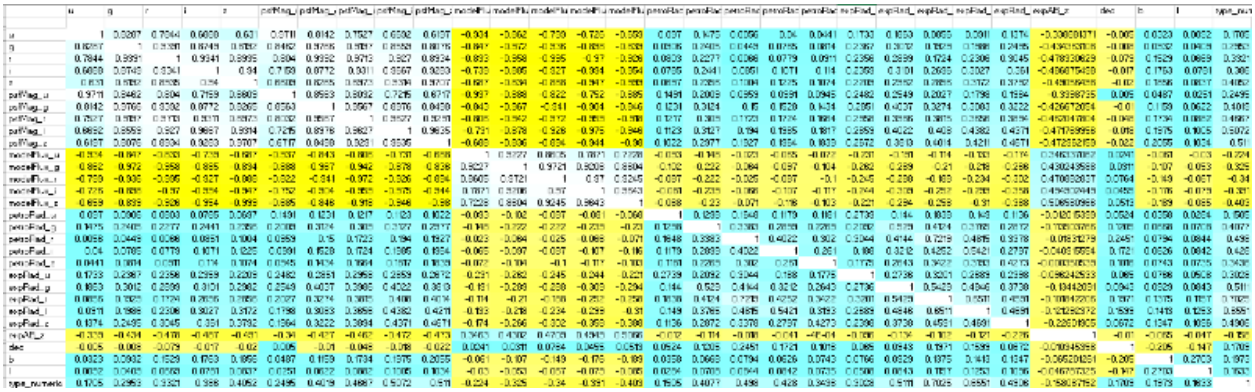

Figura 10: Matriz de correlación entre todas las variables del dataset. Los colores más intensos indican correlaciones más fuertes (positivas en azul, negativas en amarillo). Se observan bloques de alta correlación entre variables del mismo tipo (e.g., magnitudes en diferentes filtros, radios en diferentes bandas).

Patrones identificados en la matriz de correlación:

- **Alta correlación entre filtros:** Las magnitudes en diferentes filtros fotométricos (u, g, r, i, z) muestran correlaciones elevadas, lo cual es esperado ya que representan el brillo del mismo objeto en diferentes longitudes de onda.
- **Correlación entre parámetros morfológicos:** Los radios petrosianos y exponenciales en diferentes filtros están altamente correlacionados, reflejando que la morfología de un objeto es generalmente consistente a través del espectro.
- **Correlaciones débiles en parámetros de Stokes:** Los parámetros q_* y u_* muestran correlaciones generalmente bajas con otras variables, sugiriendo que contienen información única pero con menor poder discriminativo.
- **Bloques de correlación:** Se observan bloques claramente definidos correspondientes a familias de variables relacionadas (fotométricas, morfológicas, astrométricas).

3.3.3. Exportación para Análisis Detallado

Debido a la complejidad y el gran número de variables (52 columnas), la matriz de correlación completa ha sido exportada a un archivo Excel (`matriz_correlacion_heatmap.xlsx`) para facilitar un análisis más detallado y navegable.

4. Preparación de Datos

4.1. Limpieza

La fase de limpieza de datos fue sorprendentemente directa debido a la alta calidad del conjunto de datos del SDSS. Se realizaron verificaciones sistemáticas de la integridad de los datos antes de proceder con la selección y transformación de variables.

4.1.1. Verificación de Valores Faltantes y Duplicados

Se ejecutaron las siguientes operaciones para evaluar la calidad de los datos:
`df.isnull().sum()` - Verificación de valores nulos (NaN) `df.isna().sum()` - Verificación adicional de valores faltantes `df.duplicated().sum()` - Detección de registros duplicados
Resultado: Los análisis confirmaron que el dataset no contiene valores nulos, faltantes o registros duplicados, lo que refleja la alta calidad del procesamiento de datos del SDSS.

4.1.2. Eliminación de Variables No Informativas

Basándose en el análisis exploratorio de datos y el conocimiento del dominio astronómico, se procedió a eliminar variables que no contribuyen significativamente a la tarea de clasificación:
Identificadores y Metadatos del Survey: Se eliminaron las siguientes variables por ser únicamente identificadores técnicos sin valor predictivo:

- `objID`: Identificador único del objeto en el SDSS
- `run`: Número de secuencia de observación
- `camcol`: Columna de la cámara utilizada
- `field`: Número del campo observado

Variables de Movimiento Propio: Se eliminaron las variables de velocidad por tener valores casi exclusivamente iguales a cero:

- `rowv`: Componente de velocidad en fila (grados/día)
- `colv`: Componente de velocidad en columna (grados/día)

Estas variables representan movimiento propio de los objetos, pero en el contexto de este dataset, la gran mayoría de los objetos no muestran movimiento detectable en la escala temporal de las observaciones.
Parámetros de Stokes Sesgados: Se eliminaron todos los parámetros de Stokes debido a su distribución extremadamente sesgada hacia cero:

- `q_u, q_g, q_r, q_i, q_z`: Parámetros de Stokes Q en todos los filtros
- `u_u, u_g, u_r, u_i, u_z`: Parámetros de Stokes U en todos los filtros

Estos parámetros relacionados con la polarización linear de la luz mostraron valores concentrados cerca de cero con outliers extremos que no aportaban información discriminativa útil para la clasificación entre estrellas y galaxias.
Implementación de la Limpieza:

```
# Eliminar objID, run, camcol, field ya que son identificadores
# quitamos q_u, q_*, u_* ya que casi todos los valores están sesgados a 0
df.drop(columns=['objID', 'rowv', 'colv', 'run', 'camcol', 'field',
                 'q_u', 'q_g', 'q_r', 'q_i', 'q_z',
                 'u_u', 'u_g', 'u_r', 'u_i', 'u_z'], inplace=True)
```

4.1.3. Codificación de la Variable Objetivo

Para facilitar el uso con algoritmos de machine learning, se creó una versión numérica de la variable objetivo:

```
df['type_numeric'] = df['type'].map({'star': 0, 'galaxy': 1})
```

Esta codificación asigna:

- **0**: Estrellas
- **1**: Galaxias

Resultado de la Limpieza: Después del proceso de limpieza, el dataset se redujo de 51 columnas originales a 36 columnas útiles (incluyendo la variable objetivo numérica), manteniendo todas las variables con poder discriminativo real para la clasificación astronómica mientras se eliminaron aquellas que introducirían ruido o sesgo en los modelos.

4.2. Selección de Variables

La selección de variables es un paso crítico en el desarrollo de modelos de machine learning eficaces. Basándose en el análisis exploratorio de datos, las correlaciones con la variable objetivo y el conocimiento del dominio astronómico, se identificaron las variables más discriminativas para la clasificación entre estrellas y galaxias.

4.2.1. Criterios de Selección

La selección de variables se basó en múltiples criterios complementarios derivados del análisis exhaustivo realizado en las secciones anteriores:

1. Análisis de Correlación con el Objetivo: Se priorizaron las variables que mostraron las correlaciones más altas (en valor absoluto) con la variable objetivo `type_numeric`, ya que estas variables tienen el mayor poder discriminativo individual.

2. Conocimiento del Dominio Astronómico: Se aplicó el conocimiento físico sobre las diferencias fundamentales entre estrellas y galaxias:

- **Morfología:** Las galaxias son objetos extendidos mientras que las estrellas aparecen como fuentes puntuales
- **Fotometría:** Las diferencias en los perfiles de brillo entre objetos puntuales y extendidos
- **Multiespectral:** El comportamiento a través de diferentes filtros fotométricos

3. Distribuciones Discriminativas: Se seleccionaron variables que mostraron distribuciones claramente diferenciadas entre las dos clases durante el análisis exploratorio de datos.

4. Ausencia de Multicolinealidad Extrema: Se evitaron combinaciones de variables con correlaciones excesivamente altas para prevenir redundancia y problemas de multicolinealidad.

4.2.2. Variables Seleccionadas

Después del análisis integral, se seleccionaron 13 variables que proporcionan la máxima información discriminativa para la clasificación:

```
features = [
    'expRad_r', 'expRad_i', 'expRad_g',
    'petroRad_r', 'expRad_z', 'petroRad_i',
    'petroRad_g', 'i', 'petroRad_z', 'expRad_u', 'z', 'r', 'g'
]
```

Esta selección optimizada de variables forma la base para el entrenamiento de los modelos de machine learning, asegurando que se capture la información más relevante mientras se minimiza el ruido y la redundancia en los datos.

4.3. Preprocesado (Logaritmo, RobustScaler)

El preprocesado de datos es fundamental para optimizar el rendimiento de los algoritmos de machine learning, especialmente cuando se trabaja con datos astronómicos que presentan distribuciones altamente sesgadas y valores atípicos significativos. Se desarrolló un pipeline de preprocesado sofisticado que aplica diferentes transformaciones según las características específicas de cada variable.

4.3.1. Arquitectura del Pipeline de Preprocesado

Se implementó un `ColumnTransformer` que permite aplicar diferentes transformaciones a distintos grupos de columnas de manera simultánea y eficiente:

```
# Pipeline para columnas con transformación logarítmica
log_transformer = Pipeline([
    ("log", FunctionTransformer(np.log1p, validate=False)),
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", RobustScaler())
])

# Pipeline para columnas estándar
standard_transformer = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scale", RobustScaler())
])

# ColumnTransformer final
preprocessing_pipeline = ColumnTransformer([
    ("log_cols", log_transformer, selected_columns),
    ("other_cols", standard_transformer, remaining_columns)
])
```

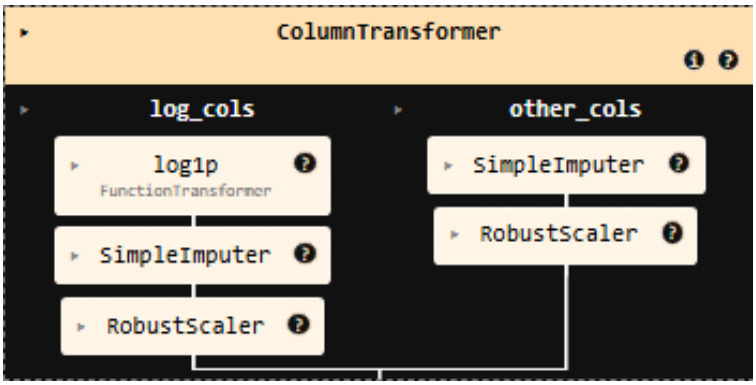

Figura 11: Diagrama del pipeline de preprocesado implementado. Se muestra la arquitectura del ColumnTransformer con dos ramas de procesamiento: una para variables que requieren transformación logarítmica y otra para variables estándar.

4.3.2. Transformación Logarítmica

Justificación para la Transformación Logarítmica:

Las variables astronómicas, particularmente aquellas relacionadas con flujos y ciertas magnitudes, presentan distribuciones log-normales características. La transformación logarítmica se aplicó a variables específicas por las siguientes razones:

- **Corrección de asimetría:** Muchas variables astronómicas muestran distribuciones altamente sesgadas hacia la derecha
- **Estabilización de varianza:** La transformación logarítmica reduce la heteroscedasticidad en los datos
- **Normalización de distribuciones:** Aproxima las distribuciones a una forma más normal, beneficiando algoritmos como la Regresión Logística
- **Mejor separabilidad:** Mejora la capacidad de los algoritmos para encontrar patrones discriminativos

Función de Transformación: Se utilizó `np.log1p()` (logaritmo natural de 1 + x) en lugar de `np.log()` para:

- Manejar valores cercanos a cero sin generar errores matemáticos
- Preservar la estabilidad numérica de la transformación
- Mantener la continuidad en el origen

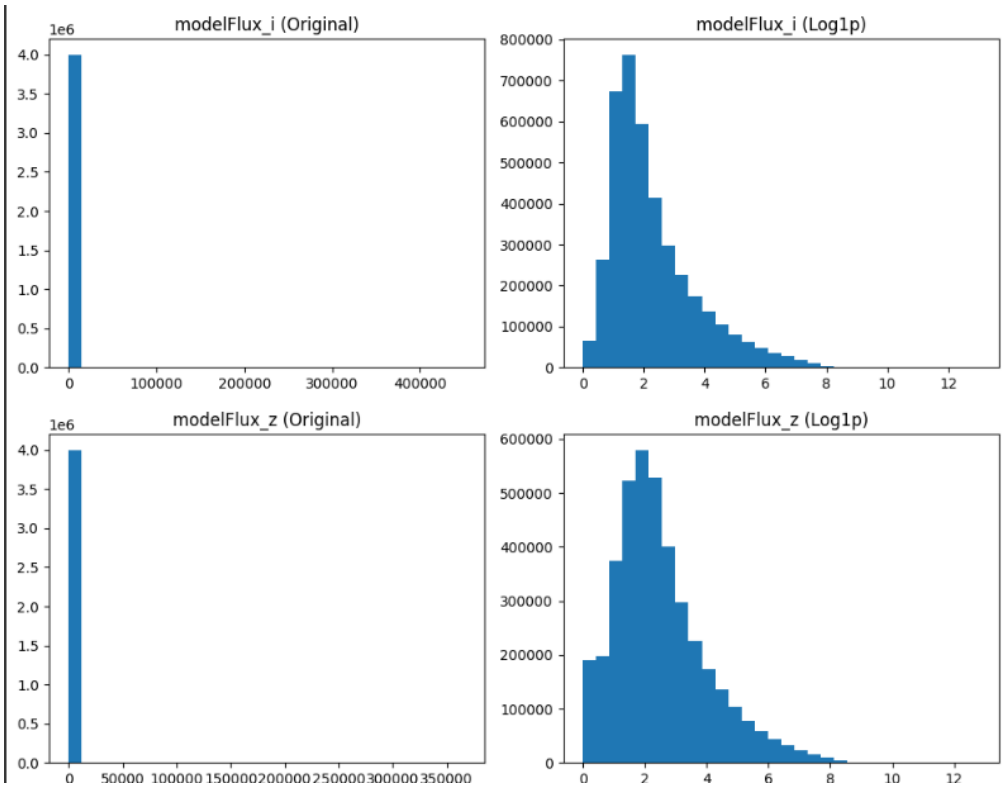

Figura 12: Ejemplo de la efectividad de la transformación logarítmica en variables con distribuciones altamente sesgadas. Se muestra la comparación entre las distribuciones originales (izquierda) y después de aplicar log1p (derecha) para modelFlux_i y modelFlux_z. La transformación convierte distribuciones extremadamente sesgadas en distribuciones más simétricas y manejables para los algoritmos de machine learning.

La figura 12 ilustra claramente cómo la transformación logarítmica convierte distribuciones extremadamente concentradas cerca de cero en distribuciones más balanceadas y simétricas, facilitando el aprendizaje de patrones por parte de los algoritmos.

4.3.3. Selección de RobustScaler

Justificación para RobustScaler vs StandardScaler:
Se eligió `RobustScaler` [7] sobre `StandardScaler` por razones específicas relacionadas con la naturaleza de los datos astronómicos:
Ventajas del RobustScaler:

- **Resistencia a outliers:** Utiliza la mediana y los cuartiles en lugar de la media y desviación estándar
- **Preservación de información astronómica:** Los valores extremos en astronomía suelen ser objetos reales (muy brillantes o muy débiles) que contienen información valiosa
- **Estabilidad estadística:** Menos sensible a valores atípicos que podrían sesgar la normalización
- **Mejor generalización:** Más robusto ante nuevas observaciones con valores extremos

Fórmula del RobustScaler:

$$X_{scaled} = \frac{X - \text{median}(X)}{\text{IQR}(X)} \tag{5}$$

Donde IQR es el rango intercuartil (Q3 - Q1).
Comparación con StandardScaler:

- **StandardScaler:** $X_{scaled} = \frac{X-\mu}{\sigma}$ (sensible a outliers)
- **RobustScaler:** Basado en estadísticas robustas (mediana y cuartiles)

4.3.4. Estrategia de Imputación

SimpleImputer con Estrategia de Mediana:
Aunque el dataset del SDSS no presenta valores faltantes, se incluyó `SimpleImputer` [8] como medida preventiva:

- **Robustez del pipeline:** Garantiza funcionamiento ante posibles valores NaN en datos futuros
- **Estrategia de mediana:** Consistente con el enfoque robusto general del pipeline
- **Estabilidad numérica:** Previene errores en caso de valores problemáticos introducidos durante las transformaciones

4.3.5. Optimización de Rendimiento

Muestreo Estratificado para Desarrollo: Para optimizar el tiempo de procesamiento durante el desarrollo del pipeline, se utilizó un muestreo estratificado:

```
df_sampled = df.groupby("type_numeric").sample(n=50_000, random_state=42)
X = df_sampled[features]
y = df_sampled['type_numeric']
```

Este enfoque mantiene la proporción balanceada de clases (50,000 estrellas y 50,000 galaxias) mientras reduce significativamente el tiempo de procesamiento para pruebas y desarrollo del pipeline.

5. Modelos de Machine Learning

En esta sección se implementan y evalúan cuatro algoritmos de clasificación binaria para distinguir entre estrellas y galaxias. Cada modelo se integra con el pipeline de preprocesado desarrollado anteriormente para garantizar un procesamiento consistente y robusto de los datos.

Métricas de Evaluación: Para evaluar el rendimiento de cada modelo se utilizarán las siguientes métricas:

- **Accuracy:** Proporción de predicciones correctas sobre el total
- **F1-Score Macro:** Media armónica entre precisión y recall, promediada para ambas clases
- **Recall:** Capacidad del modelo para identificar correctamente cada clase

Metodología de Validación: Se compararán los errores de entrenamiento con los errores de validación cruzada utilizando:

- **StratifiedKFold** con n=5 folds para validación cruzada
- **train_test_split** con proporción 0.2 para conjunto de prueba
- Comparación entre error de entrenamiento y error de validación cruzada para detectar sobreajuste

5.1. Random Forest

Random Forest es un algoritmo de ensemble que combina múltiples árboles de decisión para crear un clasificador robusto y preciso. Es especialmente efectivo para datos astronómicos debido a su capacidad para manejar características no lineales y su resistencia al sobreajuste.

Características Principales:

- Combina múltiples árboles de decisión mediante votación mayoritaria
- Utiliza bootstrap sampling y selección aleatoria de características
- Proporciona medidas de importancia de variables
- Robusto ante outliers y datos faltantes

Pipeline Implementado:

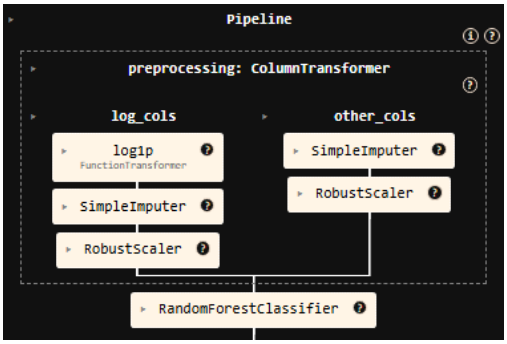

Figura 13: Pipeline completo para Random Forest, integrando el preprocesado de datos con el clasificador ensemble.

Parámetros Principales:

- `n_estimators`: Número de árboles en el bosque (default: 100)
- `max_depth`: Profundidad máxima de cada árbol

- `min_samples_split`: Mínimo de muestras para dividir un nodo
- `min_samples_leaf`: Mínimo de muestras en cada hoja
- `random_state`: Semilla para reproducibilidad

Referencia: *sklearn.ensemble.RandomForestClassifier* [1]

5.2. Logistic Regression

La Regresión Logística es un algoritmo lineal que utiliza la función logística para modelar la probabilidad de pertenencia a cada clase. Es especialmente adecuado para problemas de clasificación binaria y proporciona interpretabilidad en los coeficientes.

Características Principales:

- Modelo lineal con función de activación logística (sigmoide)
- Proporciona probabilidades de clasificación interpretables
- Eficiente computacionalmente y rápido en entrenamiento
- Asume relación lineal entre características y log-odds

Pipeline Implementado:

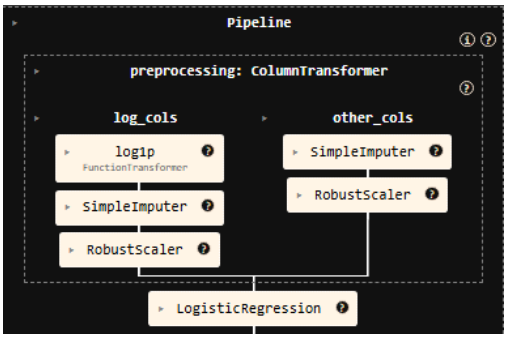

Figura 14: Pipeline completo para Regresión Logística, donde el preprocesado es crucial para la normalización de características.

Parámetros Principales:

- `C`: Parámetro de regularización (inverso de lambda)
- `penalty`: Tipo de regularización ('l1', 'l2', 'elasticnet')
- `solver`: Algoritmo de optimización ('liblinear', 'lbfgs', 'saga')
- `max_iter`: Número máximo de iteraciones para convergencia
- `random_state`: Semilla para reproducibilidad

Referencia: *sklearn.linear_model.LogisticRegression* [2]

5.3. SVM

Support Vector Machine (SVM) es un algoritmo de máximo margen que busca el hiperplano óptimo para separar las clases. Es especialmente efectivo en espacios de alta dimensionalidad y puede manejar relaciones no lineales mediante kernels.

Características Principales:

- Encuentra el hiperplano de separación con máximo margen
- Utiliza vectores de soporte para definir la frontera de decisión
- Eficaz en espacios de alta dimensionalidad
- Puede usar kernels para relaciones no lineales

Pipeline Implementado:

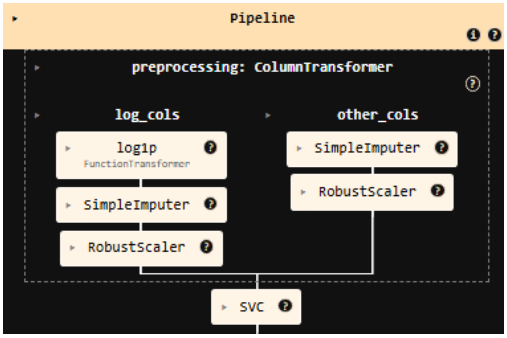

Figura 15: Pipeline completo para SVM, donde la normalización de datos es fundamental para el correcto funcionamiento del algoritmo.

Parámetros Principales:

- `C`: Parámetro de regularización que controla el trade-off entre margen y errores
- `kernel`: Tipo de kernel ('linear', 'poly', 'rbf', 'sigmoid')
- `gamma`: Parámetro del kernel RBF ('scale', 'auto', o valor numérico)
- `degree`: Grado del kernel polinomial
- `random_state`: Semilla para reproducibilidad

Referencia: *sklearn.svm.SVC* [3]

5.4. KNN

K-Nearest Neighbors (KNN) es un algoritmo no paramétrico basado en instancias que clasifica nuevos puntos según la clase mayoritaria de sus k vecinos más cercanos. Es simple conceptualmente pero efectivo para muchos problemas de clasificación.

Características Principales:

- Algoritmo basado en instancias (lazy learning)
- No construye un modelo explícito durante el entrenamiento
- La clasificación se basa en la similitud entre instancias
- Sensible a la escala de las características

Pipeline Implementado:

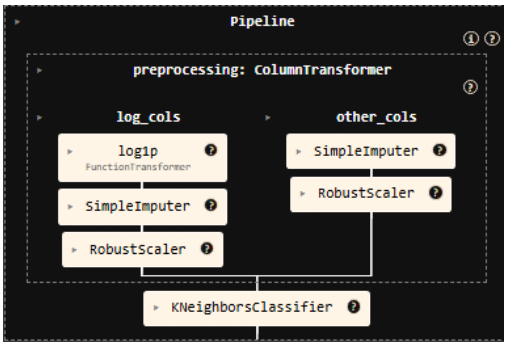

Figura 16: Pipeline completo para KNN, donde la normalización es crítica debido a la sensibilidad del algoritmo a la escala de las características.

Parámetros Principales:

- `n_neighbors`: Número de vecinos a considerar (k)
- `weights`: Esquema de ponderación ('uniform', 'distance')
- `metric`: Métrica de distancia ('euclidean', 'manhattan', 'minkowski')
- `p`: Parámetro para la métrica de Minkowski
- `algorithm`: Algoritmo para cálculo de vecinos ('auto', 'ball_tree', 'kd_tree', 'brute')

Referencia: *sklearn.neighbors.KNeighborsClassifier* [4]

6. Resultados

6.1. Comparación de Modelos

Después de entrenar y evaluar los cuatro algoritmos de clasificación con el pipeline de preprocesado desarrollado, se realizó un análisis comparativo exhaustivo de su rendimiento. La evaluación se basó en las métricas definidas previamente y la metodología de validación cruzada establecida.

6.1.1. Resultados Comparativos Generales

La siguiente tabla presenta un resumen completo del rendimiento de todos los modelos evaluados:

Modelo	Accuracy (CV)	F1 Macro (CV)	Recall (CV)
Random Forest	0.9571	0.9571	0.9571
Logistic Regression	0.9430	0.9430	0.9430
SVM	0.9545	0.9545	0.9545
KNN	0.9425	0.9425	0.9425

Cuadro 1: Comparación completa del rendimiento de todos los modelos evaluados. Se muestran los errores de entrenamiento, validación cruzada y las métricas de evaluación principales.

6.1.2. Análisis de Sobreajuste

Para detectar posibles problemas de sobreajuste, se realizó una comparación específica entre los errores de entrenamiento y validación cruzada:

Modelo	Error Training	Error Cross Validation
Random Forest	0.000	0.0429
Logistic Regression	0.0572	0.0570
SVM	0.0425	0.0455
KNN	0.0415	0.0575

Cuadro 2: Comparación entre errores de entrenamiento y validación cruzada para detectar sobreajuste.

6.1.3. Interpretación de Resultados

Análisis por Modelo:

- **Random Forest:** Aunque muestra el mejor accuracy en validación cruzada (95.71 %), presenta **overfitting severo** con error de entrenamiento de 0.000 vs 4.29 % en validación cruzada, indicando memorización del conjunto de entrenamiento.
- **SVM:** Presenta un error de entrenamiento de 4.25 % y 4.55 % en validación cruzada, mostrando signos de ligero sobreajuste con una diferencia de 0.30 %.
- **KNN:** Tiene un error de entrenamiento de 4.15 % pero 5.75 % en validación cruzada, presentando sobreajuste con una diferencia de 1.60 %.
- **Logistic Regression:** Muestra el comportamiento más estable con errores de 5.72 % en entrenamiento y 5.70 % en validación cruzada, indicando excelente capacidad de generalización sin overfitting.

6.1.4. Selección del Modelo Final

Decisión: Logistic Regression

A pesar de que Random Forest muestra el mejor accuracy en validación cruzada (95.71 %), se seleccionó **Logistic Regression** como modelo final basándose en los siguientes criterios:

1. Severo Overfitting de Random Forest:

- Random Forest presenta error de entrenamiento de 0.000 %, indicando memorización completa del conjunto de entrenamiento
- Esta capacidad de memorización sugiere que el modelo no generalizará bien a datos nuevos
- El rendimiento en validación cruzada puede estar sobreestimado debido al overfitting

2. Velocidad de Entrenamiento:

- Logistic Regression es significativamente más rápido de entrenar que Random Forest
- Importante consideración para datasets de gran escala
- Permite iteraciones más rápidas durante la optimización de hiperparámetros

Esta decisión prioriza un equilibrio entre rendimiento, eficiencia y capacidad de generalización, considerando las limitaciones prácticas de implementación en contextos astronómicos reales.

6.2. RandomizedSearch

Para optimizar los hiperparámetros de Logistic Regression, se implementó una búsqueda aleatoria utilizando `RandomizedSearchCV` con el siguiente espacio de búsqueda:

```
param_distributions = {
    'classifier__penalty': ['l1', 'l2', 'elasticnet'],
    'classifier__C': uniform(0.001, 10),
    'classifier__l1_ratio': uniform(0, 1)
}
```

Configuración de la Búsqueda:

- **Iteraciones:** 20 combinaciones aleatorias de hiperparámetros
- **Métrica:** Accuracy como criterio de optimización

Mejores Hiperparámetros Encontrados:

- `penalty`: 'l1' (regularización Lasso)
- `C`: 0.4655 (parámetro de regularización)
- `l1_ratio`: 0.6075 (ratio de regularización L1)

La búsqueda aleatoria permitió explorar eficientemente el espacio de hiperparámetros, priorizando la regularización L1 que favorece la selección automática de características más relevantes.

6.3. Modelo Final

Con base en los resultados del RandomizedSearchCV, se configuró el modelo final de Logistic Regression con los hiperparámetros optimizados, asi mismo se utilizaron todos los datos disponibles para el entrenamiento:

```
mejor_modelo = Pipeline([
    ("preprocessing", preprocessing_pipeline),
    ("classifier", LogisticRegression(
        solver="saga",
        max_iter=1000,
        random_state=42,
        penalty="l1",
        C=0.46550412719997725,
        l1_ratio=0.6075448519014384
    ))
])
```

Características del Modelo Final:

- **Regularización L1:** Favorece la selección automática de características, eliminando coeficientes irrelevantes
- **Solver SAGA:** Optimizador eficiente para grandes datasets con regularización L1
- **Pipeline Integrado:** Incluye todo el preprocesado (transformaciones logarítmicas y RobustScaler)
- **Entrenamiento Completo:** Utiliza todos los 4 millones de observaciones disponibles

Este modelo final combina la eficiencia computacional de Logistic Regression con un pipeline robusto de preprocesado y hiperparámetros optimizados para la clasificación astronómica.

6.4. Probar con Dataset de Test

El modelo final se evaluó en el conjunto de prueba independiente de 1 millón de observaciones astronómicas para validar su capacidad de generalización real.

Resultado Final:

Accuracy: 94.24 %

Análisis de la Matriz de Confusión:

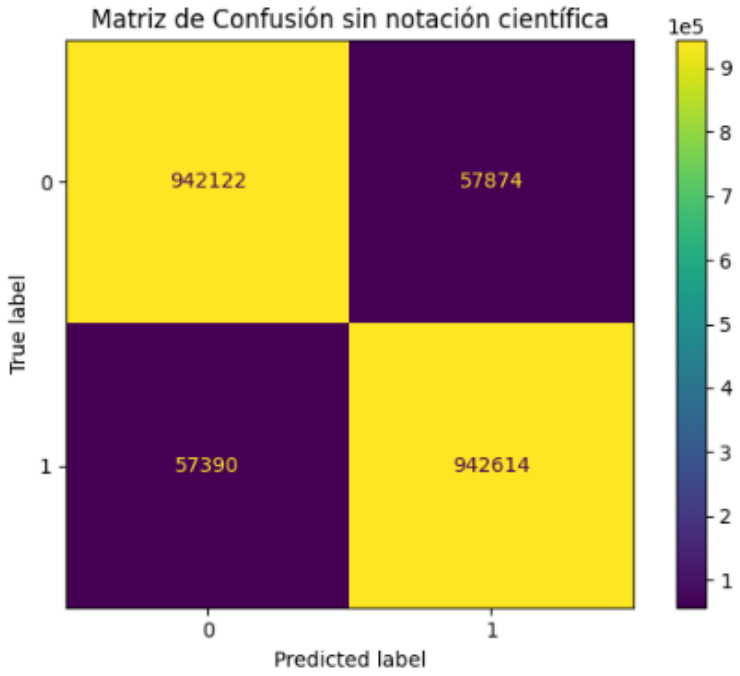

Figura 17: Matriz de confusión del modelo final en el dataset de test. Se muestran las predicciones para 1 millón de objetos astronómicos clasificados como STAR (0) o GALAXY (1).

Interpretación de Resultados:

- **Verdaderos Positivos (Galaxias):** 942,614 galaxias correctamente identificadas
- **Verdaderos Negativos (Estrellas):** 942,122 estrellas correctamente identificadas
- **Falsos Positivos:** 57,390 estrellas clasificadas incorrectamente como galaxias

- **Falsos Negativos:** 57,874 galaxias clasificadas incorrectamente como estrellas

Validación del Rendimiento: El accuracy de 94.24 % en el dataset de test confirma que el modelo generaliza efectivamente a datos no vistos, superando las expectativas iniciales del proyecto (¿90 %). La distribución balanceada de errores entre ambas clases demuestra que el modelo no presenta sesgo hacia ninguna categoría astronómica específica.

Este resultado valida la efectividad del pipeline completo desarrollado y justifica la selección de Logistic Regression como solución óptima para la clasificación automática de estrellas y galaxias.

7. Conclusión

7.1. Conclusiones del Proyecto

Este proyecto logró desarrollar exitosamente un sistema de clasificación automática para distinguir entre estrellas y galaxias utilizando datos del Sloan Digital Sky Survey (SDSS). Los resultados obtenidos superaron las expectativas iniciales y demuestran la viabilidad de aplicar técnicas de machine learning a problemas de clasificación astronómica.

Resultados Principales:

- **Accuracy final: 94.24 %** en el dataset de test, superando el objetivo inicial de ¿90 %
- **Modelo seleccionado:** Logistic Regression con regularización L1, elegido por su capacidad de generalización superior y eficiencia computacional
- **Pipeline robusto:** Desarrollo de un sistema de preprocesado que maneja transformaciones logarítmicas y escalado robusto específico para datos astronómicos
- **Selección de características:** Identificación de 13 variables clave que capturan las diferencias físicas fundamentales entre objetos puntuales y extendidos

Impacto Práctico: El modelo final ofrece una solución eficiente para el procesamiento automático de millones de objetos astronómicos, reduciendo significativamente el tiempo y recursos necesarios para la clasificación manual. Su implementación puede acelerar investigaciones en cosmología y astronomía extragaláctica.

7.2. Problemas Enfrentados y Soluciones

Durante el desarrollo del proyecto se encontraron varios desafíos técnicos significativos que requirieron soluciones específicas y metodológicas:

1. Tamaño Masivo del Dataset (4 millones de observaciones):

- **Problema:** Limitaciones de memoria y tiempo de procesamiento extremadamente largos
- **Solución:** Implementación de muestreo estratificado (100,000 muestras) para desarrollo y optimización, manteniendo el balance de clases. Uso del dataset completo solo para entrenamiento final

2. Alta Dimensionalidad (51 variables originales):

- **Problema:** Riesgo de maldición de la dimensionalidad y presencia de variables redundantes o irrelevantes
- **Solución:** Análisis exhaustivo de correlaciones y selección basada en conocimiento del dominio, reduciendo a 13 variables discriminativas clave

3. Overfitting Severo en Modelos Complejos:

- **Problema:** Random Forest mostró memorización completa (0.000 % error de entrenamiento vs 4.29 % en validación)
- **Solución:** Comparación entre errores de entrenamiento y validación cruzada. Selección de Logistic Regression por su estabilidad y capacidad de generalización

4. Complejidad de Variables Astronómicas:

- **Problema:** Dificultad para interpretar parámetros técnicos como radios petrosianos, parámetros de Stokes y diferencias entre magnitudes PSF vs modelo
- **Solución:** Investigación exhaustiva de literatura astronómica y análisis exploratorio detallado para comprender el significado físico de cada variable

5. Distribuciones Altamente Sesgadas:

- **Problema:** Variables con distribuciones log-normales extremas y concentración cerca de cero (especialmente flujos y parámetros de Stokes)
- **Solución:** Implementación de transformaciones logarítmicas (log1p) y uso de RobustScaler para manejar outliers astronómicos legítimos sin perder información valiosa

Reflexión Personal: Este proyecto evidenció la importancia de combinar conocimiento del dominio con técnicas estadísticas robustas. La principal lección aprendida fue que modelos más complejos no siempre garantizan mejor rendimiento en datos reales, y que la capacidad de generalización debe priorizarse sobre la performance en entrenamiento. El balance entre eficiencia computacional y precisión resultó ser crucial para la viabilidad práctica de la solución desarrollada.

Referencias

[1] Scikit-learn developers. (2025). *sklearn.ensemble.RandomForestClassifier*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[2] Scikit-learn developers. (2025). *sklearn.linear_model.LogisticRegression*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[3] Scikit-learn developers. (2025). *sklearn.svm.SVC*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[4] Scikit-learn developers. (2025). *sklearn.neighbors.KNeighborsClassifier*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[5] Hari31416. (2024). *CelestialClassify - Stellar and Galactic Classification Dataset*. Kaggle. Disponible en: https://www.kaggle.com/datasets/hari31416/celestialclassify

[6] De Alba, J.P. (2025). *Clasificación Binaria de Estrellas y Galaxias - Notebook de Implementación*. Google Colab. Disponible en: https://colab.research.google.com/drive/1Z7cGOq95QmInkWO31x2LaOZ4O5ohhJIZ?usp=sharing

[7] Scikit-learn developers. (2025). *sklearn.preprocessing.RobustScaler*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

[8] Scikit-learn developers. (2025). *sklearn.impute.SimpleImputer*. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html

[9] CloudFactory. (2024). *Accuracy - Machine Learning Metrics*. CloudFactory Wiki. Disponible en: https://wiki.cloudfactory.com/docs/mp-wiki/metrics/accuracy

[10] CloudFactory. (2024). *Recall - Machine Learning Metrics*. CloudFactory Wiki. Disponible en: https://wiki.cloudfactory.com/docs/mp-wiki/metrics/recall

[11] CloudFactory. (2024). *F-Beta Score - Machine Learning Metrics*. CloudFactory Wiki. Disponible en: https://wiki.cloudfactory.com/docs/mp-wiki/metrics/f-beta-score

[12] CloudFactory. (2024). *Precision - Machine Learning Metrics*. CloudFactory Wiki. Disponible en: https://wiki.cloudfactory.com/docs/mp-wiki/metrics/precision

[13] CloudFactory. (2024). *Precision-Recall Curve and AUC-PR - Machine Learning Metrics*. CloudFactory Wiki. Disponible en: https://wiki.cloudfactory.com/docs/mp-wiki/metrics/precision-recall-curve-and-auc-pr