Galaxy Zoo: Modelo para clasificación de galaxias

Daniela María Hernández Céspedes

Monografía presentada para optar al titulo de Especialista en Analítica y Ciencia de datos

Asesor
Doctor. Juan Carlos Muñoz Cuartas

Universidad de Antioquia

Facultad de Ingeniería
Especialización en analítica y ciencia de datos

Medellín Colombia

2023

Cita	Zhu, X. P., Dai, J. M., Bian, C. J., Chen, Y., Chen, S., & Hu, C. (2019).
Referencia **Estilo APA 7 (2020)**	Hernández Céspedes. (2023) . Modelo de clasificación de galaxias, implementando modelos de machine learning. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte III

Tabla de contenido

Lista de tablas

Lista de figuras

Resumen

El propósito de este proyecto es desarrollar modelos que permitan clasificar galaxias según su morfología en tres tipos: elípticas, espirales y sin definir. El primer modelo es de regresión logística y se basará en un data set que contiene una variedad de características, como las líneas de emisión, las masas estelares, la dispersión de velocidad y la clasificación de la morfología de las galaxias, datos obtenidos a través de un ejercicio de inspección visual. El segundo y tercer modelo consiste en redes convolucionales y permitirán la clasificación de galaxias directamente a través de la inspección automática de imágenes. Para esta parte del proceso se usó un conjunto de 141.553 imágenes preetiquetadas que se usaron para entrenar los modelos de clasificación de imágenes.

Toda esta data es de acceso público y se encuentran en los repositorios del proyecto de Galaxy Zoo.

Introducción.

La clasificación de galaxias según su morfología es un campo de investigación fundamental en la astronomía. Especialmente si se tiene en cuenta que en la actualidad y en el futuro cercano los censos de objetos celestes proveen millones de imágenes de objetos que claramente ya no pueden ser clasificados y posteriormente analizados uno a uno.

En este contexto, el presente proyecto tiene como objetivo desarrollar un modelo que permita la clasificación automática de galaxias en tres tipos principales: elípticas, espirales y sin definir. Para lograr esto, se empleará una variedad de características como las líneas de emisión, las masas estelares, la dispersión de velocidad y la clasificación de la morfología de las galaxias evaluadas por humanos. Esta tarea es particularmente importante ya que la clasificación adecuada de las galaxias puede proporcionar información valiosa sobre la formación y evolución de las mismas.

Para el desarrollo de esta monografía se utilizó modelos de Machine Learning de clasificación. En particular, se empleó técnicas de regresión logística y clasificación de imágenes para identificar patrones y características en el data set y en las imágenes de las galaxias que permitan diferenciarlas y clasificarlas según su morfología. Los modelos de regresión logística y clasificación de imágenes han demostrado ser muy efectivos en la identificación de patrones en grandes conjuntos de datos, por lo que se espera que sean herramientas muy útiles en la tarea de clasificación de galaxias.

Esto podría ser útil para entender mejor cómo se relacionan las características de las galaxias con su forma y estructura, lo que podría ayudar a los astrónomos a comprender mejor la formación y evolución de las galaxias.

Objetivos

- Utilizando la regresión logística multiclase, se busca entrenar un modelo para identificar cuáles son las características más importantes para la clasificación de galaxias en las tres categorías: elíptica, espiral e irregular.
- Implementar modelos de redes convolucionales para clasificar las imágenes de galaxias según su forma en elíptica, espiral o irregular. El propósito es lograr una clasificación precisa y automática de las imágenes galácticas basada en su morfología.

Capitulo 1

Este capítulo es una breve introducción a los diferentes modelos que se implementaran para el proceso de clasificación de galaxias.

1.1 Marco Teórico

Para clasificar el tipo de galaxia a partir de un conjunto de datos, se pueden utilizar diversas técnicas de aprendizaje automático, tales como:

1.1.1 Regresión logística: La regresión logística es un modelo estadístico utilizado para analizar relaciones entre una variable dependiente binaria (en este caso, el tipo de galaxia) y una o más variables independientes (las características de las galaxias). Este modelo puede utilizarse para clasificar el tipo de galaxia en función de las características que se tengan.

En la regresión logística, la variable dependiente es categórica, es decir, puede tomar uno de varios valores discretos, como A o B en una regresión binaria, o A, B, C o D en una regresión multinomial. Las variables independientes, por otro lado, pueden ser continuas o categóricas.

El objetivo de la regresión logística es encontrar una relación entre la variable dependiente y las variables independientes, estimando las probabilidades de la variable dependiente en función de las variables independientes. *(¿Qué es la regresión logística? | IBM, s. f.)*

1.1.2 Redes neuronales convolucionales (CNN): Las redes neuronales son un tipo de modelo de aprendizaje automático que simulan el comportamiento de las neuronas en el cerebro humano. En el caso de la clasificación de galaxias, se pueden entrenar redes neuronales convolucionales para identificar patrones y características que permitan diferenciar los diferentes tipos de galaxias en banco de imágenes.

Es importante mencionar que el éxito de la clasificación dependerá en gran medida de la calidad y cantidad de los datos utilizados para entrenar los modelos de aprendizaje automático. Por lo tanto, es fundamental contar con un conjunto de datos amplio y representativo de los diferentes tipos de galaxias que se quieren clasificar.

Las CNN son muy efectivas en la clasificación de imágenes y se han utilizado con éxito en aplicaciones como el reconocimiento de objetos, el reconocimiento facial y la detección de anomalías médicas en imágenes médicas. *(Li, F., & Karpathy, A. (s.f.))*

1.1.3 ResNEt50 V2: ResNet (Red Residual) es una arquitectura de redes neuronales convolucionales (CNN) desarrollada por Microsoft Research en 2015. Ha tenido un gran impacto en el campo del aprendizaje profundo y ha demostrado resultados destacados en diversas tareas de visión por computadora, como clasificación de imágenes, detección de objetos y segmentación semántica.

ResNet V2 es una versión mejorada de ResNet que presenta cambios significativos respecto a su versión anterior. Una de las mejoras más importantes es la eliminación de capas de convolución, lo cual ha logrado una mejora considerable en el rendimiento del modelo. Este cambio en la arquitectura ha permitido una mayor eficiencia y capacidad de aprendizaje en las redes ResNet V2. (*He, K., Zhang, X., Ren, S., & Sun, J. (2015))*

1.1.4 VGG19: VGG19 es una arquitectura de red neuronal convolucional (CNN) que fue presentada por el grupo de investigación Visual Geometry Group (VGG) en la Universidad de Oxford en el 2014.

Esta red consta de 19 capas que incluye capas convolucionales y de agrupación (pooling).
VGG19 logró un rendimiento sobresaliente en la tarea de clasificación de imágenes. La arquitectura se caracteriza por su capacidad para aprender representaciones ricas y profundas de las imágenes, aunque su principal desventaja es su alto costo computacional debido al gran número de parámetros.

A pesar de que han surgido arquitecturas más modernas, como ResNet y EfficientNet, VGG19 sigue siendo una referencia importante en el campo de las CNN y ha servido como base para muchas investigaciones y desarrollos posteriores. *(van Dongen-Boomsma, M., Lansbergen, M. M., & Bekker, E. M. (2019))*

1.1.5 Método LOF (Local Outlier Factor): El método LOF (Local Outlier Factor) es un algoritmo de detección de valores atípicos (outliers) en un conjunto de datos. LOF se basa en el concepto de densidad local para detectar valores atípicos en un conjunto de datos.

En términos simples, el método LOF calcula la densidad local de cada punto en el conjunto de datos en relación con sus vecinos cercanos. Luego, se compara la densidad local de cada punto con la densidad local de sus vecinos para determinar si es un valor atípico.

La idea clave detrás del método LOF es que los valores atípicos tienen una densidad local más baja que sus vecinos. Por lo tanto, los puntos que tienen una densidad local significativamente más baja que la de sus vecinos son etiquetados como valores atípicos.

Para aplicar el método LOF, primero se define el número de vecinos cercanos que se usarán para calcular la densidad local de cada punto. Luego, se calcula la densidad local para cada punto en función de la distancia de sus vecinos cercanos. Finalmente, se calcula el factor de valor atípico LOF para cada punto en función de su densidad local en relación con la densidad local de sus vecinos.

El método LOF se ha utilizado en diversas aplicaciones, como la detección de fraudes financieros, la detección de intrusiones en redes informáticas, la detección de anomalías en datos médicos, entre otras. *(Wikipedia contributors, 2023)*

1.1.6 Método PCA: El Análisis de Componentes Principales (PCA, por sus siglas en inglés) es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos. Su objetivo principal es identificar las componentes principales que explican la mayor varianza en los datos.

El PCA transforma un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas llamadas componentes principales. Estas componentes principales están ordenadas en función de su capacidad para explicar la varianza total de los datos.

El PCA es ampliamente utilizado en diversas áreas, como la reducción de dimensiones, visualización de datos, reconocimiento de patrones y compresión de datos. Permite simplificar y visualizar grandes conjuntos de datos al capturar la mayor cantidad de información posible con menos variables. *(runebook.dev. (s.f.))*

Capítulo 2

En este capítulo, se lleva a cabo la implementación del modelo de regresión logística utilizando el conjunto de datos de características de galaxias.

2.1.1 Implementación del modelo de regresión logística usando Python.

El proyecto Galaxy Zoo tiene una plataforma en línea que utiliza la ayuda de voluntarios para clasificar galaxias en diferentes categorías. Esto ha dado lugar a una gran cantidad de datos que se han utilizado en diversos estudios de investigación.

2.1.2 Descripción de data set.

Para la implementación del modelo de regresión logística, se usó las siguientes tablas que se encuentran publicadas en el proyecto Galaxy Zoo. *(Galaxy Zoo, s. f.)*

Tabla2: Esta tabla trae las clasificaciones de galaxias. El tamaño de la tabla 2 es de (667.944 filas ,16 columnas).

Tabla 4: Esta tabla proporciona una serie de medidas de la confianza de clasificación. El tamaño de la tabla es de (667.944 filas, 10 columnas).

Tabla 5: Esta tabla da los resultados del estudio de sesgo que introdujo imágenes reflejadas. El tamaño de esta tabla es de (91.303 filas, 19 columnas).

Tabla 6: Esta tabla proporciona los resultados del estudio de sesgo que introdujo imágenes monocromáticas. El Tamaño de esta tabla es de (91.303 filas, 11 columnas).

Tabla 7: Esta tabla proporciona la fracción de votos en cada una de las seis categorías clasificadas por humanos, combinando los resultados de los estudios principales y de sesgo. El tamaño de de la tabla es de (893.212 filas, 15 columnas).

En la Tabla 1 se tiene un listado de cada una de las columnas que componen a las 5 tablas, las columnas OBJECTID, RA y DEC se encuentran en todas las tablas. Para el desarrollo de este trabajo se usó el OBJECTID como la key para unir las tablas y obtener una master (tabla única), con la mayor cantidad de características posibles.

Table2	Table4	Table5	Table6	Table7
'OBJID',	‘OBJID',	‘OBJID',	'OBJID',	'OBJID',
'RA',	'RA',	'RA',	'RA',	'RA',
'DEC',	'DEC',	'DEC',	'DEC',	'DEC',
'NVOTE',	'F_UNCLASS_CLEAN',	'NVOTE_MR1',	'NVOTE_MON',	'NVOTE_TOT',
'P_EL',	'F_MISCLASS_CLEAN',	'P_EL_MR1',	'P_EL_MON',	'NVOTE_STD',
'P_CW',	'AVCORR_CLEAN',	'P_CW_MR1',	'P_CW_MON',	'NVOTE_MR1',
'P_ACW',	'STDCORR_CLEAN',	'P_ACW_MR1',	'P_ACW_MON',	'NVOTE_MR2',
'P_EDGE',	'F_MISCLASS_GREATER',	'P_EDGE_MR1',	'P_EDGE_MON',	'NVOTE_MON',
'P_DK',	'AVCORR_GREATER',	'P_DK_MR1',	'P_DK_MON',	'P_EL',
'P_MG',	'STDCORR_GREATER'	'P_MG_MR1',	'P_MG_MON',	'P_CW',
'P_CS',		'P_CS_MR1',	'P_CS_MON']	'P_ACW',
'P_EL_DEBIASED',		'NVOTE_MR2',		'P_EDGE',
'P_CS_DEBIASED',		'P_EL_MR2',		'P_DK',
'SPIRAL',		'P_CW_MR2',		'P_MG',
'ELLIPTICAL',		'P_ACW_MR2',		'P_CS']
'UNCERTAIN'		'P_EDGE_MR2',		
		'P_DK_MR2',		
		'P_MG_MR2',		
		'P_CS_MR2'		

Tabla 1. Columnas de los diferentes data sets de Galaxy Zoo

Las características que se encuentran en la tabla 1 son importantes en el análisis de datos astronómicos, ya que proporcionan información clave para identificar y caracterizar objetos celestes.

- **El OBJID:** es un identificador único para cada objeto observado, lo que facilita la identificación y seguimiento del mismo, a través de diferentes observaciones.
- **La RA y DEC:** proporcionan la posición precisa de un objeto en el cielo, lo que permite a los astrónomos apuntar sus telescopios y hacer observaciones detalladas del objeto.
- **El REDSHIFT:** es una herramienta importante para medir la distancia y la velocidad de los objetos astronómicos, lo que ayuda a los astrónomos a entender mejor la estructura del universo y su evolución.

- La morfología de las galaxias es importante para entender su formación y evolución, y la clasificación BPT se utiliza para identificar diferentes tipos de galaxias según sus emisiones espectrales.
- Las magnitudes SDSS y la masa estelar proporcionan información sobre la luminosidad y la masa de los objetos, lo que permite comparar diferentes objetos y entender sus propiedades.
- Finalmente, la dispersión de velocidad estelar y la luminosidad corregida por extinción son medidas importantes para caracterizar la composición y dinámica de los objetos celestes. En resumen, estas características son fundamentales para el análisis y clasificación de objetos celestes en el universo observable.

2.1.3 Proceso de unificación y limpieza de los datos.

El objetivo principal de esta parte del proceso es crear una tabla maestra que contenga todos los campos presentes en la tabla 1 y que los registros sean lo más limpios posible. Esta tabla será esencial para facilitar y optimizar la implementación del modelo de regresión logística multiclase al final del proyecto. Al contar con una tabla maestra completa y limpia, se podrá manipular y procesar la información de manera más eficiente, lo que contribuirá a mejorar la precisión y eficacia de la clasificación de las galaxias. En definitiva, el éxito del proyecto depende en gran medida de la creación de una tabla maestra de alta calidad.

Para el proceso de unificación y limpieza de los datos se realizaron los siguientes pasos:

Paso1: En la tabla 1, se puede observar que varios campos se repiten, por lo que se ha decidido conservar solo aquellas tablas que aporten información significativa al conjunto de datos principal. Para este ejercicio, se han seleccionado las tablas 2, 4 y 7. Obteniendo como resultado un Data Frame (DF) con un tamaño de (667.944 filas, 29 columnas).

Paso 2: Traemos solo campos únicos de cada data set, para no generar duplicidad de columnas.

Paso 3: Del marco de datos final, separamos la columna a predecir, que contiene la clasificación de las galaxias en espirales, elípticas e indefinidas**.**

Paso 4: Confirmamos que no se tengan campos nulos en nuestras columnas finales y implementamos la función describe para hacer un reporte que nos muestre la distribución de cada una de las columnas.

En la siguiente Tabla 2 se observa los resultados de la función describe de Python, donde se observan los valores máximos, mínimos, la desviación estándar, los cuartiles, la media y el numero de registros diferentes de nulos, de cada una de las columnas.

Index	STDCORR_CLEAN	F_MISCLASS_GREATER	AVCORR_GREATER	STDCORR_GREATER	NVOTE_TOT	NVOTE_STD	NVOTE_MR1	NVOTE_MR2	NVOTE_MON
count	6.68e+05	6.68e+05	6.68e+05	6.68e+05	6.68e+05	6.68e+05	6.68e+05	6.68e+05	6.68e+05
mean	0.0903	0.186	0.124	0.0617	76.2	38.8	12.5	12.5	12.5
std	0.0639	0.167	0.0866	0.0386	110	13.8	36.5	36.5	36.4
min	0	0	0	0	4	4	0	0	0
25%	0.02	0.036	0.047	0.028	29	28	0	0	0
50%	0.104	0.144	0.122	0.067	35	34	0	0	0
75%	0.146	0.296	0.197	0.096	57	51	0	0	0
max	0.225	1	0.308	0.208	780	94	245	259	251

Tabla 2. Describe del DF final, sin ningún tipo de procesamiento.

En la Figura 1 se evidencia que las variables NVOTE, NVOTE_STD, Y NVOTE_TOT, presentan una escala de distribución muy grande comparado con las demás variables. Además, las variables NVOTR_TOT, NVOTE_MR1, NVOTE_MR2 y NVOTE_MON presenta una desviación estándar muy marcada respecto a las demás variables.

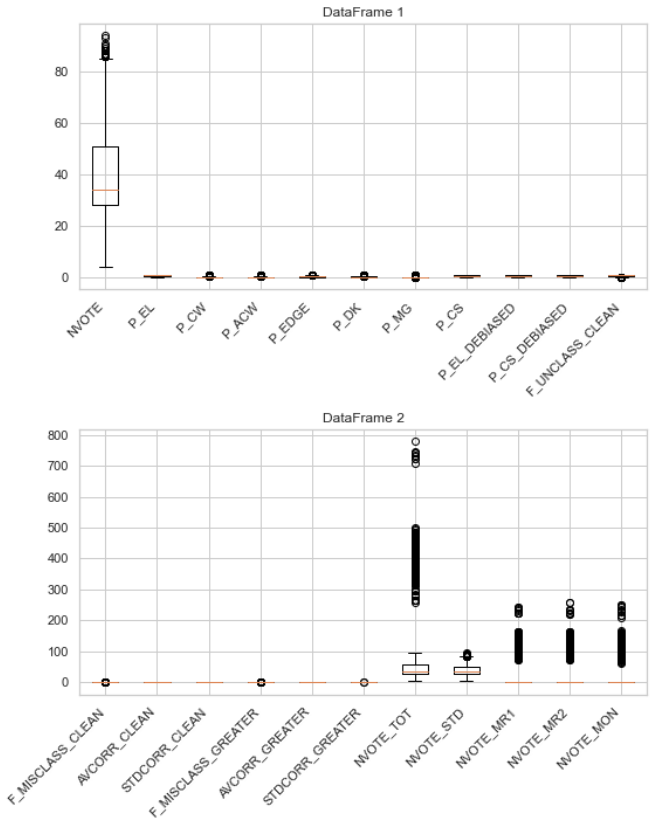

Figura 1. Diagramas de bigotes que representan la distribución sin procesar de los diferentes campos del DataFrame.

En la Figura 2 se observa cómo se distribuyen las muestras según su clasificación en espiral, elíptica e indefinida.

- Galaxia UNCERTAIN : 415529
- Galaxia SPIRAL: 190225
- Galaxia ELLIPTICAL: 62190

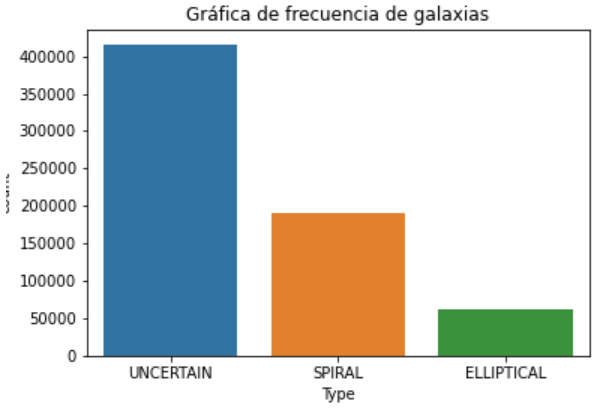

Figura 2. Gráfico de barras que ilustra la frecuencia en la que aparece la galaxia en el DataFrame.

Paso 5: En este proyecto, se ha implementado una normalización de las variables utilizando el método Min-Max de Python. Este método permite escalar todas las variables a un rango de 0 a 1, como se observa en la Figura 3, lo que facilita la comparación entre ellas y reduce el impacto de las variables con valores extremos en el modelo.

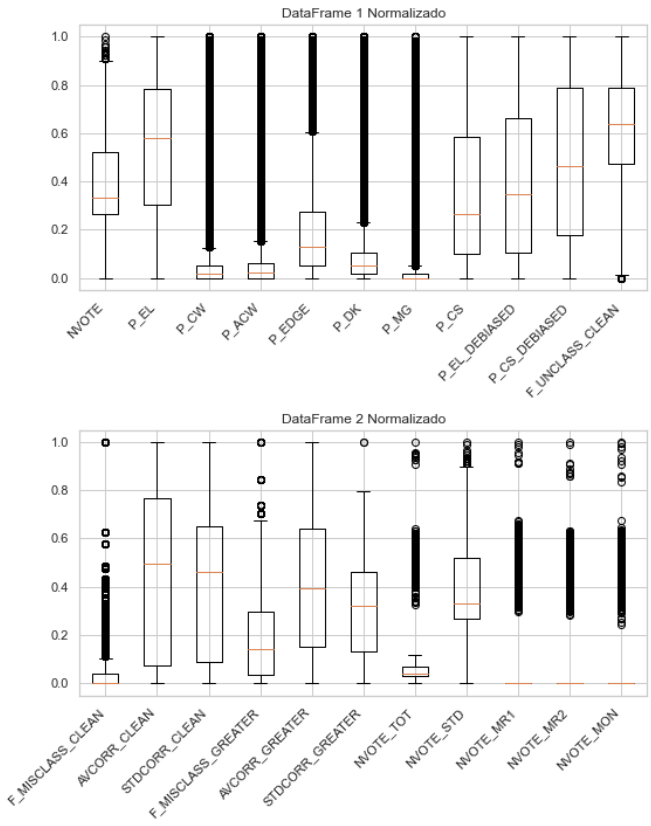

Figura 3**.** Representa el diagramade bigotes con los datos normalizados en escala de 0 a 1.

Paso6: Para realizar una identificación de la linealidad entre las variables y llevar a cabo un proceso de limpieza efectivo, se utilizó el método de correlación. Este proceso de análisis de correlación nos ha permitido evaluar la fuerza y la dirección de la relación entre cada par de variables, y así determinar cuáles de ellas están altamente correlacionadas y podrían redundar en la predicción.

La Tabla 3 presenta la matriz de correlación, donde los cuadros que tienden al color azul indican una correlación entre el 50% y el 100%, mientras que los cuadros que tienden al color rojo indican correlaciones por debajo del 50%.

correlacion - DataFrame

Index	NVOTE	P_EL	P_CW	P_ACW	P_EDGE	P_DK	P_MG	P_CS	_EL_DEBIASE	_CS_DEBIASE	NCLASS_CLI	ISCLASS_CL	'CORR_CLEA	'CORR_CLE	SCLASS_GRE	'ORR_GREA	CORR_GREA	NVOTE_TOT	NVOTE_STD	NVOTE_MR1	NVOTE_MR2	NVOTE_MON
NVOTE	100	-0.0662	-0.0152	-0.167	0.727	-0.427	-0.688	0.404	-0.518	0.607	0.036	0.402	0.219	0.178	0.33	0.425	0.401	13	100	0.464	0.445	0.45
P_EL	-0.0662	100	-42.1	-43.2	-54.9	2.86	-19.1	-90.3	90.3	-81.2	28.4	22.9	24.5	24	-3.78	-7.85	-15.6	9.38	-0.0662	9.45	9.44	9.44
P_CW	-0.0152	-42.1	100	-10.6	-8.72	-16.1	-3.44	45.7	-35.7	39.8	-25	-11.3	-15.9	-13.6	-6.56	-4.05	1.82	0.561	-0.0152	0.565	0.561	0.571
P_ACW	-0.167	-43.2	-10.6	100	-9.06	-16.7	-4.11	47.2	-36.9	41.3	-25.2	-11.5	-15.9	-13.6	-6.27	-3.64	2.27	1.18	-0.167	1.2	1.23	1.2
P_EDGE	0.727	-54.9	-8.72	-9.06	100	-20.1	-13.4	62	-56.3	62.9	-3.63	-13.6	-6.68	-8.6	14	17.2	19.8	-8.14	0.727	-8.28	-8.28	-8.27
P_DK	-0.427	2.86	-16.1	-16.7	-20.1	100	8.66	-34	3.07	-35	20.9	6.06	4.2	0.169	6.58	1.32	-4.85	-8.5	-0.427	-8.48	-8.51	-8.48
P_MG	-0.688	-19.1	-3.44	-4.11	-13.4	8.66	100	-14.2	-10.4	-20.9	-2.46	-0.844	-2.81	-1.99	-2.89	-2.86	-2	-5.4	-0.688	-5.35	-5.34	-5.35
P_CS	0.404	-90.3	45.7	47.2	62	-34	-14.2	100	-83.8	93.6	-32.3	-23.3	-23.7	-22.3	2.57	7.96	16.8	-4.87	0.404	-4.96	-4.94	-4.95
P_EL_DEBIASED	-0.518	90.3	-35.7	-36.9	-56.3	3.07	-10.4	-83.8	100	-90.1	22.2	8.73	-0.734	0.0254	-37.3	-42.2	-47.2	14.7	-0.518	14.8	14.8	14.8
P_CS_DEBIASED	0.607	-81.2	39.8	41.3	62.9	-35	-20.9	93.6	-90.1	100	-26.8	-12.4	-3.62	-3.12	27.9	34.3	41.1	-9.16	0.607	-9.29	-9.28	-9.29
F_UNCLASS_CLEAN	0.036	28.4	-25	-25.2	-3.63	20.9	-2.46	-32.3	22.2	-26.8	100	29.3	44	38.9	15.9	12.2	3.25	-8.84	0.036	-8.9	-8.9	-8.87
F_MISCLASS_CLEAN	0.402	22.9	-11.3	-11.5	-13.6	6.06	-0.844	-23.3	8.73	-12.4	29.3	100	58.6	59.9	31.1	34.3	30.6	-1.85	0.402	-1.9	-1.91	-1.93
AVCORR_CLEAN	0.219	24.5	-15.9	-15.9	-6.68	4.2	-2.81	-23.7	-0.734	-3.62	44	58.6	100	97.6	65.4	68.3	57.8	-6.59	0.219	-6.66	-6.65	-6.66
STDCORR_CLEAN	0.178	24	-13.6	-13.6	-8.6	0.169	-1.99	-22.3	0.0254	-3.12	38.9	59.9	97.6	100	58.9	65.4	60.1	-5.1	0.178	-5.15	-5.14	-5.16
F_MISCLASS_GREATER	0.33	-3.78	-6.56	-6.27	14	6.58	-2.89	2.57	-37.3	27.9	15.9	31.1	65.4	58.9	100	94.7	82.4	-9.17	0.33	-9.26	-9.26	-9.26
AVCORR_GREATER	0.425	-7.85	-4.05	-3.64	17.2	1.32	-2.86	7.96	-42.2	34.3	12.2	34.3	68.3	65.4	94.7	100	94.8	-9.35	0.425	-9.45	-9.44	-9.46
STDCORR_GREATER	0.401	-15.6	1.82	2.27	19.8	-4.85	-2	16.8	-47.2	41.1	3.25	30.6	57.8	60.1	82.4	94.8	100	-7.81	0.401	-7.91	-7.89	-7.92
NVOTE_TOT	13	9.38	0.561	1.18	-8.14	-8.5	-5.4	-4.87	14.7	-9.16	-8.84	-1.85	-6.59	-5.1	-9.17	-9.35	-7.81	100	13	98.9	98.9	98.9
NVOTE_STD	100	-0.0662	-0.0152	-0.167	0.727	-0.427	-0.688	0.404	-0.518	0.607	0.036	0.402	0.219	0.178	0.33	0.425	0.401	13	100	0.464	0.445	0.45
NVOTE_MR1	0.464	9.45	0.565	1.2	-8.28	-8.48	-5.35	-4.96	14.8	-9.29	-8.9	-1.9	-6.66	-5.15	-9.26	-9.45	-7.91	98.9	0.464	100	99.1	99.1
NVOTE_MR2	0.445	9.44	0.561	1.23	-8.28	-8.51	-5.34	-4.94	14.8	-9.28	-8.9	-1.91	-6.65	-5.14	-9.26	-9.44	-7.89	98.9	0.445	99.1	100	99
NVOTE_MON	0.45	9.44	0.571	1.2	-8.27	-8.48	-5.35	-4.95	14.8	-9.29	-8.87	-1.93	-6.66	-5.16	-9.26	-9.46	-7.92	98.9	0.45	99.1	99	100

Tabla 3: Representación de la Correlación entre variables del data set.

En este caso, se decidido descartar aquellas variables que tengan una correlación entre ellas superior o igual al 95%. Este criterio de selección se basa en la idea de que cuando dos variables están altamente correlacionadas, la inclusión de ambas en el modelo podría redundar en una disminución de la precisión y un aumento en la complejidad del modelo.

En este orden de ideas se descartarán las variables: NVOTE_STD, AVCORR_CLEAN, NVOTE_TOT, NVOTE_MR1, NVOTE_MR2, ya que aportan la misma información que las variables NVOTE_MON, NVOTE. Quedando con un data set de (667.944, 17)

Paso 7: Para llevar a cabo el proceso de eliminación de datos atípicos, se utilizó el método LOF (Local Outlier Factor), el cual nos permite detectar aquellos puntos que se alejan significativamente del patrón general de los datos.

A través de este método, se han identificado un total de 66.795 datos los cuales fueron eliminados posteriormente con el fin de reducir su impacto en la precisión del modelo. El detalle de los resultados se encuentra en la Figura 4.

Al eliminar los datos atípicos, se asegura que nuestro modelo se base en patrones y tendencias reales del conjunto de datos, y no en valores extremos o atípicos que puedan sesgar la predicción. De esta manera, podemos obtener resultados más precisos y confiables al momento de realizar predicciones sobre los datos.

```
In [73]: data_outlier_LOF = Detection_Outlier_LOF(df_nor1, n_neighbors=5, contamination =  0.10)
 El DataFrame sin outliers aplicando el método LOF está almacendo en el objeto: data_sin_outilers_LOF

 Número de muestras o filas con datos atípicos: 66795

 Aplicando el método LOF el promedio de la variacion absoluta de las entropias es: 0.001438

                      Entropia Original  Entropia LOF   dif_abs
P_EL                          0.239153      0.241747  0.002594
P_CW                          0.086943      0.086794  0.000149
P_ACW                         0.096932      0.097144  0.000212
P_EDGE                        0.225324      0.229781  0.004457
P_DK                          0.150647      0.150956  0.000309
P_MG                          0.051674      0.049379  0.002295
P_CS                          0.239549      0.242238  0.002688
P_EL_DEBIASED                 0.235585      0.237773  0.002188
P_CS_DEBIASED                 0.226799      0.227985  0.001186
F_UNCLASS_CLEAN               0.255533      0.257341  0.001808
F_MISCLASS_CLEAN              0.073591      0.073572  0.000019
STDCORR_CLEAN                 0.226047      0.227527  0.001480
F_MISCLASS_GREATER            0.229484      0.230991  0.001506
AVCORR_GREATER                0.243399      0.243753  0.000354
STDCORR_GREATER               0.283083      0.284075  0.000992
NVOTE_STD                     0.338319      0.339510  0.001192
NVOTE_MON                     0.037433      0.036411  0.001021
```

Figura 4. Resultados de la implementación del método LOF.

La Figura 5.A presenta la distribución de las variables después de haber eliminado los valores atípicos, mientras que en la Figura 5.B muestra la misma distribución, pero esta vez con los valores atípicos incluidos. Al observar visualmente ambas figuras, podemos notar que el método LOF aplicado a variables como NVOTE, logra reducir significativamente la cantidad de valores atípicos presentes en los datos.

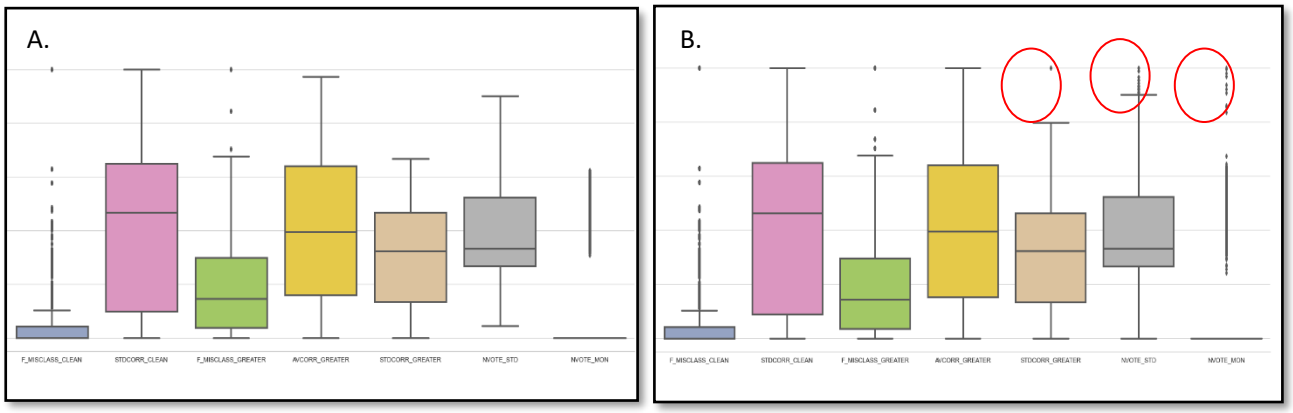

Figura 5. A. Distribución de las variables después de eliminar datos atípicos. B. Distribución de las variables antes de eliminar datos atípicos.

2.1.4 Implementación de PCA.

En resumen, PCA es una técnica de análisis multivariado que se utiliza para reducir la dimensionalidad de los datos, identificando los componentes principales que describen la mayor cantidad posible de variabilidad en los datos.

Para este proyecto, se implementó el método de PCA para reducir la dimensionalidad del conjunto de datos. Después de aplicar PCA, se calculó la tasa de varianza explicada por cada una de las componentes, lo que permitió identificar las componentes principales con los que se trabajaría.

En la Figura 6 se puede observar que la tasa de varianza explicada disminuye a medida que aumenta el número de componentes. Se observa que las dos primeras componentes principales explican la mayor parte de la variabilidad en los datos, y que, a partir de la tercera componente, la tasa de varianza explicada disminuye significativamente.

Por lo tanto, se decidió trabajar con las dos primeras componentes principales, ya que estas componentes explican la mayor parte de la variabilidad en los datos y permiten una representación más eficiente y compacta del conjunto de datos en un espacio de menor dimensión. Esto facilita la visualización y exploración del conjunto de datos, así como la implementación de modelos de aprendizaje automático con un menor número de variables.

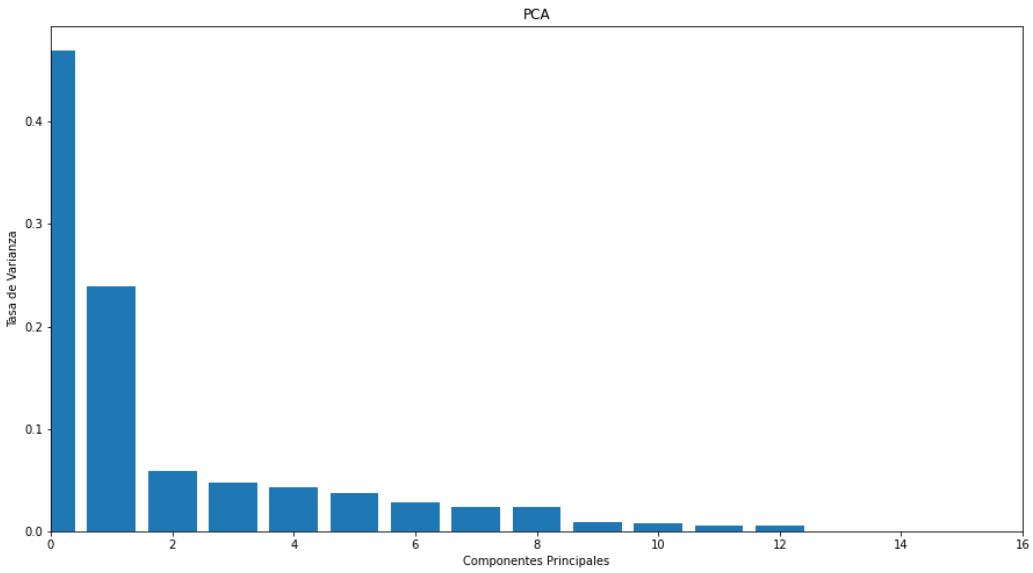

Figura 6. Varianza de las componentes principales.

Para identificar las variables más influyentes en PCA usando Python, se implementó la función *pca.components_* de la *librería sklearn.decomposition*.

En la Tabla 4 se observa cuales son las variables que mas aportan a cada una de las componentes principales, las variables de color azul son las mas influyentes y las de color rojo son las que menos aportan en la construcción de las componentes principales.

CP 1

P_CS_DEBIASED	0.507
P_EL_DEBIASED	0.47
P_CS	0.455
P_EL	0.417
P_EDGE	0.221
AVCORR_GREATER	0.165
STDCORR_GREATER	0.13
P_ACW	0.12
P_CW	0.113
F_UNCLASS_CLEAN	0.105
F_MISCLASS_GREATER	0.0806
P_DK	0.0308
STDCORR_CLEAN	0.0271
NVOTE_MON	0.0259
F_MISCLASS_CLEAN	0.0243
P_MG	0.00643
NVOTE_STD	0.000409

CP 2

AVCORR_GREATER	0.574
STDCORR_CLEAN	0.57
STDCORR_GREATER	0.347
F_MISCLASS_GREATER	0.325
F_UNCLASS_CLEAN	0.173
P_CS	0.171
P_EL	0.161
F_MISCLASS_CLEAN	0.128
P_ACW	0.0798
P_CW	0.0777
P_EL_DEBIASED	0.0648
NVOTE_MON	0.0247
P_DK	0.0156
P_EDGE	0.0139
P_CS_DEBIASED	0.0133
P_MG	0.00555
NVOTE_STD	0.00103

Tabla 4. Variables que más peso aportaron para CP1 y CP2.

2.1.5 Implementación del modelo de regresión logística con PCA.

Una vez aplicado el análisis de componentes principales (PCA), las componentes principales CP1 y CP2 se usan como entrada para el modelo de regresión lineal multidimensional.

En este caso, dividimos el conjunto de datos en dos conjuntos, de la siguiente manera: el 80% (480919 registros) del conjunto de datos se utiliza como datos de entrenamiento, mientras que el 20% (120230 registros) restante se reserva para las pruebas.

- **Resultados del modelo de regresión logística con el conjunto de datos de entrenamiento.**

Usando el conjunto de datos de entrenamiento se obtuvo un porcentaje de precisión (Accuracy) del 89% para la base de entrenamiento.

En la Figura 7 se muestra la matriz de confusión para los resultados del modelo utilizando el conjunto de datos de entrenamiento.

- El 88% de las galaxias elípticas fueron clasificadas correctamente.
- El 83% de las galaxias espirales fueron clasificadas correctamente.
- El 91% de las galaxias con forma irregular fueron clasificadas correctamente.

La matriz de confusión se asemeja a un mapa de calor en el cual los tonos más claros indican las áreas con mayor cantidad de registros, mientras que los tonos más oscuros señalan las zonas con menos registros.

```
==========Accuracy Logistic Regression con PCA  Train =====
multinomial - lbfgs: 0.8892225094038705
```

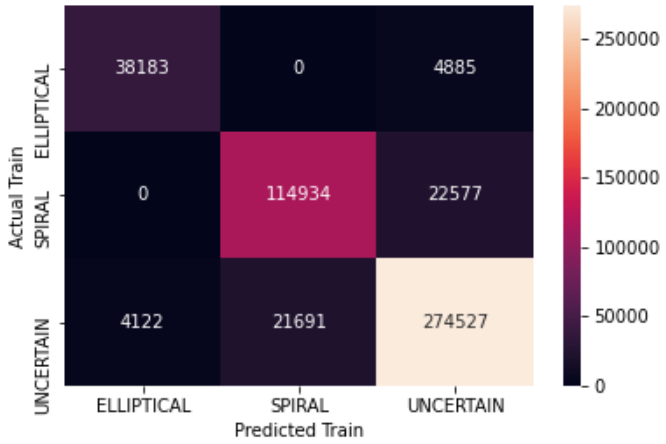

Figura 7. Matriz de confusión del set de entrenamiento con PCA.

- **Resultados del modelo de regresión logística con el conjunto de datos de pruebas.**

Como resultado se obtuvo un accuracy del 89% para la base de prueba. La Figura 8 muestra la matriz de confusión para los resultados del modelo utilizando el conjunto de datos de pruebas.

- El 88% de las galaxias elípticas fueron clasificadas correctamente.
- El 83% de las galaxias espirales fueron clasificadas correctamente.
- El 91% de las galaxias con forma irregular fueron clasificadas correctamente.

Los colores más claros en la matriz de confusión indican la agrupación de la mayor cantidad de registros, mientras que los colores más oscuros indican una menor cantidad de registros.

```
==========Accuracy Logistic Regression con PCA Test ===
multinomial - lbfgs: 0.8882059386176495
```

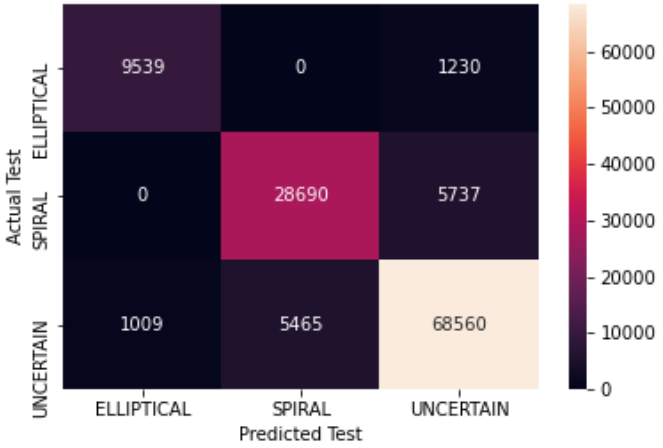

Figura 8**.** Matriz de confusión del set de pruebas con PCA.

2.1.6 Implementación del modelo de regresión logística sin PCA.

Repetimos el experimento, esta ves con la información original y obtuvimos un Accuracy para el set de train del 95.9% y para el set de text del 96%.

- **Resultados del modelo de regresión logística con el conjunto de datos de Entrenamiento.**

Utilizando un conjunto de datos de entrenamiento que representa el 80% del total de la base de datos (480,919 registros), logramos obtener un nivel de precisión del 95.9%. En la Figura 9, se presenta la matriz de confusión que nos permite interpretar lo siguiente:

- El 93% de las galaxias elípticas fueron clasificadas correctamente.
- El 95% de las galaxias espirales fueron clasificadas correctamente.
- El 97% de las galaxias con forma irregular fueron clasificadas correctamente

```
==========Accuracy Logistic Regression train ================
multinomial - lbfgs: 0.9599724693763398
```

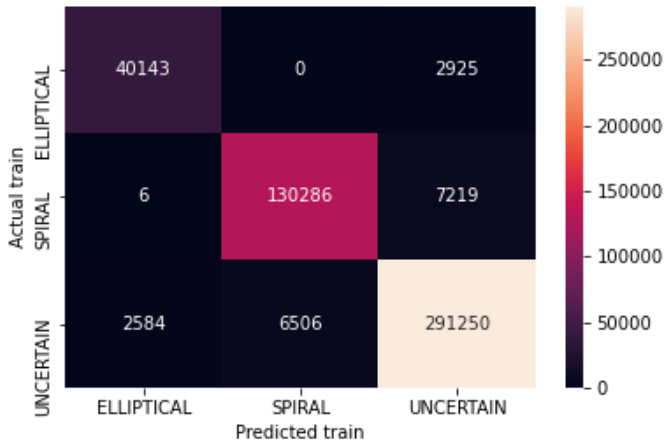

Figura 9. Matriz de confusión del set de entrenamiento con los registros originales.

- **Resultados del modelo de regresión logística con el conjunto de datos de prueba.**

Al emplear el conjunto de datos de pruebas, que representa el 20% del total de registros, logramos alcanzar una precisión de predicción del 96%. En la Figura 10, se muestra la matriz de confusión, la cual brinda una visualización detallada de los resultados obtenidos por el modelo de regresión logística. Esto nos permite analizar de manera más precisa los resultados de la predicción.

- El 93% de las galaxias elípticas fueron clasificadas correctamente.
- El 95% de las galaxias espirales fueron clasificadas correctamente.
- El 97% de las galaxias con forma irregular fueron clasificadas correctamente.

```
==========Accuracy Logistic Regression test ================
multinomial - lbfgs: 0.9601264243533228
```

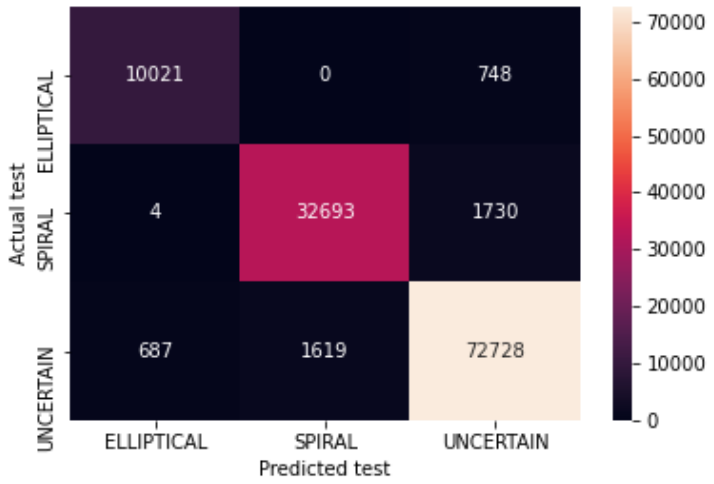

Figura 10: Matriz de confusión usando todas las variables del data set.

2.1.7 Conclusiones

En este caso, se puede interpretar que el modelo de regresión lineal multidimensional sin utilizar el método de PCA produce resultados muy similares al modelo que incluyó el PCA. El hecho de que ambos modelos tengan una precisión (accuracy) superior al 85% en el conjunto de prueba sugiere que ambos modelos pueden predecir con precisión la clasificación de las galaxias.

Sin embargo, cabe mencionar que el modelo sin PCA probablemente utilice más variables en su análisis, lo que puede aumentar la complejidad del modelo y dificultar la interpretación de los resultados. Por lo tanto, en este caso, el uso de PCA puede ser beneficioso para reducir la dimensionalidad de los datos y simplificar el modelo, sin comprometer la precisión de las predicciones.

En conclusión este modelo ha demostrado ser efectivo y ampliamente utilizado en tareas de clasificación debido a su simplicidad y capacidad para manejar variables predictoras.

Capítulo 3

En este capítulo, se realiza la implementación de dos modelos de redes convolucionales: ResNet50V2 y VGG19.

3.1.1. Implementación del modelo ResNet y VGG19 usando tensorflow keras.

El proyecto Galaxy Zoo tiene una plataforma en línea que pone a disposición una galería de imágenes de galaxias, cada imagen se encuentra asociada a un objecid el cual nos permite identificar el tipo y características de la galaxia, logrando así hacer un proceso de etiquetamiento a las imágenes.

3.1.2. Descripción de imágenes

Para implementar los modelos de VGG19 y ResNet50v2, se dispone de un conjunto de 295.305 imágenes, cada una con las siguientes características:

- Formato: .jpg
- Dimensiones: 424 píxeles de ancho por 424 píxeles de alto.
- Resolución: 96 píxeles por pulgada (ppp) tanto en sentido horizontal como vertical.
- Profundidad de color: 24 bits.

En la Figura 11 se muestra un ejemplo representativo del tipo de imágenes que serán procesadas por estos modelos.

Figura 11. Muestra de las imágenes originales del data set de imágenes de Galaxy Zoo.

3.1.3. Proceso de transformación y etiquetado de las imágenes

Para la implementar los modelos VGG19 y ResNet50V2, se llevó a cabo un preprocesamiento de las imágenes. En dicho proceso, se convirtieron todas las imágenes a escala de grises Figura 12, esto se realizo con el objetivo de mejorar los resultados de los modelos al momento de clasificar las imágenes. Luego se cargaron las imágenes usando Python y se realizó un proceso de etiquetado a las imágenes.

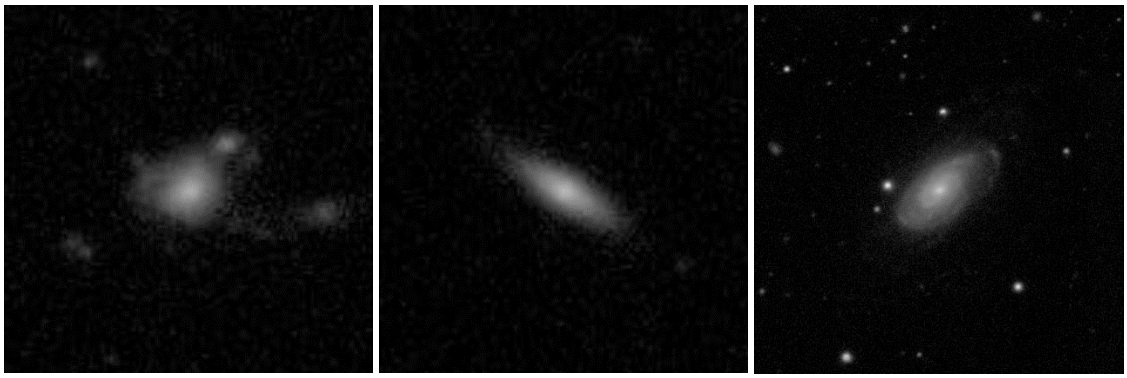

Figura 12. Imágenes de galaxias reprocesadas en escala de grises.

Para el proceso de etiquetado disponemos de un archivo csv que nos indica el nombre de la imagen y el objetid de la imagen y el tipo de galaxi asociado a la imagen. En la Figura 13 se muestra el formato del archivo que se usa para el proceso de etiquetado.

- Objetid: Identificador único de la galaxia.
- Sample: indica que la imagen es original del proyecto de Galaxy zoo.
- Asset_id: es el nombre de a imagen.
- Type: clasificación de la galaxia es spiral, elliptica y uncertain
- Spiral: muestra un 1 cuando la galaxia es de tipo spiral, de lo contrario es 0
- Elliptical: muestra un 1 cuando la galaxia es de tipo Elliptical, de lo contrario es 0
- Uncertain: muestra un 1 cuando la galaxia es de tipo Uncertain, de lo contrario es 0

A	B	C	D	E	F	G
OBJID	sample	asset_id	Type	SPIRAL	ELLIPTICAL	UNCERTAIN
5,87723E+17	original	3	UNCERTAIN	0	0	1
5,87723E+17	original	4	ELLIPTICAL	0	1	0
5,87723E+17	original	5	ELLIPTICAL	0	1	0
5,87723E+17	original	6	UNCERTAIN	0	0	1
5,87723E+17	original	7	UNCERTAIN	0	0	1
5,87723E+17	original	8	UNCERTAIN	0	0	1
5,87723E+17	original	9	UNCERTAIN	0	0	1
5,87723E+17	original	11	ELLIPTICAL	0	1	0
5,87723E+17	original	12	SPIRAL	1	0	0
5,87723E+17	original	13	UNCERTAIN	0	0	1
5,87723E+17	original	14	UNCERTAIN	0	0	1
5,87723E+17	original	15	UNCERTAIN	0	0	1

Figura 13. Formato del archivo csv que se usó para el proceso de etiquetado

3.1.4. Implementación del modelo ResNet 50 v2 y VGG19

Una vez que las imágenes fueron previamente procesadas y etiquetadas, se procedió a tomar una muestra de 20.000 imágenes. Esta muestra se dividió en dos conjuntos de imágenes, asignando el 80% de las imágenes al conjunto de pruebas y el 20% restante al conjunto de validación. Cada Modelo se entrenó con 10 épocas.

Ajustes para el modelo de ResNet50 V2

El modelo ResNet50V2 se configuro para trabajar como un extractor de características pre-entrenado en el conjunto de datos ImageNet, utilizando imágenes de entrada de 224x224 píxeles y 3 canales de color. Se entrena para clasificar imágenes en tres categorías diferentes y utiliza la función de activación softmax para generar probabilidades de clasificación. *(Team, K. (s. f.-a)).*

Para el modelo de ResNet50 V2 se usó los siguientes hiperparámetro:

- **include _top= False:** esto indica que no se quiere incluir la capa complementaria.
- **Weights = 'imagenet':** Indica que se utilizarán los pesos pre-entrenados en el conjunto de datos ImageNet. Estos pesos pre-entrenados ayudan al modelo a iniciar con una representación visual aprendida de una amplia variedad de imágenes.
- **input_tensor=None:** Significa que se utilizará el tensor de entrada predeterminado.
- **input_shape=(224, 224,1):** Lo que indica que se esperan imágenes de entrada con una resolución de 224x224 píxeles y 1 canales de color.
- **pooling=None:** Significa que no se realizará agrupación después de la etapa de convolución.
- **classes=3:** Significa que el modelo se entrenará para clasificar imágenes en tres categorías diferentes.

- **classifier_activation="softmax":** indica que se utilizará la función softmax para obtener las probabilidades de las diferentes clases en la clasificación final.

Optimizador SGD (Stochastic Gradient Descent)

este optimizador SGD con una tasa de aprendizaje baja y un momentum moderado se utilizará para ajustar los pesos del modelo durante el entrenamiento. La tasa de aprendizaje baja ayuda a un entrenamiento más estable y preciso, mientras que el momentum ayuda a acelerar el proceso de optimización. *(Team, s. f.-a)*

Para el optimizador se uso los siguientes hiperparámetro:

- **learning_rate=0.0001:** Indica que los ajustes del peso del modelo en función del error serán muy pequeños en cada paso, lo que puede ayudar a un entrenamiento más lento y estable.
- **momentum=0.09:** Indica que se aplicará un impulso moderado para acelerar el proceso de optimización.

Ajustes para el modelo de VGG19:

El modelo VGG19 está diseñado para funcionar como un extractor de características que ha sido previamente entrenado en el conjunto de datos ImageNet. Utiliza imágenes de entrada con dimensiones de 224x224 píxeles y 3 canales de color. Durante el entrenamiento, el modelo aprende a clasificar imágenes en tres categorías distintas y utiliza la función de activación softmax para generar las probabilidades correspondientes a cada clase en la etapa de clasificación. *(Team, s. f.-b)*

Para el modelo de ResNet50 V2 se usó los siguientes hiperparámetro:

- **include _top= False:** esto indica que no se quiere incluir la capa complementaria.
- **Weights = 'imagenet':** Indica que se utilizarán los pesos pre-entrenados en el conjunto de datos ImageNet. Estos pesos pre-entrenados ayudan al modelo a iniciar con una representación visual aprendida de una amplia variedad de imágenes.
- **input_tensor=None:** Significa que se utilizará el tensor de entrada predeterminado.
- **input_shape=(224, 224, 3):** Lo que indica que se esperan imágenes de entrada con una resolución de 224x224 píxeles y 3 canales de color (rojo, verde y azul).

- **pooling=None:** Significa que no se realizará agrupación después de la etapa de convolución.
- **classes=3:** Significa que el modelo se entrenará para clasificar imágenes en tres categorías diferentes.
- **classifier_activation="softmax":** indica que se utilizará la función softmax para obtener las probabilidades de las diferentes clases en la clasificación final.

Optimizador Adam

el optimizador Adam con la tasa de aprendizaje especificada se utiliza para optimizar y ajustar los pesos del modelo durante el entrenamiento. El valor de learning_rate determina la magnitud de los ajustes realizados en cada paso del proceso de optimización. *(Team, K. (s. f.-a)).*

3.1.5 Resultados

Resultado del modelo ResNet50V2.

En la Figura 14 se presentan los resultados mediante las gráficas de Accuracy y función de pérdida. Se puede observar visualmente cómo mejora el rendimiento del modelo a medida que aumenta el número de épocas.

En cuanto a los resultados específicos, se logra un accuracy del 70% tanto para el conjunto de validación como para el conjunto de pruebas. Esto indica que el modelo es capaz de clasificar correctamente el 70% de las imágenes en ambos conjuntos.

En relación a la función de pérdida, se observa que comienza en un 87% y disminuye hasta alcanzar un 67% para el conjunto de pruebas. Para el conjunto de validación, la función de pérdida inicial es del 80% y también llega a un 67% al final. Estos valores indican que el modelo ha logrado reducir significativamente la pérdida y ha mejorado su capacidad de hacer predicciones más precisas.

Además, se destaca la estabilidad del modelo para ambos conjuntos de imágenes, lo que sugiere que el modelo es consistente y confiable en su rendimiento.

Estos resultados evidencian el buen desempeño y la capacidad predictiva del modelo implementado.

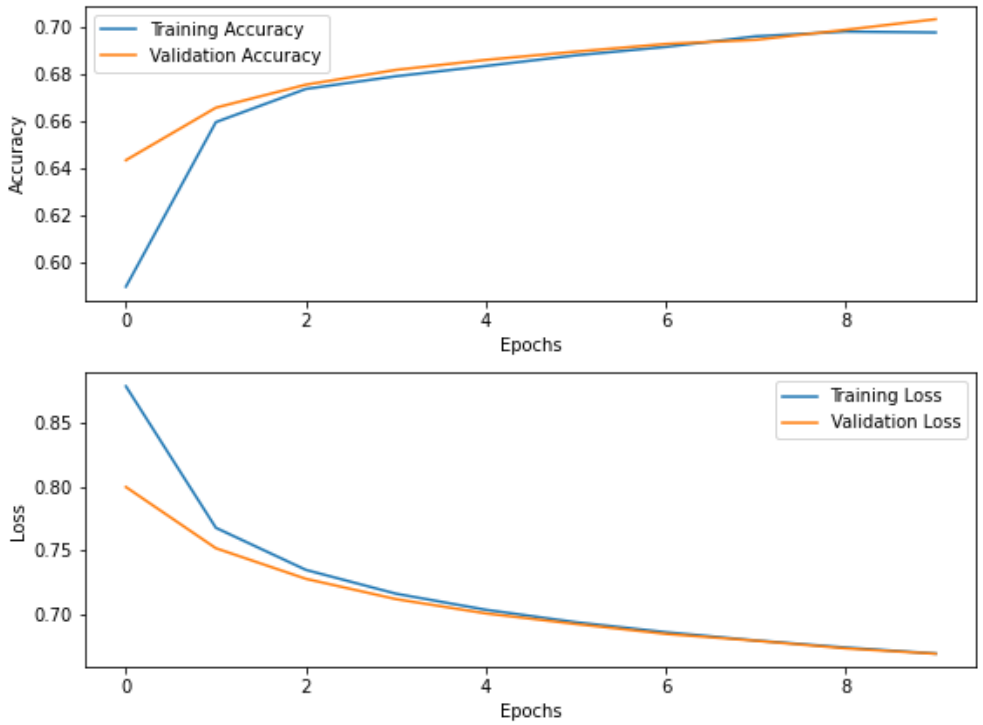

Figura 14. Resultados de precisión (accuracy) y función de pérdida del modelo de ResNet50V2.

En las Figuras 15 y 16 se presentan las matrices de confusión correspondientes al conjunto de pruebas y al conjunto de testing, respectivamente. Estas matrices permiten una visualización intuitiva de la cantidad de aciertos y errores del modelo en la clasificación de cada tipo de galaxia.

Para el conjunto de pruebas, se obtuvieron los siguientes resultados en la matriz de confusión:

- Clasificación "Uncertain": El 33.68% de las galaxias clasificadas como "Uncertain" fueron correctamente identificadas por el modelo.
- Clasificación "Elliptical": El 73.31% de las galaxias clasificadas como "Elliptical" fueron correctamente identificadas por el modelo.
- Clasificación "Spiral": El 78.08% de las galaxias clasificadas como "Spiral" fueron correctamente identificadas por el modelo.

Para el conjunto de validación, se obtuvieron los siguientes resultados en la matriz de confusión:

- Clasificación "Uncertain": El 32.13% de las galaxias clasificadas como "Uncertain" fueron correctamente identificadas por el modelo.
- Clasificación "Elliptical": El 79.90% de las galaxias clasificadas como "Elliptical" fueron correctamente identificadas por el modelo.

- Clasificación "Spiral": El 79.39% de las galaxias clasificadas como "Spiral" fueron correctamente identificadas por el modelo.

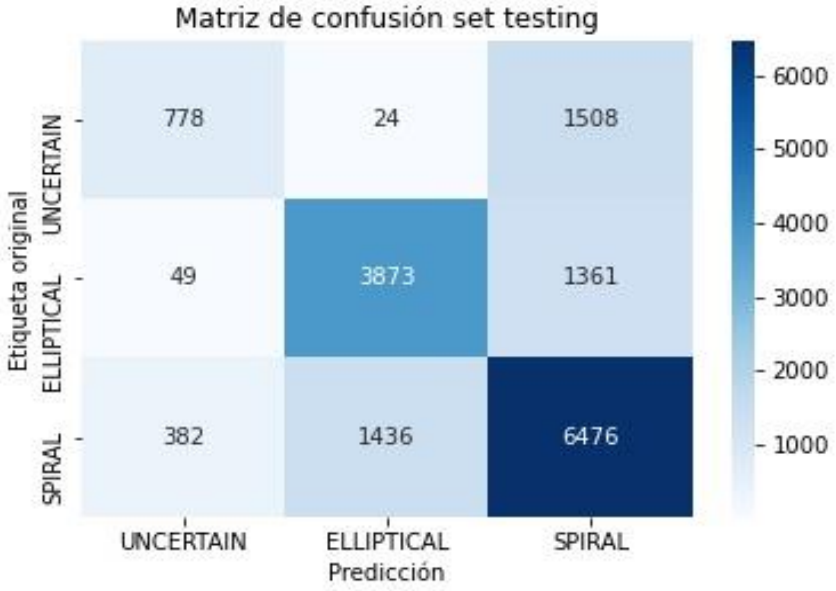

Figura 15. Matriz de confusión del conjunto de testing para el modelo ResNet50V2.

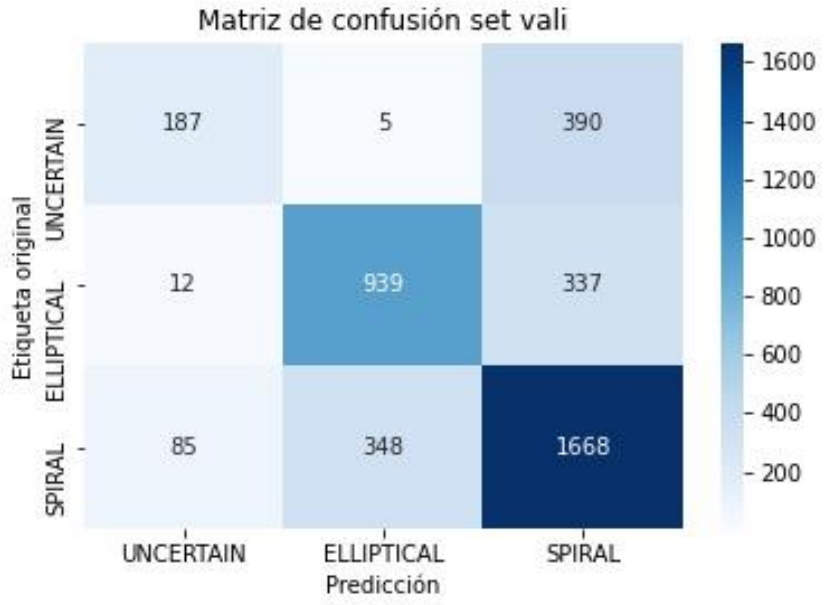

Figura 16. Matriz de confusión del conjunto de validación para el modelo ResNet50V2.

Resultado del modelo VGG19.

En la Figura 17 se muestra el comportamiento del accuracy y la función de pérdida del modelo VGG19.

En relación a los resultados, se obtiene un accuracy del 66% tanto para el conjunto de validación como para el conjunto de pruebas. Esto indica que el modelo es capaz de clasificar correctamente el 66% de las imágenes en ambos conjuntos.

En cuanto a la función de pérdida, se observa que inicialmente comienza en un 95% y disminuye hasta alcanzar un 75% para el conjunto de validación. Para el conjunto de pruebas, la función de pérdida inicial es del 90% y también baja hasta un 74%. Estos valores indican que el modelo ha logrado reducir la pérdida a lo largo del entrenamiento.

Sin embargo, se destaca la inestabilidad del modelo para el conjunto de validación, donde la función de pérdida muestra variaciones más pronunciadas. Esto sugiere que el modelo puede tener dificultades para generalizar correctamente a datos no vistos durante el entrenamiento, lo que puede afectar su rendimiento en situaciones del mundo real.

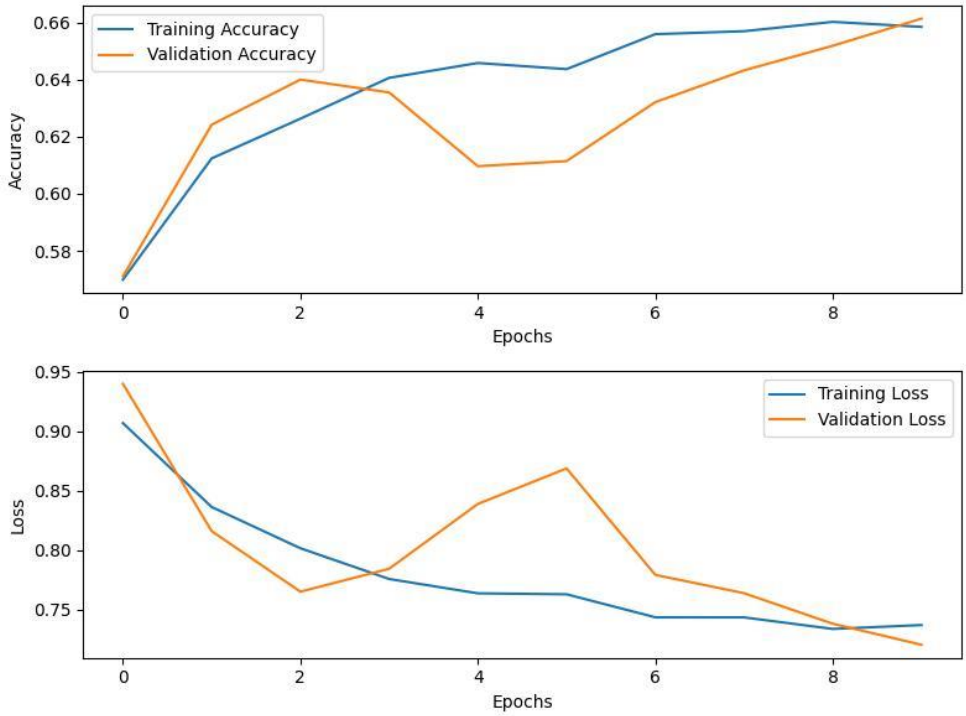

Figura 17. Resultados de precisión (accuracy) y función de pérdida del modelo de VGG19.

En la Figura 17 se muestra la matriz de confusión del modelo VGG19 para el conjunto de pruebas, y en la Figura 18 se muestra la matriz de confusión para el conjunto de validación. Estas matrices permiten visualizar de manera intuitiva la cantidad de aciertos y errores del modelo en la clasificación de cada tipo de galaxia.

En relación a los resultados obtenidos en el conjunto de pruebas, se observa lo siguiente en la matriz de confusión:

- Clasificación "Uncertain": El 24% de las galaxias clasificadas como "Uncertain" fueron correctamente identificadas por el modelo.
- Clasificación "Elliptical": El 63.75% de las galaxias clasificadas como "Elliptical" fueron correctamente identificadas por el modelo.
- Clasificación "Spiral": El 80.97% de las galaxias clasificadas como "Spiral" fueron correctamente identificadas por el modelo.

Y para el conjunto de validación se obtienen los siguientes resultados:

- Clasificación "Uncertain": El 25.97% de las galaxias clasificadas como "Uncertain" fueron correctamente identificadas por el modelo.
- Clasificación "Elliptical": El 59.72% de las galaxias clasificadas como "Elliptical" fueron correctamente identificadas por el modelo.
- Clasificación "Spiral": El 80.49% de las galaxias clasificadas como "Spiral" fueron correctamente identificadas por el modelo.

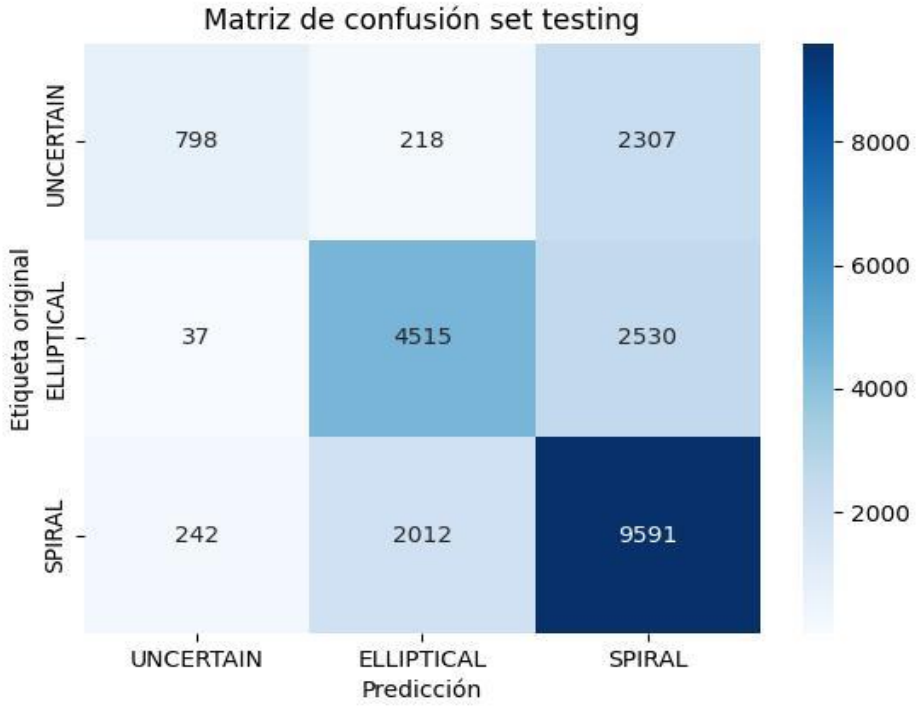

Figura 18. Matriz de confusión del conjunto de pruebas para el modelo VGG19.

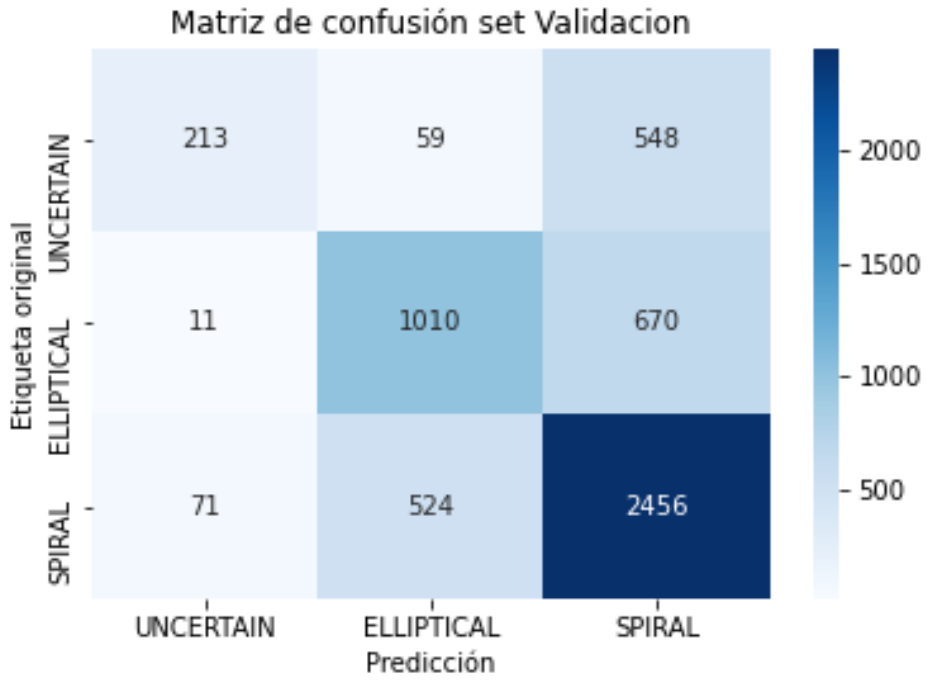

Figura 19. Matriz de confusión del conjunto de validación para el modelo VGG19.

3.1.5. Conclusiones

A nivel general los resultados del modelo ResNet Fue considerablemente mejor que los resultados del modelo VGG19.

Desempeño: El rendimiento de ambos modelos puede variar según el conjunto de datos y la tarea específica. Sin embargo, se observó que ResNet50V2 fue notablemente más eficaz en el aprendizaje profundo debido a su capacidad para mitigar la degradación del rendimiento en redes más profundas. Por otro lado, VGG19 mostró un desempeño menos favorable, ya que este modelo tiende a sufrir de sobreajuste en conjuntos de datos específicos. No obstante, no se puede afirmar con certeza que este haya sido el inconveniente en el contexto de este proyecto en particular.

Eficiencia computacional: ResNet50V2 tiene una eficiencia computacional relativamente mayor en comparación con VGG19 debido a la incorporación de conexiones residuales, lo que reduce el número de parámetros a aprender y mejora la propagación del gradiente durante el entrenamiento.

Uso de recursos: En términos de uso de recursos, VGG19 requiere más memoria y poder de cómputo debido a su mayor cantidad de capas y parámetros en comparación con ResNet50V2.

En conclusión, los modelos de redes convolucionales han demostrado ser altamente eficientes para la clasificación de imágenes de galaxias. En este proyecto, se ha identificado que el modelo ResNet50V2 muestra resultados superiores, un mejor desempeño y un rendimiento más destacado en comparación con el modelo VGG19.

Por otro lado, si bien el modelo VGG19 también logra resultados aceptables, se observa una mayor inestabilidad en el conjunto de validación, lo que podría indicar dificultades para generalizar correctamente a datos no vistos previamente.

4. Referencias

Zhu, X. P., Dai, J. M., Bian, C. J., Chen, Y., Chen, S., & Hu, C. (2019). Galaxy morphology classification with deep convolutional neural networks. *Astrophysics and Space Science*, *364*, 1-15.

Wikipedia contributors. (2023). Local outlier factor. Wikipedia. https://en.wikipedia.org/wiki/Local_outlier_factor

¿Qué es la regresión logística? | IBM. (s. f.). https://www.ibm.com/es-es/topics/logistic-regression

scikit-learn - sklearn.decomposition.PCA Análisis de componentes principales (ACP). (s. f.). https://runebook.dev/es/docs/scikit_learn/modules/generated/sklearn.decomposition.pca

Galaxy Zoo. (s. f.). https://zoo4.galaxyzoo.org/

Team, K. (s. f.-a). Keras documentation: ResNet and ResNetV2. https://keras.io/api/applications/resnet/

Team, K. (s. f.-b). Keras documentation: VGG16 and VGG19. https://keras.io/api/applications/vgg/

Team, K. (s. f.-a). Keras documentation: Optimizers. https://keras.io/api/optimizers/