

UNIVERSIDAD ADOLFO IBÁÑEZ

TESIS DE MAGÍSTER

---

**Algoritmo de filtración e identificación de estrellas  
binarias.**

---

*Autor:*

Luciano Godoi Caceres

*Profesor guía:*

Diego J. Muñoz

*Comité de defensa:*

Luis Aburto

Felipe Rojas

*Tesis realizada acorde a los requerimientos para el grado de  
Master of Science in Data Science*

*de la*

Facultad de Ingeniería y Ciencias

31 de julio de 2023

**UAI**

FACULTAD DE  
INGENIERÍA Y  
CIENCIAS

UNIVERSIDAD ADOLFO IBAÑEZ

## *Resumen*

Facultad de Ingeniería y Ciencias

Master of Science in Data Science

### **Algoritmo de filtración e identificación de estrellas binarias.**

por Luciano Godoi Caceres

Este estudio aborda el reto de identificar y clasificar estrellas binarias en astronomía, un desafío crítico dada la inmensa cantidad de datos astronómicos disponibles y la relevancia de las estrellas binarias para diversas aplicaciones científicas. A pesar de los esfuerzos actuales por mejorar los sistemas de almacenamiento y gestión de datos, persisten limitaciones en rendimiento y precisión, por lo que se propone un nuevo algoritmo de aprendizaje automático, más eficiente y capaz de superar estas barreras.

La metodología se basa en el Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD), que incluye la selección y preprocesamiento de datos, identificación de patrones y evaluación de modelos de clasificación. La transparencia y reproducibilidad se garantizan compartiendo el código generado en una biblioteca pública.

Los resultados demuestran que la regresión logística proporciona un rendimiento superior en la distinción entre estrellas binarias y alineaciones aleatorias en comparación con la Estimación de Densidad Kernel. El algoritmo propuesto se validó utilizando un catálogo de prueba de alineaciones estelares, demostrando alta precisión en la detección de estas. Este trabajo busca superar las limitaciones de los sistemas actuales y potenciar la gestión y interpretación de datos astronómicos masivos.

**Palabras clave:** Machine learning, Clasificación, Astronomía, Estrellas Binarias, Catálogo.

UNIVERSIDAD ADOLFO IBÁÑEZ

## *Abstract*

Faculty of Engineering and Science

Master of Science in Data Science

### **Algorithm for filtering and identification of binary stars**

by Luciano Godoi Caceres

This study tackles the challenge of identifying and classifying binary stars in astronomy, a critical issue given the immense amount of astronomical data available and the relevance of binary stars for various scientific applications. Despite current efforts to improve data storage and management systems, there remain limitations in performance and accuracy. Therefore, a new, more efficient machine learning algorithm is proposed, capable of overcoming these barriers.

The methodology is based on the Knowledge Discovery in Databases (KDD) process, which includes data selection and preprocessing, pattern identification, and classification model evaluation. Transparency and reproducibility are ensured by sharing the generated code in a public library.

The results demonstrate that logistic regression provides superior performance in distinguishing between binary stars and random alignments compared to Kernel Density Estimation. The proposed algorithm was validated using a test catalog of stellar alignments, showing high accuracy in their detection. This work aims to overcome the limitations of current systems and enhance the management and interpretation of massive astronomical data.

**Keywords:** Machine learning, Classification, Astronomy, Binary stars, Catalog.

# Índice general

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>II</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Definición del problema u oportunidad</b>	<b>4</b>
<b>3. Estado del arte</b>	<b>6</b>
<b>4. Hipótesis y objetivos</b>	<b>9</b>
<b>5. Metodología</b>	<b>11</b>
<b>6. Propuesta</b>	<b>14</b>
6.1. Descripción y Preprocesamiento de Variables Utilizadas de GAIA . . . . .	14
6.1.1. Descripción de las variables . . . . .	15
6.1.2. Preprocesamiento . . . . .	16
6.2. Algoritmo de filtración de candidatos binarios . . . . .	18
6.2.1. Búsqueda de vecinos cercanos: . . . . .	18
6.2.2. Selección de candidatos binarios: . . . . .	18
6.2.3. Búsqueda de vecinos sobre candidatos binarios: . . . . .	18
6.2.4. Implementación del algoritmo 'balltree' . . . . .	18
6.2.5. Cálculo de los parámetros . . . . .	19
6.3. Estandarización . . . . .	21
6.4. Exploración de las distribuciones en el modelo de Machine Learning .	22
6.5. Modelos de Aprendizaje Automático para la Clasificación de Binarias Reales y Alineamientos Aleatorios . . . . .	24
6.6. Relevancia de la Variable 'g<18 local source den' en la Capacidad Predictiva del Modelo . . . . .	25
6.7. Análisis de la Complejidad Computacional de Algoritmos para la Búsqueda de Estrellas Vecinas y Selección de Candidatos Binarios . . . . .	27
6.8. Aplicación y Corrección del Método KDE para el Análisis de Alineaciones en Datos Binarios: Un Enfoque Dual de Correlación Pearson-Spearman . . . . .	28
<b>7. Resultados y análisis</b>	<b>30</b>
7.1. <b>Aplicación de KDE y Regresión Logística en el Análisis de estrellas binarias: Un Estudio Comparativo</b> . . . . .	30
7.2. Regresión Logística . . . . .	35
7.3. Validación y Precisión de la Regresión Logística en la Detección de Alineaciones Estelares: Un Estudio Basado en Simulaciones . . . . .	41

7.4. Mejora en la Eficiencia y Precisión: Aplicación de K-Neighbors Regressor para la Detección de Estrellas Binarias . . . . .	43
<b>8. Conclusiones</b>	<b>45</b>
8.1. Conclusiones generales . . . . .	45
8.2. Trabajo futuro . . . . .	46
8.3. Código y Uso . . . . .	47
Referencias . . . . .	48
<b>A. Selección de la metología</b>	<b>51</b>
<b>B. Complejidad</b>	<b>53</b>
<b>C. Regresión Logística</b>	<b>57</b>
C.1. Equilibrio entre confianza y volumen de datos . . . . .	59
C.2. Binary Cross-Entropy . . . . .	61
C.3. Optimizadores . . . . .	63
<b>D. Cortes Astrofísicos</b>	<b>64</b>
D.1. Candidatos binarios . . . . .	64
D.2. Búsqueda de vecinos cercanos . . . . .	65
D.3. Búsqueda de vecinos sobre candidatos binarios . . . . .	65

## Capítulo 1

# Introducción

En el fascinante campo de la astronomía, la generación de datos está en constante expansión, en gran parte gracias a los avances tecnológicos que han remodelado la forma en que se realiza la investigación astronómica. En particular, herramientas como la teledetección espacial y los satélites, como el GAIA ([Gaia Collaboration y cols., 2016](#)), han facilitado la recolección de información detallada y precisa sobre una gran cantidad de estrellas. Esto ha impulsado la necesidad de desarrollar nuevas estrategias y métodos para analizar y comprender los datos derivados de estas innovaciones tecnológicas.

El lanzamiento del satélite GAIA al espacio ha sido crucial para desentrañar los misterios de nuestra galaxia, la Vía Láctea. Con su precisión sin precedentes, GAIA ha mapeado la posición, movimiento y brillo de más de mil millones de estrellas. Este tesoro de información ha permitido a los científicos comprender mejor la forma y composición de la Vía Láctea, revelando su estructura en espiral, la distribución de estrellas y la historia evolutiva de nuestra galaxia. Estos descubrimientos revolucionarios no solo nos permiten explorar nuestros orígenes cósmicos, sino que también abren nuevas puertas para comprender la formación y evolución de otras galaxias en el universo. GAIA es una ventana al pasado y una guía para el futuro en nuestra búsqueda en comprender el universo ([Isaeva, 2012](#)).

En este panorama en constante evolución, el estudio de las estrellas binarias se ha convertido en una tarea esencial para impulsar nuestro conocimiento astronómico. Las estrellas binarias son sistemas de dos estrellas que orbitan en torno a un centro de masa común debido a su mutua atracción gravitatoria. Estas abarcan una amplia gama de fenómenos astrofísicos, desde la evolución estelar hasta la relatividad general, y desempeñan un papel crucial en la formación de elementos pesados ([Abate, C., Pols, O. R., Karakas, A. I., y Izzard, R. G., 2015](#)), fundamentales para la existencia de vida tal como la conocemos. Sorprendentemente, la galaxia está repleta de estos sistemas, Dany Vanbeveren ([Vanbeveren, 2001](#)) sugiere que hasta la mitad de todas las estrellas podrían existir en pares. ¿Por qué hay tantas? Las complejas interacciones gravitacionales en los cúmulos estelares durante la formación de las estrellas pueden ser la clave, impulsando la formación de estas parejas estelares en un fenómeno que continuamos desentrañando.

No obstante, a pesar de su importancia, el estudio de las estrellas binarias presenta ciertos desafíos. El más destacado es la dificultad inherente a su identificación. En nuestra percepción bidimensional del cielo, las estrellas pueden parecer estar cerca unas de otras cuando, de hecho, pueden estar a distancias muy diferentes. Este

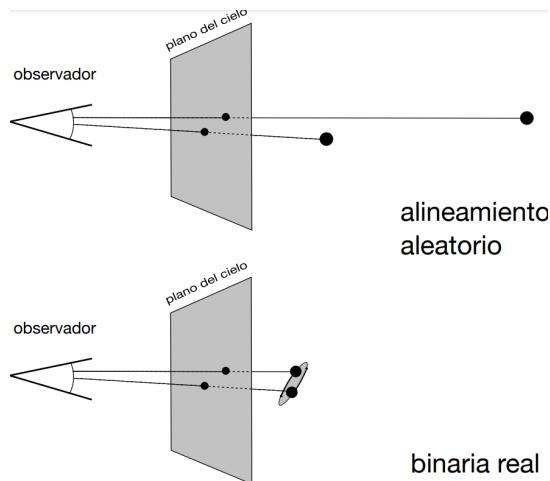


FIGURA 1.1: Diagrama explicativo donde se diferencia un par binario real de un alineamiento. Las “binarias falsas” (arriba) son simplemente alineamientos aleatorios de estrellas que parecen estar cerca debido a nuestra percepción bidimensional del cielo, pero en realidad están a distancias muy diferentes. Por otro lado, Los sistemas binarios reales (abajo) consisten en dos estrellas que orbitan un centro de masa común y están físicamente relacionadas. Diferenciar entre estas dos situaciones es uno de los desafíos principales en el estudio de las estrellas binarias.

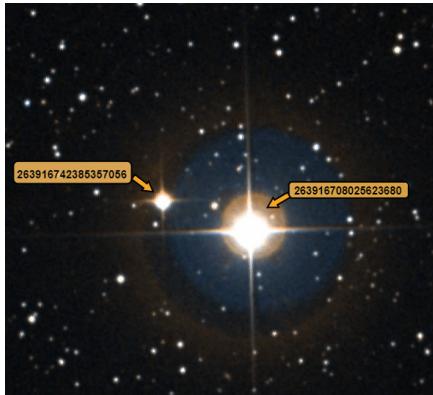
fenómeno es lo que se conoce como un alineamiento estelar aleatorio. Aunque estas estrellas parecen estar cerca en el cielo, no están físicamente asociadas como las estrellas binarias.

Aquí es donde conceptos como el paralaje y el movimiento propio son vitales. El paralaje, el cambio aparente en la posición de una estrella cuando se observa desde diferentes puntos de vista, nos permite medir distancias a estrellas cercanas. Mientras tanto, el movimiento propio, el desplazamiento real de una estrella a través del cielo debido a su movimiento físico en el espacio, nos ayuda a entender su trayectoria. Juntos, estos métodos nos permiten tener una visión más precisa de la estructura tridimensional del universo y de las distribuciones espaciales de las estrellas.

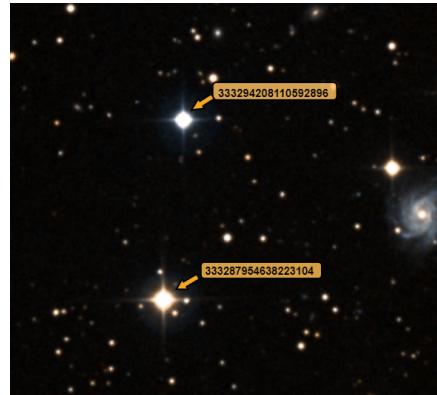
Por tanto, en nuestro estudio nos centramos en la creación de un método sofisticado para la detección y clasificación de estrellas binarias. Nuestro objetivo es desarrollar un catálogo exhaustivo y fiable, que contribuya a profundizar nuestro entendimiento en este campo de la astronomía. Esperamos que este algoritmo sea capaz de manejar grandes volúmenes de datos y que pueda adaptarse y crecer a medida que los catálogos astronómicos se expandan.

Una de las principales innovaciones de nuestro enfoque es la incorporación de un modelo de aprendizaje automático en este algoritmo. Esta integración promete incrementar la eficiencia y precisión en la clasificación de las estrellas binarias. Este modelo será capaz de abordar uno de los desafíos más grandes en el estudio de las estrellas binarias: la capacidad de diferenciar entre verdaderas estrellas binarias y simples alineamientos aleatorios de estrellas. De este modo, podremos minimizar los errores en nuestra interpretación de los datos, mejorando así la calidad y fiabilidad de nuestro trabajo.

La principal aportación de esta herramienta es la automatización y aceleración del



(A) Esta imagen muestra un sistema de estrellas binarias reales.



(B) La imagen ilustra un alineamiento aleatorio de estrellas.

FIGURA 1.2: Las dos imágenes anteriores cuentan con una etiqueta por estrella que corresponde al Source\_id de la estrella. Ambas imágenes contrastan la diferencia entre un sistema de estrellas binarias reales y una “binaria falsa” o alineamiento aleatorio de estrellas.

procesamiento y análisis de grandes volúmenes de datos, lo que en métodos tradicionales requeriría un tiempo significativamente mayor. Su eficiencia permite una exploración más rápida de las estrellas binarias, impulsando nuestra comprensión sobre estas.

Este nuevo método generará un catálogo más preciso de estrellas binarias reales, que será fundamental para avanzar en varios aspectos del estudio astronómico. Nos permitirá calibrar con mayor precisión las estimaciones de las edades estelares en diversas regiones del espacio y establecer relaciones más exactas entre las masas iniciales y finales de las estrellas. Esto último es esencial para profundizar en nuestra comprensión de la evolución estelar y la formación de objetos compactos, como las enanas blancas. Además, a través del efecto de corrimiento al rojo gravitacional en los sistemas binarios, podremos calcular con mayor precisión las masas de las enanas blancas, y obtener información más detallada sobre las etapas previas a su formación.

Por otro lado, este método permitirá la investigación de las propiedades y la distribución de las masas estelares en el universo, proporcionando información valiosa sobre la formación y evolución estelar. En casos favorables, las estrellas binarias reales nos proporcionarán datos para determinar los radios de las estrellas involucradas en los sistemas binarios, lo cual es crucial para mejorar la comprensión de la estructura y las propiedades físicas de las estrellas y para validar y perfeccionar los modelos teóricos existentes. Así, este trabajo tiene el potencial de desafiar y expandir nuestro actual conocimiento del universo.

## Capítulo 2

# Definición del problema u oportunidad

La astronomía, como campo científico, se encuentra frente a dos problemas intrincados. En primer lugar, nos encontramos con una cantidad abrumadora de datos astronómicos. Los catálogos astronómicos, como GAIA ([Gaia Collaboration y cols., 2016](#)), almacenan información sobre aproximadamente 1.8 mil millones de estrellas, equivalente a unos 10 terabytes de datos. Este vasto mar de información presenta un desafío significativo cuando se trata de analizar y entender la información que contiene.

El segundo problema es la identificación precisa de las estrellas binarias reales. En astronomía, a menudo nos encontramos con alineaciones aleatorias de estrellas que pueden confundirse fácilmente con estrellas binarias. La complejidad computacional de emparejar cada estrella con todas las demás en estos catálogos masivos es enorme. Se convierte en una tarea ardua diferenciar entre estrellas binarias genuinas y alineaciones aleatorias de estrellas, lo cual es esencial para obtener un entendimiento profundo de estos sistemas.

La existencia de estos problemas está profundamente arraigada en la propia naturaleza de la astronomía y sus desafíos inherentes. Principalmente, la inmensidad de los datos astronómicos y la perspectiva limitada que tenemos del universo debido a nuestra posición en el espacio. Como lo señala El-Badry ([El-Badry, Rix, y Heintz, 2021](#)), las alineaciones aleatorias y las estrellas binarias auténticas pueden encontrarse en regiones distintas, pero superpuestas, del espacio de parámetros, lo que añade un nivel adicional de complejidad a esta tarea.

Este estudio propone abordar estos desafíos a través de la innovación en los métodos de análisis y procesamiento de datos. Tradicionalmente, se han utilizado técnicas como el Estimador de Densidad Kernel (KDE) para lidiar con estos problemas. Sin embargo, a medida que aumenta la cantidad de datos y la necesidad de procesamiento eficiente, se requieren enfoques más sofisticados y robustos. La eficiencia del KDE se ve disminuida cuando se aplica a grandes conjuntos de datos en espacios de alta dimensionalidad. Esta limitación surge debido a que el algoritmo del KDE, en su versión más sencilla, necesita evaluar el kernel para cada punto en el conjunto de datos de referencia en relación con cada consulta en el conjunto de datos de entrada, lo que lleva a una complejidad temporal cuadrática ([Backurs, Indyk, y Wagner, 2019](#)). Con conjuntos de datos masivos, conteniendo millones de puntos en múltiples dimensiones, la cantidad total de evaluaciones de kernel necesarias incrementa drásticamente, resultando en un tiempo de ejecución prohibitivo. En este contexto,

la incorporación de algoritmos de machine learning se presenta como una estrategia prometedora para mitigar estos desafíos.

Este algoritmo de aprendizaje automático tiene como objetivo mejorar el índice de recuperación (recovery rate) de estrellas binarias reales en comparación con las alineaciones aleatorias, lo que sería un salto significativo en la precisión y eficiencia de la identificación de estrellas binarias. Específicamente, el algoritmo se diseñará para clasificar estrellas binarias de las alineaciones aleatorias utilizando los mismos parámetros mencionados por El-Badry ([El-Badry y cols., 2021](#)). A través de este enfoque, el estudio busca no solo mejorar la precisión en la identificación de estrellas binarias, sino también aumentar la eficiencia del proceso. Es decir, se busca no solo obtener resultados más exactos, sino hacerlo de una manera más rápida y eficiente. En este sentido, la eficiencia se mide tanto en términos de la velocidad del proceso de identificación como en la capacidad del algoritmo para manejar y procesar grandes volúmenes de datos. Esta combinación de precisión y eficiencia promete ser un elemento clave en la superación de los desafíos que enfrenta actualmente la astronomía en la identificación de estrellas binarias.

El algoritmo propuesto fue diseñado para satisfacer múltiples criterios o requerimientos. Incluye la clasificación de posibles estrellas binarias, la clasificación de estas candidatas y la generación de un catálogo final de estrellas binarias auténticas. Este catálogo puede contribuir significativamente a nuestra comprensión de la astronomía, ya que permitirá un estudio más detallado de las propiedades de estas estrellas binarias.

Por último, la creación de este catálogo de estrellas binarias reales aportará beneficios significativos a la comunidad científica. Entre otros, se puede usar para la calibración de edades estelares, el establecimiento de relaciones de masa inicial-final, el cálculo de masas de enanas blancas a partir del corrimiento al rojo gravitacional, y para estimar la abundancia de estrellas progenitoras de enanas blancas. Además, también puede ser útil para el estudio de las masas de estrellas y, en ciertos casos favorables, podría permitir determinar los radios de las estrellas involucradas.

## Capítulo 3

# Estado del arte

Con la creciente acumulación de datos astronómicos, la comunidad académica ha reconocido la necesidad de mejorar los métodos de manejo y análisis de dichos datos, especialmente aquellos recolectados por satélites como GAIA a través de consultas ADQL. Este tipo de consulta, que es similar a SQL pero diseñada específicamente para trabajar con bases de datos astronómicas, permite manipulaciones geométricas y temporales esenciales para el estudio astronómico ([Gaia Collaboration y cols., 2016](#)).

La eficiencia en el almacenamiento y acceso a grandes volúmenes de datos constituye uno de los retos primordiales en este campo de investigación. La literatura académica ha discutido exhaustivamente la necesidad de sistemas de almacenamiento escalables y eficientes, resaltando que la habilidad para extraer valor de Big Data se fundamenta en gran parte en la eficiencia de dichos procesos de almacenamiento y gestión de datos ([Barbierato, Gribaudo, y Iacono, 2013](#)). De este modo, la creación de infraestructuras de almacenamiento de datos capaces de manejar la magnitud de los catálogos astronómicos contemporáneos se ha convertido en una prioridad notable ([Riquelme Santos, Ruiz, y Gilbert, 2006](#)).

Los avances en la compilación de catálogos de estrellas binarias han proporcionado conjuntos de datos robustos y variados que reflejan la diversidad y la frecuencia de estos sistemas estelares. Un ejemplo notable es un catálogo basado en GaiaDR2 ([Sapozhnikov, Kovaleva, Malkov, y Sytov, 2020a](#)), que incluye estrellas de movimiento propio común dentro de un radio de 100 pc alrededor del Sol, excluyendo aquellos sistemas de multiplicidad superior a dos. Esta meticulosa selección ha resultado en un catálogo compuesto por 10358 pares de estrellas binarias ([Sapozhnikov, Kovaleva, Malkov, y Sytov, 2020b](#)). De manera similar, el catálogo del Grupo Binario Kepler hasta noviembre de 2016 incluyó 3541 objetivos binarios. De estos, 1601 fueron también observados por LAMOST DR4 ([Zhang, Qian, Wu, y Zhou, 2019](#)) y se obtuvieron exitosamente parámetros atmosféricos para 1379 de ellos. Entre estos, se derivaron propiedades físicas exitosas para 1320 estrellas, dando lugar a un subconjunto sustancial del catálogo ([Zhang, Qian, Wu, y Zhou, 2020](#)). En contraste, la Lista de Identificación de Binarias (ILB) es un catálogo estelar creado para facilitar las referencias cruzadas entre diferentes catálogos de estrellas binarias. A partir de 2015, comprende designaciones para aproximadamente 120000 sistemas dobles/múltiples ([Malkov, Karchevsky, Kaygorodov, y Kovaleva, 2016](#)). Estos catálogos reflejan la continua dedicación y esfuerzo por comprender los sistemas binarios y su prevalencia en nuestra galaxia.

Un aspecto crítico que ha sido bien documentado es el sesgo en los datos astronómicos, y en particular la dificultad de distinguir entre estrellas binarias verdaderas y

alineaciones aleatorias. Este problema se intensifica cuando las estrellas están muy juntas, lo que puede provocar errores en la identificación y clasificación de los objetos (Rybicki y cols., 2021).

En respuesta a estos desafíos, la implementación de algoritmos de aprendizaje automático ha emergido como un enfoque prominente en la investigación. Estos métodos no solo están ayudando a mejorar la eficiencia del análisis de datos, sino que también están permitiendo a los investigadores manejar la complejidad inherente a los datos astronómicos y obtener insights que antes eran inaccesibles.

En este contexto, el trabajo de K. El-Badry y Sébastien Lépine se destaca por su enfoque en la construcción de extensos catálogos de estrellas binarias espacialmente resueltas utilizando los datos de Gaia eDR3 y el catálogo de Movimiento Propio Lépine-Shara respectivamente. Ambos abordan el problema de distinguir entre estrellas binarias verdaderas y alineaciones aleatorias, y cada uno ofrece un enfoque único para tratar con el reto académico inherente al análisis de estos sistemas (El-Badry y cols., 2021; Lépine y Bongiorno, 2007).

Los trabajos de Andrews y Hartman, por otro lado, han implementado la técnica de Estimación de Densidad Kernel (KDE) para estimar la densidad local en el espacio de parámetros de los candidatos a binarios y las alineaciones casuales. Este método de suavizado de datos ha demostrado ser útil en el análisis de datos multidimensionales en el estudio de estrellas binarias (Andrews, Chanamé, y Agüeros, 2017; Hartman y Lépine, 2020).

Mientras el manejo de los datos astronómicos continúa presentando dificultades científicas significativas, la comunidad académica está implementando y desarrollando nuevas técnicas y métodos para enfrentar estos retos. La Estimación de Densidad Kernel, las técnicas bayesianas, y los algoritmos de aprendizaje automático están siendo utilizados con éxito para filtrar estrellas binarias reales de alineaciones aleatorias, y se anticipa que estos enfoques seguirán evolucionando a medida que se recolecten y analicen más datos astronómicos.

Como se ha podido observar, la tarea de clasificar estrellas binarias reales ha sido abordada por diversos investigadores desde una variedad de enfoques. El más relevante para este trabajo es el de El-Badry, quien desarrolló un catálogo de estrellas binarias a partir de los cortes astrofísicos previamente establecidos por Lépine. Este método implementó el algoritmo KDE para estructurar un espacio paramétrico compuesto por puntos que representan las características de los candidatos binarios. Para cada par binario, El-Badry calculó un valor que se puede interpretar como una "probabilidad" de que el par constituya una estrella binaria, aunque este valor no es una probabilidad en el sentido estricto ya que no se limita a ser menor o igual a 1.

El resultado de este proceso fue un catálogo que contiene aproximadamente un millón de candidatos binarios. Aunque este algoritmo es efectivo para la recopilación de candidatos binarios, su desempeño disminuye notablemente cuando se trata de discriminar entre estrellas binarias y alineaciones aleatorias. Por ejemplo, puede requerir hasta 17 horas para clasificar 1,8 millones de candidatos, lo cual se convierte en un inconveniente importante considerando el crecimiento exponencial de los datos disponibles.

Dado este desafío, se ha llevado a cabo un análisis exhaustivo de las variables involucradas y su espacio paramétrico. El objetivo de este análisis es prevenir que el

algoritmo se vuelva obsoleto en el futuro a medida que la cantidad de datos continúa aumentando. Este problema en particular ha impulsado la necesidad de mejorar la eficiencia y la precisión de la clasificación de estrellas binarias.

## Capítulo 4

# Hipótesis y objetivos

**Hipótesis:** Existen algoritmos más precisos y eficientes que la Estimación de Densidad Kernel (KDE) para mejorar la clasificación de las estrellas binarias utilizando los datos del proyecto GAIA eDR3.

**Objetivo General:** Probar y comparar los algoritmos existentes en términos de precisión y eficiencia en la clasificación de estrellas binarias, con el fin de determinar si pueden superar a KDE en este aspecto, utilizando los datos del proyecto GAIA eDR3.

### Objetivos específicos:

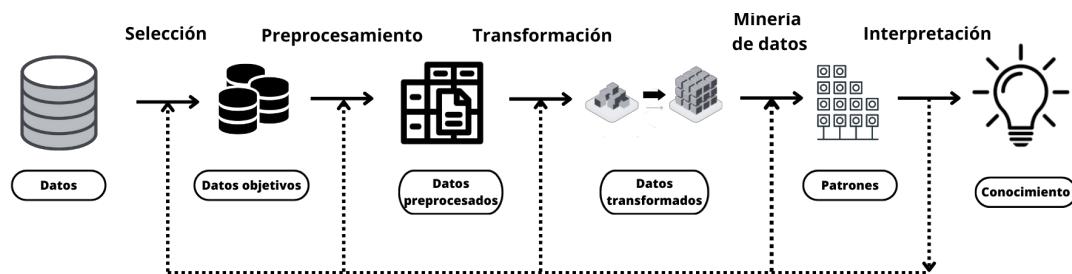
1. Adquirir un entendimiento sólido de los fundamentos de la astronometría y la astronomía, incluyendo los conceptos clave, las técnicas básicas y las aplicaciones prácticas de estas disciplinas.
2. Aprender a utilizar ADQL para interpretar datos astronómicos del proyecto GAIA y entender las herramientas necesarias para generar un catálogo de estrellas binarias.
3. Aplicar correctamente el algoritmo de clasificación y filtración de estrellas binarias, obteniendo así candidatos binarios que posteriormente llevaremos a un rango dinámico.
4. Implementar técnicas de tratamiento de grandes cantidades de datos, como el sobremuestreo, reducción de *dataset* y análisis de correlaciones, para mejorar la eficiencia y precisión de los modelos.
5. Crear varios modelos de clasificación que permitan predecir si una estrella es binaria o no, utilizando características y variables obtenidas de las estrellas, como la distancia entre ellas. Utilizando como etiqueta un catálogo de estrellas binarias anterior.
6. Evaluar el rendimiento de diferentes algoritmos de clasificación, incluyendo Naive Bayes, LDA, QDA, SVM y redes neuronales, para determinar cuál ofrece el mejor rendimiento en la clasificación de estrellas binarias utilizando métricas de comparación como la *precisión*, *recall*, *f1-score* y *accuracy*.
7. Realizar un análisis exploratorio de los datos (*EDA*) para determinar la importancia de los parámetros mencionados.
8. Evaluar la relevancia de variables adicionales sugeridas en la investigación para optimizar la clasificación de estrellas binarias, y por otro lado, considerar la eliminación de variables si los resultados obtenidos así lo indican.

9. Desarrollar y compartir una función pública en una biblioteca de código abierto, que permita a otros investigadores y entusiastas de la astronomía obtener un catálogo de estrellas binarias utilizando los modelos y técnicas desarrollados durante esta investigación.

## Capítulo 5

# Metodología

En este trabajo se ha implementado la metodología del Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD), un enfoque estructurado en la ciencia de datos para extraer conocimientos significativos de grandes cantidades de información. A grandes rasgos, la metodología KDD abarca las siguientes etapas: selección de datos, preprocesamiento, transformación, minería de datos e interpretación y evaluación. Para más detalles sobre la elección de esta metodología, remitirse al Apéndice A. En los siguientes apartados, detallaremos cada uno de estos pasos.



**Figura 5.1:** Diagrama de KDD que ilustra el proceso de descubrimiento de conocimiento en bases de datos. El diagrama muestra las etapas clave del proceso, incluyendo la selección y preparación de datos, la aplicación de técnicas de minería de datos, la evaluación de los resultados y la interpretación del conocimiento descubierto. Cada etapa representa una fase crucial en el ciclo de KDD, que busca transformar datos en información y conocimiento útiles.

1. **Datos:** En el presente estudio, hemos realizado un exhaustivo análisis de los datos proporcionados por el satélite GAIA. Este repositorio de información astronómica, de relevancia incomparable, abarca aproximadamente 1.8 mil millones de estrellas, proporcionando un vasto campo de estudio para nuestros objetivos de investigación.
2. **Selección de datos:** Se obtienen los datos del proyecto GAIA eDR3 mediante consultas ADQL, lo que resulta en una lista de aproximadamente 64.000.000 de estrellas. Estos datos incluyen características de las estrellas y variables que pueden ser obtenidas entre ellas, como la distancia.
3. **Preprocesamiento de datos:** Se llevó a cabo un Análisis Exploratorio de Datos (EDA) con el objetivo de obtener una comprensión detallada de la estructura y características inherentes de los datos. Es importante subrayar que no fue necesaria la aplicación de técnicas de preprocesamiento convencionales, tales como el manejo de valores faltantes, la gestión de variables categóricas o la normalización de datos. Esta particularidad se debe a que los datos proporcionados

por el satélite GAIA se encuentran sujetos a ciertas incertidumbres, las cuales hemos tenido en cuenta y hemos mitigado en la fase de extracción de los datos, mediante consultas ADQL específicas que hemos detallado previamente.

4. **Transformación de datos:** En esta etapa del análisis, procedemos a identificar una lista de candidatos potenciales a ser estrellas binarias a partir del conjunto de estrellas inicial. Cabe señalar que se excluyen aquellas estrellas originadas en regiones particularmente densas de la galaxia para evitar sesgos en la selección. Posteriormente, procedemos a reescalar los parámetros de interés de cada estrella candidata, garantizando que todos los datos estén contenidos dentro de un rango dinámico uniforme.
5. **Identificación de Patrones:** Posterior a la transformación de los datos, llevamos a cabo un proceso de comparación. Contrastamos la población de estrellas binarias, sobre las cuales tenemos un alto grado de certeza de su clasificación, con la población de estrellas con alineaciones aleatorias. A través de este análisis, logramos discernir ciertos patrones emergentes en diversas variables, que caracterizan de manera distintiva a cada grupo..
6. **Creación de modelos de clasificación:** Se implementan varios modelos de clasificación, incluyendo Naive Bayes, LDA, QDA, SVM y redes neuronales, con el objetivo de predecir si una estrella es binaria o no. Cada modelo se entrena utilizando los datos de entrenamiento y se ajusta a los parámetros óptimos utilizando técnicas de validación cruzada.
7. **Evaluación de modelos:** Se evalúa el rendimiento de los modelos utilizando métricas de comparación, como la precisión, recall, f1-score y accuracy. Esto permite determinar qué modelos ofrecen un mejor desempeño en la clasificación de estrellas binarias.
8. **Análisis de resultados y comparación de variables:** Se realiza un análisis de los resultados obtenidos para determinar la importancia de las variables en cada modelo y si se pueden identificar patrones consistentes en la clasificación de estrellas binarias. Además, se compara la importancia de las variables con los resultados obtenidos en estudios anteriores para identificar posibles mejoras en la clasificación.

La elección de esta metodología se basa en una revisión bibliográfica de metodologías utilizadas en trabajos similares. Se han seleccionado modelos de clasificación ampliamente utilizados y estudiados en el campo del aprendizaje automático. Naive Bayes, LDA, QDA, SVM y redes neuronales son modelos conocidos por su capacidad para realizar clasificaciones precisas en diferentes dominios de aplicación.

La metodología propuesta sigue una secuencia lógica y estructurada, que permite a otros investigadores replicar el trabajo y obtener resultados comparables. Se presta especial atención al preprocesamiento de datos, la selección de modelos, la evaluación del rendimiento y el análisis de los resultados obtenidos. Además, se realiza un análisis bibliográfico exhaustivo para justificar la elección de cada modelo y asegurar su relevancia en el contexto de la clasificación de estrellas binarias.

Es importante destacar que todo el código desarrollado en este proyecto se pretende dejar disponible en una biblioteca pública, con el objetivo de promover la transparencia, la reproducibilidad y el avance de la investigación en el campo de la clasificación de estrellas binarias. Al dejar el código público, se brinda a otros investigadores

la oportunidad de acceder, utilizar y validar el enfoque propuesto. Esto es con el fin de:

- **Reproducibilidad:** Al proporcionar el código utilizado en el proyecto, otros investigadores tienen la capacidad de replicar los experimentos y validar los resultados obtenidos. Esto fomenta la confianza en los hallazgos y permite una mejor comprensión de los métodos utilizados.
- **Colaboración y mejora continua:** Al compartir el código, se crea una oportunidad para que otros investigadores realicen mejoras, proporcionen retroalimentación y colaboren en el desarrollo de nuevas técnicas y enfoques. Esto promueve la colaboración y acelera el progreso científico en el campo.
- **Transparencia y ética:** Al hacer que el código sea público, se promueve la transparencia en la investigación científica. Otros investigadores pueden revisar y evaluar el enfoque utilizado.
- **Aprendizaje y educación:** Al compartir el código en una biblioteca pública, se proporciona un recurso educativo valioso para la comunidad científica y para aquellos interesados en el tema. Otros investigadores pueden estudiar el código, comprender las técnicas implementadas y aprender de ellas, lo que fomenta el intercambio de conocimientos y el avance colectivo.
- **Reutilización y ahorro de tiempo:** Al tener acceso al código desarrollado en este proyecto, otros investigadores pueden reutilizar y adaptar partes del mismo para sus propios estudios. Esto ahorra tiempo y esfuerzo, ya que no es necesario empezar desde cero, y permite enfocarse en aspectos específicos de investigación en lugar de la implementación básica de algoritmos.

Al seguir esta metodología, se espera obtener resultados confiables y significativos, que contribuyan al avance del conocimiento en el campo de la clasificación de estrellas binarias y que puedan ser validados y replicados por otros investigadores.

## Capítulo 6

# Propuesta

Proponemos explorar una gama de técnicas de aprendizaje automático, con el objetivo de mejorar la clasificación de las estrellas binarias y los alineamientos aleatorios. Esto lo realizaremos comparando los modelos en términos de precisión y eficiencia en la clasificación de estrellas binarias, buscando superar el rendimiento de la Estimación de Densidad Kernel (KDE) existente. Estos modelos se adaptarán en el marco del proyecto GAIA eDR3, una de las fuentes de datos astronómicos más extensas y precisas disponibles actualmente. A continuación se detallará varios aspectos claves de este análisis.

### 6.1. Descripción y Preprocesamiento de Variables Utilizadas de GAIA

En la presente sección, se detallan las variables seleccionadas de la base de datos de GAIA para el estudio de la clasificación de sistemas binarios. A continuación, se describe cada variable, incluyendo su tipo, unidad y una breve descripción de lo que representa. También se presentan los criterios de filtrado aplicados a estas variables para optimizar la precisión y eficiencia del modelo de aprendizaje automático utilizado posteriormente en nuestro análisis.

Nombre	Tipo	Unidad	Descripción
source_id	long	-	Identificador de fuente único (único dentro de una versión específica de datos)
ra	double	deg	Ascensión recta
dec	double	deg	Declinación
parallax	double	mas	Paralaje
parallax_error	float	mas	Error estándar de paralaje
pmra	double	mas $\cdot yr^{-1}$	Movimiento propio en la dirección de la ascensión recta
pmdec	double	mas $\cdot yr^{-1}$	Movimiento propio en la dirección de la declinación
pmra_error	float	mas $\cdot yr^{-1}$	Error estándar del movimiento propio en la dirección de la ascensión recta
pmdec_error	float	mas $\cdot yr^{-1}$	Error estándar del movimiento propio en la dirección de la declinación
phot_g_mean_mag	float	mag	Magnitud media en la banda G

### 6.1.1. Descripción de las variables

La **Ascensión Recta** se asemeja a la longitud en la Tierra y es medida en horas, minutos y segundos. Es una medida angular que nos dice cuánto tiempo ha pasado desde un punto de referencia llamado Punto Vernal, similar al meridiano de Greenwich en la Tierra. La Ascensión Recta nos permite localizar un objeto en el cielo a lo largo del plano ecuatorial, que es la proyección del ecuador terrestre en el espacio.

La **Declinación**, por otro lado, es similar a la latitud en la Tierra y también se mide en grados. Es una medida angular que indica qué tan lejos está un objeto del ecuador celeste. La Declinación es positiva para objetos ubicados en el hemisferio norte celeste y negativa para los ubicados en el hemisferio sur celeste. Junto con la Ascensión Recta, la Declinación nos brinda una ubicación precisa de un objeto en el cielo.

El **paralaje** es un concepto que nos ayuda a medir distancias en el espacio. Imagina que tienes un objeto cerca de ti y cierras un ojo mientras lo miras. Si luego abres el otro ojo y cierras el primero, notarás que el objeto parece moverse con respecto a los objetos más lejanos. Ese cambio aparente de posición se llama paralaje.

En astronomía, los científicos usan este principio para medir distancias a estrellas cercanas. Usan la posición de la estrella en el cielo cuando están en lados opuestos de la órbita de la Tierra alrededor del Sol. Al comparar las posiciones y el ángulo del paralaje, pueden calcular qué tan lejos está la estrella.

El **error estándar de paralaje** ocurre debido a limitaciones técnicas y factores como la precisión de las mediciones y las variaciones en las condiciones atmosféricas. Cuanto más lejos esté un objeto en el espacio, menos paralaje tendrá y más difícil será medirlo con precisión. Por lo tanto, el error de paralaje es la incertidumbre asociada con la medición de distancias utilizando esta técnica.

El **movimiento propio en la dirección de la ascensión recta** indica cómo una estrella se está moviendo horizontalmente en relación con las estrellas de fondo en la misma línea de longitud celeste. Si una estrella tiene un movimiento propio en ascensión recta, significa que está desplazándose hacia el este o hacia el oeste a medida que pasa el tiempo.

Por otro lado, el **movimiento propio en la dirección de la declinación** describe cómo una estrella se está moviendo verticalmente en relación con las estrellas de fondo en la misma línea de latitud celeste. Si una estrella tiene un movimiento propio en declinación, significa que está desplazándose hacia el norte o hacia el sur a medida que transcurre el tiempo.

El **error estándar del movimiento propio**, tanto en la ascensión recta, como en la declinación, nos proporciona información sobre la confiabilidad y precisión de las mediciones del cambio gradual de posición de un objeto celeste en la dirección de la ascensión recta y declinación a lo largo del tiempo. Cuanto menor sea el valor del error estándar, mayor será la precisión de las mediciones y, por lo tanto, mayor confianza tendremos en los resultados. Un error estándar más alto indica una mayor dispersión o incertidumbre en las mediciones, lo que implica una menor precisión en la determinación del movimiento propio del objeto en cuestión.

La **magnitud media en la banda G** es una medida utilizada en astronomía para describir el brillo de un objeto celeste, como una estrella o una galaxia, en una región específica del espectro electromagnético conocida como banda G. Esencialmente, la

magnitud media en la banda G nos da una idea de qué tan brillante aparece un objeto en el rango de longitudes de onda que abarca esa banda.

El sistema de magnitudes astronómicas es inverso y logarítmico. La inversión se refiere a que a medida que la magnitud aumenta, el brillo disminuye. Por ejemplo, una estrella de magnitud 1 es más brillante que una de magnitud 2, y una de magnitud 2 es más brillante que una de magnitud 3, y así sucesivamente. Cada incremento de 1 en la magnitud implica una disminución de brillo aproximada de 2.5 veces.

Por otro lado, el aspecto logarítmico radica en cómo se relaciona la magnitud con el brillo. En lugar de seguir una escala lineal, la escala de magnitudes utiliza una escala logarítmica. Esto significa que un cambio de 1 en la magnitud corresponde a un cambio de brillo en una cantidad proporcional en una escala logarítmica. Por ejemplo, una diferencia de 5 magnitudes implica una diferencia de brillo de 100 veces.

### 6.1.2. Preprocesamiento

Antes de aplicar el algoritmo, se implementan ciertos filtros al conjunto de datos, estos provienen de la consulta realizada a GAIA mediante ADQL:

- Parallax >1
- Parallax\_over\_error >5
- Parallax\_error <2
- Phot\_g\_mean\_mag no es nulo

La consulta ADQL (Astronomical Data Query Language) tiene como objetivo reducir la contaminación de alineaciones fortuitas y limitar la muestra de pares de objetos astronómicos a aquellos con astrometría moderadamente precisa.

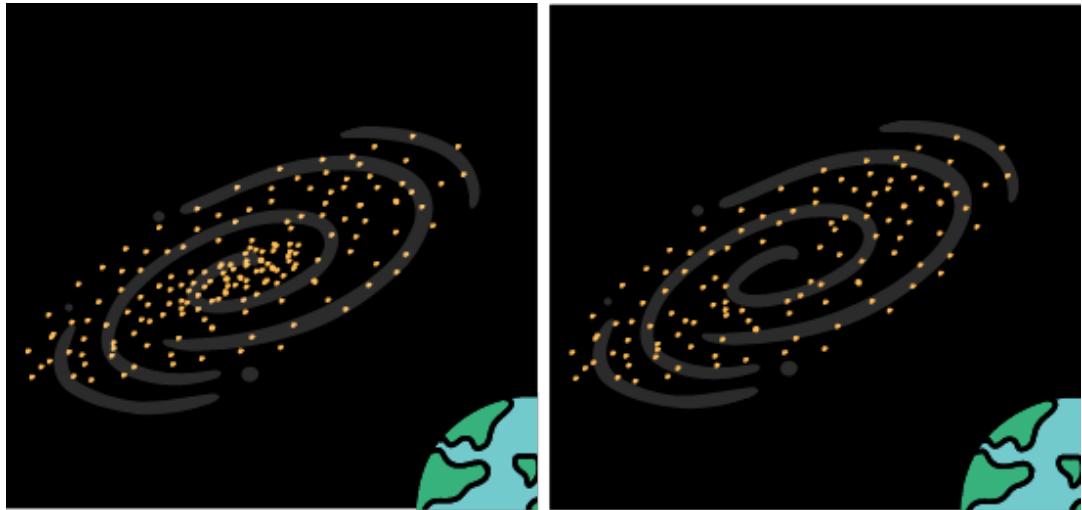
En primer lugar, se establecen criterios de selección basados en mediciones de paralaje. Se eligen pares en los que ambos componentes tienen paralajes mayores a 1 milisegundo de arco (mas). Esto corresponde a un límite de distancia nominal de 1 kilopársec (kpc), aunque debido a errores de paralaje, esta distancia está borrosa en realidad.

Además, se aplican condiciones sobre la precisión del paralaje. Se seleccionan pares en los que la incertidumbre fraccional de la paralaje sea menor al 20 por ciento y la incertidumbre absoluta de paralaje sea menor a 2 mas.

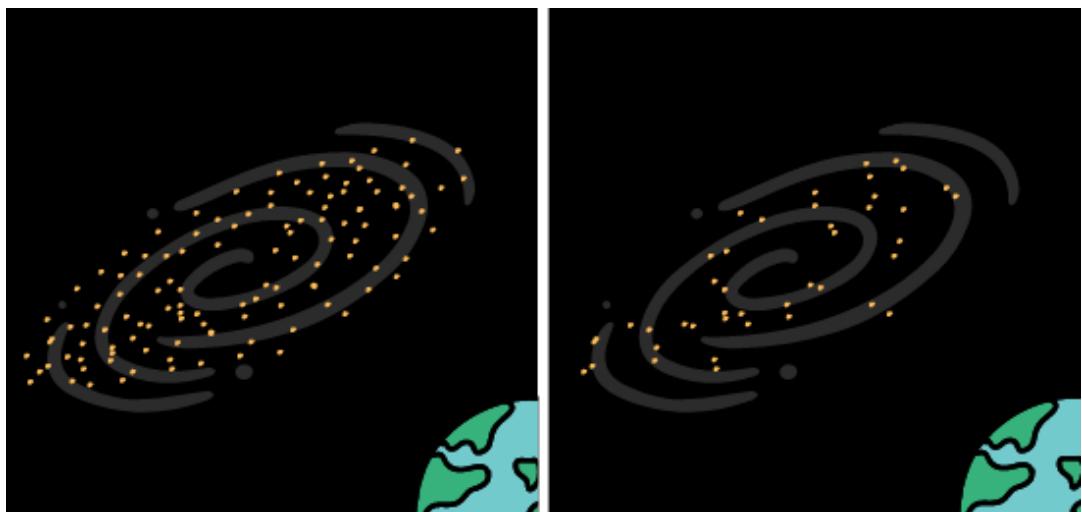
Por último, se verifica que las magnitudes en la banda G (G-band) estén disponibles para los objetos seleccionados. La banda G es una región específica del espectro electromagnético utilizada en el estudio de las estrellas.

Es importante mencionar que en la consulta no se solicita explícitamente el paralaje sobre el error. Esto se debe a que dicho valor puede ser calculado utilizando la siguiente fórmula:

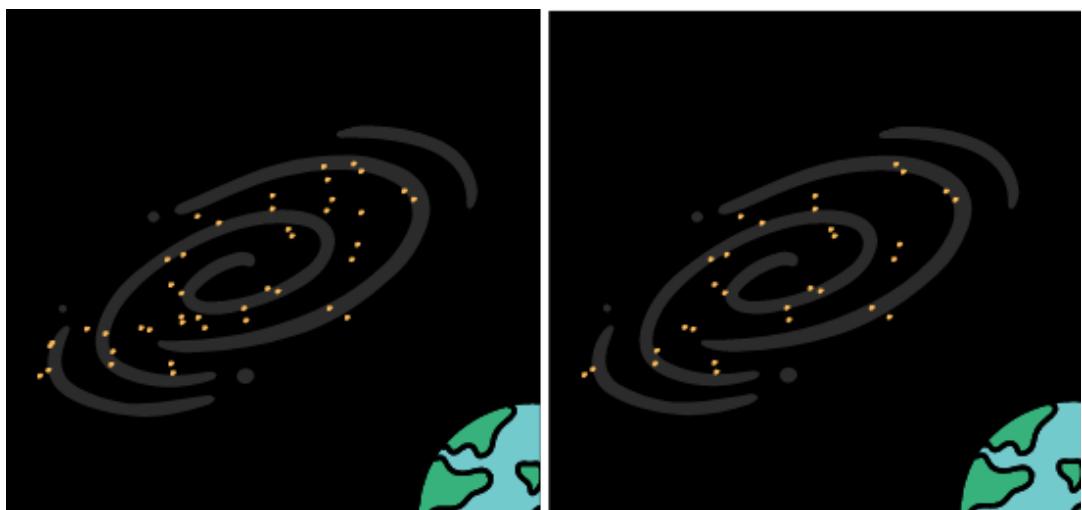
$$\text{parallax over error} = \frac{\text{parallax}}{\text{parallax\_error}}$$



**Figura 6.1:** Búsqueda de vecinos cercanos: Filtro de consistencia en parámetros de astrometría.



**Figura 6.2:** Selección de candidatos binarios: Identificación basada en criterios astronómicos y búsqueda de vecinos cercanos.



**Figura 6.3:** Búsqueda de vecinos sobre candidatos binarios: Filtrado de sistemas complejos.

## 6.2. Algoritmo de filtración de candidatos binarios

En esta sección, presentamos el algoritmo de filtración de candidatos binarios, diseñado para abordar el desafío de identificar y analizar eficientemente sistemas estelares compuestos por dos estrellas en órbita mutua. Es importante mencionar que un candidato binario es un par de estrellas que son candidatas a ser clasificadas como un par binario real o una alineación.

### 6.2.1. Búsqueda de vecinos cercanos:

Primordialmente se realiza la búsqueda de vecinos más cercanos para cada estrella y retorna el número de vecinos que cumplen ciertos criterios de consistencia en sus parámetros de astrometría. Esto es con el fin de filtrar aquellas estrellas que provengan de zonas mas densas de la galaxia, específicamente no se consideran aquellas estrellas que tengan mas de 30 vecinos en un radio de 5 parsecs en el espacio tridimensional. En la figura 6.1 se observa el filtrado de estrellas provenientes de regiones densas de la galaxia.

### 6.2.2. Selección de candidatos binarios:

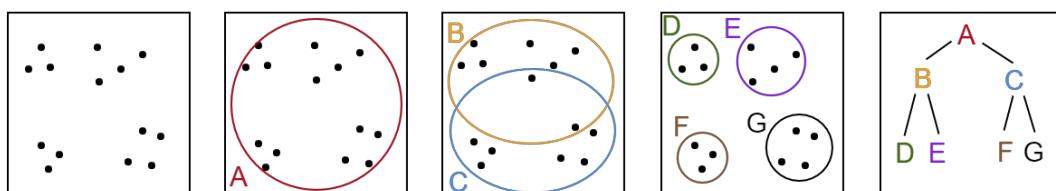
Luego se identifican posibles sistemas binarios en base a ciertos criterios astronómicos especificados en el apéndice D , considerando una separación angular máxima de 1 parsec entre los objetos. En la figura 6.2, se pueden observar los candidatos a estrellas binarias agrupados según criterios astronómicos.

### 6.2.3. Búsqueda de vecinos sobre candidatos binarios:

Luego se filtran aquellos pares binarios que se encuentren muy cerca entre sí, para asegurarse que los candidatos binarios no formen parte de sistemas complejos conformados por más de dos estrellas. En la figura 6.3, se muestra el filtrado de los candidatos binarios que forman parte de sistemas complejos.

### 6.2.4. Implementación del algoritmo 'balltree'

Una potente herramienta algorítmica para optimizar la búsqueda y clasificación de sistemas estelares es el "balltree". Este algoritmo permite construir un árbol binario de partición de espacio, ofreciendo un método de búsqueda eficiente en altas dimensiones. Su aplicación en nuestro estudio se ha dado en tres etapas mencionadas anteriormente.



**Figura 6.4:** Diagrama representativo del algoritmo de BallTree. Inicia con puntos dispersos, que son englobados en un conjunto A. Este se subdivide en B y C, que a su vez se dividen en D, E, F y G respectivamente, creando una estructura jerárquica.

Finalmente, la estructura se representa como un árbol binario.

Una vez en este paso se obtiene una lista de candidatos binarios como la que se puede ver a continuación, en donde la primera estrella corresponde a la más luminosa, mientras que la segunda podría ser su pareja binaria real, o ser un alineamiento.

		ra1	ra2	dec1	dec2	...
1	Par binario					...
2	Par binario					...
3	Par binario					...
...	...	...	...	...	...	...

**Figura 6.5:** Representación de la estructura de los datos hasta esta etapa.

### 6.2.5. Cálculo de los parámetros

A partir de los datos que tenemos, podemos generar los siguientes parámetros:

- **Angular separation**

Para calcular la separación angular está utilizando la fórmula del semiverseno (haversine formula en inglés), que es una alternativa a la ley de los cosenos para la geometría esférica que puede ser más precisa cuando los puntos están muy cerca uno del otro.

La fórmula de la ley del haversine es:

$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\Delta\delta}{2} \right) + \cos(\delta_1) \cos(\delta_2) \sin^2 \left( \frac{\Delta\alpha}{2} \right)} \right)$$

En esta fórmula,  $d$  representa la distancia entre los dos puntos en la esfera,  $\delta_1$  y  $\delta_2$  son las declinaciones de los dos puntos, mientras que  $\Delta\alpha$  y  $\Delta\delta$  corresponden a las diferencias entre los puntos de la ascensión recta y declinación, respectivamente. El término  $r$  se refiere al radio de la esfera. Sin embargo, como nuestro interés radica en el ángulo en vez de una distancia lineal, el término  $2r$  puede obviarse.

La entrada de las coordenadas están en grados, por lo que primero se deben convertir las coordenadas a radianes para utilizarlas en la fórmula. Luego, el resultado (que está en radianes) se traspasa a grados multiplicándolo por  $180/\pi$ . Finalmente, convierte los grados a segundos de arco multiplicando por 3600.

- **Parallax - primary**

No es necesario ningún cálculo para el paralaje, debido a que es un dato que provee el satélite directamente, pero para efectos del algoritmo, se considera solo el paralaje de la estrella más brillante.

- **Parallax difference error**

El error en la diferencia de dos valores es generalmente la raíz cuadrada de la suma de los cuadrados de los errores individuales. Esto se debe a la propagación de errores: si tienes dos cantidades con incertidumbres, la incertidumbre en cualquier función de esas dos cantidades depende de cómo la función cambia con cada cantidad y la incertidumbre en cada cantidad.

$$\text{parallax\_diff\_error} = \sqrt{\text{parallax\_error1}^2 + \text{parallax\_error2}^2} \quad (6.1)$$

- **G<18 local source density**

Variable explicada a profundidad en el párrafo 6.6, la cual es una transformación de la variable  $\Sigma_{18}$ .

- **Tangential velocity - primary**

'pm1' es el movimiento propio total de la estrella, el cual se calculó a partir del movimiento propio en ascensión recta y declinación.

Para convertir el movimiento propio de milisegundos de arco por año a radianes por segundo, se realizan las siguientes operaciones:

1. Se multiplica por 'np.pi/180' para convertir de grados a radianes.
2. Se divide por '60' dos veces para convertir de segundos de arco a grados (una vez para convertir a minutos de arco, y una vez más para convertir a grados).
3. Se divide por '1000' para convertir de milisegundos de arco a segundos de arco.
4. Se divide por '31536000' para convertir de años a segundos.

A continuación, se convierte el paralaje de la estrella, que se da en milisegundos de arco, en una medida de distancia en parsecs. Esto se hace tomando el recíproco ('1/['parallax1']') y multiplicando por '1000' para convertir de milisegundos de arco a segundos de arco. El paralaje y la distancia son inversamente proporcionales. Luego se convierte esta distancia de parsecs a kilómetros multiplicando por '3.086e13', que es el número de kilómetros en un parsec. Finalmente, se multiplica el movimiento propio (ahora en radianes por segundo) y la distancia (ahora en kilómetros) para obtener la velocidad tangencial en km/s.

$$\text{pm}_1 = \sqrt{\text{pmra}_1^2 + \text{pmdec}_1^2} \quad (6.2)$$

$$v_{\text{tang}} = \left| \text{pm}_1 \times \frac{\pi}{180} \times \frac{1}{60} \times \frac{1}{60} \times \frac{1}{1000} \times \frac{1}{31536000} \times \frac{1}{\text{parallax}_1} \times 1000 \times 3,086 \times 10^{13} \right| \quad (6.3)$$

- **Parallax difference**

Esta fórmula proporciona una medida de cuántas veces el error estándar (la raíz cuadrada de la varianza) la diferencia en paralaje entre las dos estrellas es.

Esto puede ser útil para evaluar la significancia de la diferencia observada en paralaje.

$$\text{Parallax\_diff} = \frac{\text{parallax}_1 - \text{parallax}_2}{\sqrt{\text{parallax\_error}_1^2 + \text{parallax\_error}_2^2}} \quad (6.4)$$

#### ■ Proper motion difference

La diferencia de movimiento propio es una forma de cuantificar cuánto se mueven dos estrellas en el cielo a lo largo del tiempo. Si la diferencia observada es mucho menor que la esperada, podría indicar que las dos estrellas están de hecho en órbita una alrededor de la otra.

$$\Delta\mu \leq \Delta\mu_{\text{orbit}} + 2\sigma_{\Delta\mu}, \quad (6.5)$$

Aquí,  $\Delta\mu$  se refiere a la disparidad observada en el movimiento propio, mientras que  $\sigma_{\Delta\mu}$  representa su nivel de incertidumbre. El término  $\Delta\mu_{\text{orbit}}$  es la máxima discrepancia anticipada en el movimiento propio atribuible al movimiento orbital. Ambos  $\Delta\mu$  y  $\sigma_{\Delta\mu}$  se derivan como sigue:

$$\Delta\mu = \left[ (\mu_{\alpha,1}^* - \mu_{\alpha,2}^*)^2 + (\mu_{\delta,1} - \mu_{\delta,2})^2 \right]^{1/2},$$

$$\sigma_{\Delta\mu} = \frac{1}{\Delta\mu} \left[ \left( \sigma_{\mu_{\alpha,1}^*}^2 + \sigma_{\mu_{\alpha,2}^*}^2 \right) \Delta\mu_{\alpha}^2 + \left( \sigma_{\mu_{\delta,1}}^2 + \sigma_{\mu_{\delta,2}}^2 \right) \Delta\mu_{\delta}^2 \right]^{1/2},$$

Donde  $\Delta\mu_{\alpha}^2 = (\mu_{\alpha,1}^* - \mu_{\alpha,2}^*)^2$  y  $\Delta\mu_{\delta}^2 = (\mu_{\delta,1} - \mu_{\delta,2})^2$ . Aquí  $\mu_{\alpha,i}^* \equiv \mu_{\alpha,i} \cos \delta_i$ , y tanto  $\alpha$  como  $\delta$  son la ascensión recta y la declinación respectivamente, mientras que  $\mu_{\alpha}$  y  $\mu_{\delta}$  son el movimiento propio en las respectivas direcciones de ascensión recta y declinación. Esto es considerando:

$$\Delta\mu_{\text{orbit}} = 0,44 \text{ mas yr}^{-1} \times \left( \frac{\omega}{\text{mas}} \right)^{3/2} \left( \frac{\theta}{\text{arcsec}} \right)^{-1/2}$$

### 6.3. Estandarización

La estandarización desempeña un papel crítico en la etapa de preparación de datos para la mayoría de los algoritmos de aprendizaje automático, mitigando cualquier sesgo potencial debido a las unidades de medida y la escala de las variables. En la siguiente tabla, detallamos cada parámetro junto con su unidad original, su versión estandarizada y una descripción concisa. Cabe destacar que cada parámetro se ajusta de manera específica a su distribución y naturaleza, con el fin de optimizar la eficiencia del modelo de aprendizaje automático que se implementará. En esencia, este proceso consiste en trasladar las variables a un rango dinámico, respetando de esta manera sus respectivas distribuciones.

Parametro	Unidad	Parametro escalado	Descripción
$\theta$	arsec	$\log \theta$	Angular separation
$\omega_1$	mas	$4/\omega_1$	Parallax (primary)
$\sigma_{\Delta\omega}$	mas	$4 \cdot \sigma_{\Delta\omega}$	Parallax difference error
$\Sigma_{18}$	$\text{deg}^{-2}$	$4 \cdot \log(\Sigma_{18})$	G<18 local source density
$v_{\perp,1}$	$\text{km} \cdot \text{s}^{-1}$	$v_{\perp,1}/50$	Tangential velocity (primary)
$\Delta\omega/\sigma_{\Delta\omega}$	-	$ \Delta\omega /\sigma_{\Delta\omega}$	Normalized parallax difference
$(\Delta\mu - \Delta\mu_{\text{orbit}})/\sigma_{\Delta\mu}$	-	$2\text{erf}[(\Delta\mu - \Delta\mu_{\text{orbit}})/\sigma_{\Delta\mu}]$	Scaled proper motion difference

## 6.4. Exploración de las distribuciones en el modelo de Machine Learning

Nos adentraremos en la exploración de las distribuciones en el contexto de nuestro modelo de Machine Learning. Resulta fundamental comprender cómo se distribuyen nuestras variables y cómo estas difieren entre estrellas binarias reales y alineaciones aleatorias. El análisis de las distribuciones nos permite adquirir una mayor comprensión de los datos y de su estructura subyacente, lo cual puede ser crucial para la eficacia del modelo que implementaremos.

En la figura 6.1 presentamos las distribuciones de las siete variables claves, comparando las estrellas binarias reales con las alineaciones aleatorias. Este análisis nos permitirá visualizar y entender mejor las diferencias entre estas dos categorías. Se contrarrestan las distribuciones de las estrellas binarias y las alineaciones aleatorias. Estas categorías se distinguen gracias a la probabilidad conocida como  $R_{\text{chance\_alignments}}$  (resumida como  $r_{\text{ca}}$ ), desarrollada en el estudio de Kareem El-Badry (El-Badry y cols., 2021).

Representamos la densidad estimada mediante KDE (Kernel Density Estimation) de alineaciones fortuitas en un punto  $\vec{x}$  en el espacio de parámetros de 7 dimensiones como  $\mathcal{N}_{\text{chance align}}(\vec{x})$  y la de los potenciales binarios como  $\mathcal{N}_{\text{candidates}}(\vec{x})$ . Se anticipa que la última medida es la suma de las densidades de alineaciones aleatorias y de binarios genuinos. Posteriormente calculamos el cociente entre estas dos medidas,

$$r_{\text{ca}} = \mathcal{R}(\vec{x}) = \mathcal{N}_{\text{chance align}}(\vec{x}) / \mathcal{N}_{\text{candidates}}(\vec{x}). \quad (6.6)$$

Este cociente simboliza aproximadamente la posibilidad de que un potencial binario en la posición  $\vec{x}$  sea una alineación fortuita, por lo que optar únicamente por candidatos con valores pequeños de  $\mathcal{R}$  es una estrategia eficaz para descartar las alineaciones fortuitas.  $\mathcal{R}$  no es exactamente una probabilidad, por ejemplo, no es estrictamente menor que uno, pero es una estimación valiosa para tal. Calculamos los valores de  $\mathcal{R}$  para todos los elementos de los catálogos de potenciales binarios y de alineaciones fortuitas.

Para clarificar, definimos como binarias genuinas aquellos pares de estrellas que tienen menos de un 10 % de probabilidad de ser un alineamiento, según la medida  $r_{\text{ca}}$ . Por otro lado, las alineaciones aleatorias son binarias artificiales que provienen de un catálogo creado para este propósito, denominado "Mock", como se menciona en el párrafo 7.1.

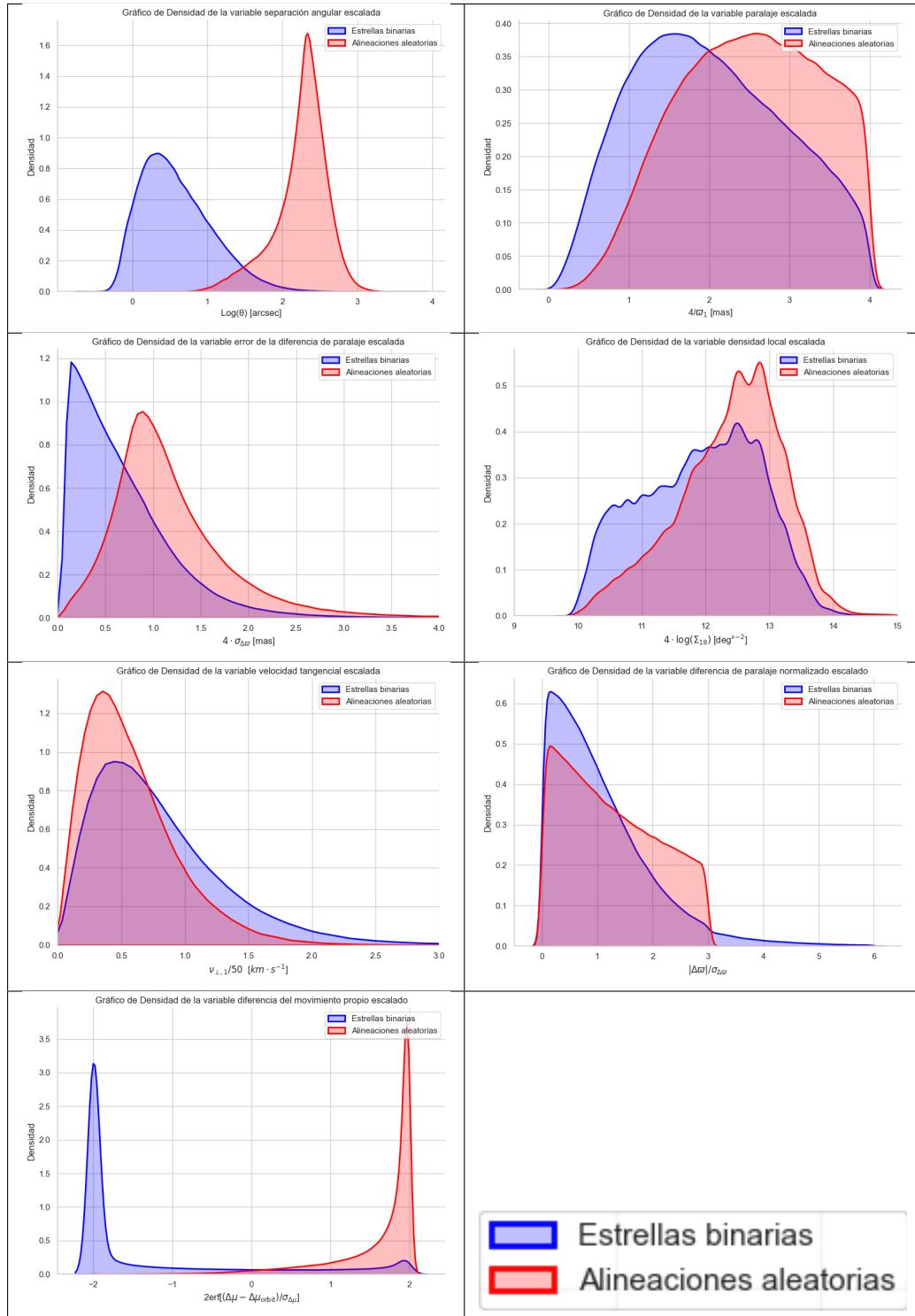
Al separar y comparar estas distribuciones, proporcionamos una imagen clara de cómo se diferencian estas dos categorías, lo que resulta esencial para nuestra comprensión de las estrellas binarias y su clasificación.

Continuando con nuestra exploración, es importante destacar que la comparación y el análisis de distribuciones se realizará en un conjunto de datos específicos.

El conjunto de datos está compuesto por:

- 500.000 Candidatos del catálogo normal con  $r_{\text{ca}} < 0.1$ .
- 250.000 Candidatos del catálogo normal con  $r_{\text{ca}} > 0.1$ .
- 250.000 Alineaciones aleatorias del catálogo Mock.

CUADRO 6.1: Distribución de las estrellas binarias reales versus alineaciones aleatorias para la variables escaladas.



Posterior a esto, agregamos una nueva columna llamada 'r\_ca', que tiene valor 1 si corresponde a un candidato binario del catalogo normal con  $r.ca < 0.1$  y 0 para el resto de candidatos. Este conjunto de datos nos permitirá llevar a cabo una comparación efectiva y detallada de las distribuciones de nuestras variables, contribuyendo significativamente a la eficacia y precisión de nuestro modelo de aprendizaje automático, su efectividad se puede apreciar en el párrafo 7.1.

## 6.5. Modelos de Aprendizaje Automático para la Clasificación de Binarias Reales y Alineamientos Aleatorios

Los modelos de machine learning, específicamente los de clasificación, pueden facilitar enormemente la tarea de distinguir entre binarias reales y alineamientos aleatorios. Sin embargo, los diferentes modelos de machine learning poseen ventajas y desventajas únicas que los hacen más adecuados para ciertos tipos de datos y tareas. A continuación, se presentan breves descripciones de varios modelos de machine learning que se han utilizado para esta clasificación.

- AdaBoost: Este enfoque de aprendizaje potenciado se centra en aprender de los errores anteriores, mejorando la clasificación en conjuntos de datos con separaciones de clases desafiantes (Freund y Schapire, 1997).
- XGBoost: Un algoritmo de boosting eficiente y rápido, especialmente hábil para manejar grandes conjuntos de datos y adaptable para problemas de regresión y clasificación (Chen y Guestrin, 2016).
- LightGBM: Destaca por su eficiencia en velocidad y memoria, con habilidades para entrenar modelos en conjuntos de datos de gran tamaño, lo que lo hace adecuado para situaciones con grandes volúmenes de información (Ke y cols., 2017).
- CatBoost: Este modelo maneja eficazmente características categóricas y tiene la ventaja de prevenir el sobreajuste, lo que lo hace útil cuando las clases no están equilibradas (Prokhorenkova, Gusev, Vorobev, Dorogush, y Gulin, 2019).
- Stochastic Gradient Descent (SGD): Método de optimización efectivo cuando el número de muestras es considerablemente grande, haciendo de SGD una opción valiosa para datasets masivos (De Sa, Feldman, Ré, y Olukotun, 2017).
- Perceptron: Un algoritmo de aprendizaje supervisado diseñado para problemas de clasificación binaria, reconocido por su simplicidad y eficacia en la separación de clases (Rosenblatt, 1958).
- Linear Discriminant Analysis (LDA): Un método que busca un límite lineal para separar de manera óptima las clases, útil cuando se asume que los datos siguen una distribución normal (FISHER, 1936).
- Quadratic Discriminant Analysis (QDA): Similar al LDA, pero permite fronteras de decisión cuadráticas, beneficioso cuando la separación entre las clases no es lineal.
- K-Nearest Neighbors (KNN): Algoritmo basado en instancias que clasifica basándose en la proximidad en el espacio de características, lo que permite agrupaciones de objetos similares (Cover y Hart, 1967).

- Support Vector Machines (SVM): Algoritmo que busca el hiperplano óptimo para separar clases, efectivo tanto en situaciones lineales como no lineales gracias al uso de kernels (Cortes y Vapnik, 1995).
- Decision Tree: Modelo de clasificación basado en decisiones jerárquicas respecto a las características, fácil de interpretar y capaz de manejar relaciones no lineales (Quinlan, 1986).
- Random Forest: Ensamble de árboles de decisión que minimiza la varianza y aumenta la precisión, útil para tratar con numerosas características y relaciones complejas entre ellas (Breiman, 2001).
- Gradient Boosting: Un algoritmo de boosting basado en optimización del gradiente, eficaz para lograr alta precisión y capaz de manejar relaciones complejas entre las características (Friedman, 2001).
- Naive Bayes: Clasificador probabilístico que utiliza el teorema de Bayes asumiendo independencia entre características, es rápido y efectivo en grandes conjuntos de datos con muchas características (Russell y Norvig, 2002).
- Logistic Regression: Clasificador que emplea una función logística para la clasificación, siendo un modelo lineal más efectivo cuando las clases pueden ser separadas por un hiperplano (Hosmer Jr, Lemeshow, y Sturdivant, 2013).
- Neural Network: Modelo inspirado en la estructura del cerebro humano, capaz de modelar relaciones complejas y no lineales entre las características (Rumelhart, Hinton, y Williams, 1986).

## 6.6. Relevancia de la Variable 'g<18 local source den' en la Capacidad Predictiva del Modelo

La medición de la densidad celeste local denominada  $\Sigma_{18}$ , ilustra el conteo de fuentes por grado cuadrado que (a) se ajustan a los límites de nuestra consulta preliminar (6.1.2) y (b) poseen una luminosidad superior a  $G = 18$ . Estimamos el valor de  $\Sigma_{18}$  alrededor de cada aspirante a binario, contabilizando la cantidad de fuentes en un radio de 1 grado del cuerpo principal y luego dividiéndolo por  $\pi$ . Los valores obtenidos para  $\Sigma_{18}$  oscilan desde 280, en dirección a los polos galácticos, hasta 8700, hacia el centro de la galaxia. Es probable que una proporción considerable de las fuentes en dirección al centro galáctico sean estrellas de fondo que no se encuentren realmente dentro del volumen de búsqueda de 1 kpc.

Este parámetro fue adaptado a un rango dinámico legible por el algoritmo de aprendizaje automático, tal como se detalla en 6.3. La variable transformada recibe el nombre de 'g18 local source den'.

Basado en la complejidad que representa calcular la densidad local para cada estrella, se planteó la posibilidad de eliminar la variable con el objetivo de optimizar recursos. Es por esto que evaluamos a profundidad el impacto de su eliminación sobre la capacidad predictiva de nuestro modelo.

El coeficiente de correlación de Pearson se utilizó para evaluar la relación lineal entre la variable 'g18 local source den' y la variable 'r\_ca'. Si el coeficiente de correlación es bajo, esto indica que las dos variables no están muy relacionadas y, por lo tanto, eliminar la variable puede tener poco impacto en la información disponible en el

conjunto de datos. Sin embargo, si el coeficiente de correlación es alto, esto indica que las dos variables están fuertemente relacionadas y eliminar la variable puede tener un impacto significativo en la información disponible en el conjunto de datos.

Por otro lado, el coeficiente de determinación ( $R^2$ ) y el error cuadrático medio (MSE) se utilizaron para evaluar la capacidad predictiva del modelo utilizando todas las variables en comparación con el modelo utilizando todas las variables menos la variable que se quiere eliminar. Si el valor de  $R^2$  es menor o el valor de MSE es mayor para el modelo que no incluye la variable que se quiere eliminar, esto puede indicar que la variable es importante para el modelo y que se pierde capacidad predictiva al eliminarla.

En este caso, los resultados indican que la eliminación de la variable `g18_local_source_den` tendría un impacto significativo en la información disponible en el conjunto de datos y en la capacidad predictiva del modelo. El coeficiente de correlación de Pearson muestra una correlación significativa entre la variable que se quiere eliminar y la variable objetivo, lo que indica que ambas variables están relacionadas. Además, los valores de  $R^2$  y MSE indican que la eliminación de la variable tendría un impacto significativo en la capacidad del modelo para predecir la variable objetivo. Por lo tanto, se recomienda no eliminar la variable `g18_local_source_den` del conjunto de datos.

Es por esto que abordamos el desafío de optimizar el cálculo de la variable  $\Sigma_{18}$ , la cual está limitada por el tamaño del catálogo estelar que se utiliza, siendo usualmente más baja en catálogos de menor tamaño.

Con el objetivo de obtener una estimación de  $\Sigma_{18}$  más precisa y eficiente desde el punto de vista computacional, propusimos diferentes métodos para optimizar su cálculo. Bajo la hipótesis de que la distribución de la densidad de las estrellas permanece inalterada y sólo se desplaza horizontalmente, sugerimos que, al estandarizar la variable, podríamos estimar la densidad local de cada estrella independientemente de la cantidad total de estrellas en el catálogo.

Para evaluar esta hipótesis, realizamos dos experimentos. En el primer experimento, seleccionamos aleatoriamente 50 estrellas y calculamos  $\Sigma_{18}$  utilizando un número variable de estrellas en el catálogo, con tamaños de 100, 1.000, 10.000, 100.000, 1.000.000, 10.000.000 y 64.000.000 de estrellas. Observamos variaciones significativas en la distribución de la densidad estelar con el tamaño del catálogo. En el segundo experimento, replicamos la metodología con 500 candidatos, omitiendo el cálculo de  $\Sigma_{18}$  con 100 estrellas debido al mayor número de candidatos. En ambos casos, nuestros resultados indican que la distribución de **la densidad estelar es sensible al tamaño del catálogo**.

Una alternativa que consideramos para abordar este problema se asemeja a la elaboración de una cartografía: en lugar de calcular  $\Sigma_{18}$  para cada candidato binario de forma individual, propusimos la creación de un “mapa de densidad” celeste. Similar a cómo una carta náutica describe la profundidad del océano, este mapa detallaría la densidad estelar de cada región del cielo. Por lo tanto, para estimar  $\Sigma_{18}$  para una estrella en particular, simplemente necesitamos ubicarla en el mapa y consultar su densidad.

La primera metodología que consideramos para implementar esta idea fue el uso de un modelo de machine learning, específicamente un modelo de regresión. Por

lo tanto, planteamos la siguiente hipótesis: ¿Es posible construir un modelo de machine learning que pueda estimar la densidad local de cada estrella proporcionando únicamente sus coordenadas celestes, ascensión recta (RA) y declinación (Dec)?

Decidimos utilizar inicialmente el modelo de Random Forest para evaluar esta hipótesis. El modelo mostró una precisión excelente, con un Mean Squared Error (MSE) de 46.62, un valor bastante bajo, teniendo en cuenta que  $\Sigma_{18}$  suele variar entre 270 y 3000. Sin embargo, nos encontramos con una dificultad práctica: el modelo resultante era demasiado grande para exportar, con un tamaño de 13.7 GB. Por lo tanto, nos vimos obligados a buscar un modelo alternativo que fuera menos costoso en términos de almacenamiento o que tuviera menos parámetros.

El Mean Squared Error (MSE) se destaca como la métrica ideal para comparar los modelos en nuestro caso debido a varias razones. En primer lugar, MSE es una métrica cuadrática que penaliza de manera significativa las discrepancias más grandes entre los valores predichos y reales. Esto es especialmente relevante en nuestro caso, dado que incluso pequeñas variaciones en la densidad de estrellas pueden tener un impacto significativo en nuestras conclusiones. Además, como MSE es una métrica promediada, es menos susceptible a ser distorsionada por valores extremos o anomalías en los datos, lo que aporta robustez a nuestra evaluación. Por último, MSE tiene la ventaja de tener la misma unidad que la variable de interés ( $\Sigma_{18}$  en nuestro caso), facilitando la interpretación y comparación de los resultados entre diferentes modelos de regresión. A continuación, describimos los resultados obtenidos al probar con otros modelos de machine learning.

Modelos	Tiempo de entrenamiento	MSE
Random Forest Regressor	22m 43s	46.62
K Neighbors Regressor	6s	53.35
XGBoost Regressor	17s	3990
CatBoost Regressor	29s	4818
Light Gradient Boosting Machine	10s	9769
Linear Regression	7s	309675

CUADRO 6.2: Tabla de modelos con tiempo de entrenamiento y MSE  
(ordenado por MSE)

Debido a su eficiencia en términos de tiempo de computación y almacenamiento, seleccionamos el modelo K-Neighbors Regressor (KNR). Vale la pena mencionar que decidimos deliberadamente sobreajustar este modelo para obtener una representación más precisa de la densidad estelar en nuestro mapa.

Para verificar la efectividad del modelo, incluso sin el sobreajuste, calculamos  $\Sigma_{18}$  para 300 estrellas, un proceso que tomó 25 minutos. Al aplicar KNR, el cálculo se completó en tan solo 0.1 segundos, con un MSE de 40.28.

## 6.7. Análisis de la Complejidad Computacional de Algoritmos para la Búsqueda de Estrellas Vecinas y Selección de Candidatos Binarios

El primer proceso, la "Búsqueda de vecinos", se centra en encontrar las estrellas dentro de un radio específico, con complejidad de  $O(n \log n + k_1^2)$ , donde  $n$  es el

total de estrellas y  $k_1$  es el número de estrellas dentro del radio de búsqueda (5 parsec proyectado).

El segundo proceso, "Selección de candidatos binarios", busca las estrellas dentro de un radio correspondiente a 1 parsec y su complejidad se define como  $O(n \log n + k_2^2)$ , donde  $k_2$  es el número de estrellas dentro de este radio.

El tercer proceso, "Búsqueda de vecinos sobre candidatos binarios", presenta una complejidad de  $O(c \log c + k_3^2)$ , donde  $c$  representa la cantidad total de candidatos binarios y  $k_3$  es el número de estrellas dentro del radio de búsqueda (5 parsec).

Finalmente, el cálculo de  $\Sigma_{18}$ , se emplea un modelo de Regresión basado en K Vecinos Cercanos (KNR). Este modelo fue entrenado con dos parámetros: ascensión recta y declinación, y se demora solo unos segundos en calcular el valor de  $\Sigma_{18}$ . Aunque la complejidad exacta de los algoritmos de aprendizaje automático puede variar dependiendo de varios factores, en términos generales, la Regresión KNR tiene una complejidad de  $O(Dn)$  para el entrenamiento y  $O(Dnk)$  para la consulta, donde  $D$  es el número de dimensiones,  $n$  es el número total de puntos de datos y  $k$  es el número de vecinos más cercanos. En este caso, como  $D = 2$  y  $k$  es pequeño (entre 1 y 5), podríamos considerar que la complejidad es prácticamente lineal.

Por lo tanto, la complejidad total del algoritmo ahora se puede expresar como  $O(2n \log n + k_1^2 + k_2^2 + c \log c + Dnk)$ , donde las constantes no afectan el crecimiento a largo plazo en la notación O-grande. Para aclarar, en este contexto  $n$  se refiere a la cantidad total de estrellas,  $c$  a la cantidad de candidatos binarios,  $k_1, k_2, k_3$  son los números de estrellas dentro de los respectivos radios de búsqueda y  $Dnk$  representa la complejidad del algoritmo de Regresión KNR. Para un análisis detallado de estas complejidades, se puede consultar el Apéndice B del estudio.

## 6.8. Aplicación y Corrección del Método KDE para el Análisis de Alineaciones en Datos Binarios: Un Enfoque Dual de Correlación Pearson-Spearman

En un esfuerzo por replicar el estudio conducido por El-Badry ([El-Badry y cols., 2021](#)) y derivar nuestra variable de interés, `r_ca`, que servirá como la pieza central para nuestros modelos de machine learning, implementamos el método KDE que se empleó en su estudio original para el mismo conjunto de datos. En los siguientes apartados, desglosaremos en profundidad este proceso, destacaremos las diferencias encontradas en nuestro enfoque, y explicaremos cómo abordamos y resolvimos estas discrepancias.

Se realizó una evaluación al método KDE (Kernel Density Estimation) sobre el conjunto de datos utilizando la función `KernelDensity` de la librería `sklearn.neighbors` ([Pedregosa y cols., 2011](#)). Para ello, calculé el KDE para binarias reales y alineaciones aleatorias mediante la definición de un kernel gaussiano con un ancho de banda de 0.2 y un algoritmo automático.

Luego se evalúo cada KDE para los parámetros de cada binario real y volví a escalar por el número de objetos de cada muestra. A continuación, calculé una proporción entre la probabilidad de una alineación de azar y la probabilidad de un binario real para cada punto, obteniendo así una probabilidad de pertenencia al gran cluster de alineaciones aleatorias.

Sin embargo, se encontró que los valores resultantes de la probabilidad no concordaban con los valores de referencia del paper que estaba analizando, y tras una revisión de los datos se indentificó que había algunos valores atípicos que alteraban los resultados. Para solucionar esto, se transformó los valores atípicos cambiándoles su probabilidad de 1e9 % a un 200 % de que sea una alineación, ya que, con ese valor basta para asegurar de que es una alineación.

Después de este cambio, se obtuvo nuevas correlaciones de Pearson y Spearman entre la probabilidad de la evaluación en contraste con los valores de referencia del paper, y los valores resultantes concordaban más con los valores de referencia del paper.

La correlación de Pearson se basa en la covarianza entre dos variables y es una medida de la fuerza y dirección de la relación lineal entre ellas. Esta correlación es adecuada para variables que tienen una distribución normal y una relación lineal.

Por otro lado, la correlación de Spearman se basa en la clasificación ordinal de los datos y no en sus valores numéricos. Es decir, mide la fuerza y dirección de la relación entre dos variables pero sin tener en cuenta su distribución o forma de distribución. Esta correlación es adecuada para variables no lineales o cuando los datos tienen valores atípicos.

Al utilizar ambas correlaciones, podemos tener una mayor confianza en los resultados obtenidos, ya que si ambas correlaciones son similares, se puede inferir que la relación entre las variables es robusta y no depende de la distribución de los datos. Si, por otro lado, las correlaciones difieren significativamente, esto puede indicar que la relación entre las variables es más compleja o que se necesita más análisis para comprender la relación.

## Capítulo 7

# Resultados y análisis

CUADRO 7.1: Rendimiento de diferentes modelos en diferentes conjuntos de datos.

Nombre del Modelo	Precisión F1 Score	Precisión de Mock	Precisión en Binarias Reales	Tiempo de Aplicación (segundos o minutos)
Decision tree	0.982	0.982	0.982	1.8 s
KNN	0.982	0.983	0.980	2m 28s
LDA	0.974	0.993	0.956	0.0 s
Naive Bayes	0.980	0.973	0.985	1.9 s
Neural Network	0.983	0.979	0.985	4.2 s
QDA	0.980	0.971	0.987	2 s
Random Forest	0.983	0.981	0.983	17 s
SVM	0.981	0.980	0.982	16m 24s
Gradient Boosting	0.982	0.982	0.982	8 s
AdaBoost	0.981	0.982	0.979	56 s
CatBoost	0.983	0.982	0.982	2 s
LightGBM	0.983	0.982	0.982	5 s
Perceptron	0.968	0.994	0.943	1.6 s
Stochastic Gradient Descent	0.981	0.979	0.983	1.6 s
XGBoost	0.982	0.982	0.982	3.2 s
Logistic Regression	0.992	0.999	0.999	2 s

### 7.1. Aplicación de KDE y Regresión Logística en el Análisis de estrellas binarias: Un Estudio Comparativo

El enfoque de este estudio implicó el uso de la Estimación de Densidad Kernel (KDE) para crear un espacio paramétrico de 7 dimensiones. En este espacio, se identificaron dos grandes cúmulos. Cada punto en este espacio paramétrico se le fue asignado una probabilidad de pertenencia a cada cúmulo, es decir, si era una estrella binaria o una alineación aleatoria. Basándose en estas probabilidades, se categorizó a cada punto. Si la probabilidad de que fuera una estrella binaria era mayor al 90 %, se consideró clase 1, de lo contrario, se consideró clase 0.

KDE demostró ser útil en este contexto, gracias a su capacidad para manejar datos multivariados y descubrir estructuras complejas dentro de los datos. Además, KDE puede generar una estimación suave de la densidad de probabilidad, lo cual es útil

para identificar áreas con alta y baja densidad de puntos. Sin embargo, KDE puede resultar computacionalmente costoso para grandes conjuntos de datos y puede ser difícil de interpretar. El proceso de cálculo del Kernel Density Estimation (KDE) utilizado en este análisis requirió una duración significativa, extendiéndose por un período de aproximadamente 17 horas.

Hemos estudiado la importancia de varias variables a través de diversos enfoques. Entre ellos se destacan los árboles de decisión, Shap-score, el Análisis de Componentes Principales (PCA), la regresión lineal con Lasso y la regresión logística. Las importancias obtenidas son las siguientes:

CUADRO 7.2: Importancias de las variables utilizando diferentes métodos

Variable	Árboles de Decisión	SHAP Score	PCA	Lasso Lineal	Regresión Logística
angularsepscaled	0.902	0.364	0.402	-0.336	-10.022
scaled_pm_diff	0.069	0.125	0.872	-0.087	-1.989
parallax_diff_error	0.007	0.010	0.149	-0.005	-3.019
parallax	0.007	0.012	0.179	-0.026	-2.399
g18_local_source_den	0.007	0.010	0.139	-0.018	-1.323
norm_parallax_diff	0.005	0.008	0.037	-0.015	-1.542
tang_vel	0.003	0.004	0.049	0.000	2.165

Dado que las importancias de las variables varían significativamente, hemos optado por desarrollar una métrica que nos permita estimar las variables más relevantes. A continuación, presentamos las métricas normalizadas para cada variable, obtenidas mediante diferentes métodos:

- $I_{\text{tree}}$ : Importancia normalizada para árboles
- $I_{\text{pca}}$ : Importancia normalizada usando PCA
- $I_{\text{lasso}}$ : Importancia normalizada usando LASSO
- $I_{\text{shap}}$ : Importancia normalizada usando SHAP

Entonces, la métrica combinada se calcula como:

$$I_{\text{combined}} = \frac{I_{\text{tree}} + I_{\text{pca}} + I_{\text{lasso}} + I_{\text{shap}}}{4}$$

CUADRO 7.3: Importancias normalizadas para cada variable.

Variable	$I_{\text{combined}}$
angularsepscaled	0.750
scaled_pm_diff	0.318
tang_vel	0.250
g18_local_source_den	0.129
parallax_diff_error	0.125
parallax	0.110
norm_parallax_diff	0.092

La métrica combinada,  $I_{\text{combined}}$ , representa el promedio de las importancias normalizadas de cada método para cada variable. Esto permite tener en cuenta el rango relativo de importancia en cada método y obtener una medida combinada de la importancia general de cada variable. El valor de  $I_{\text{combined}}$  estará en el rango de 0 a 1, donde 0 indica que la variable no es importante en ningún método y 1 indica que la variable es muy importante en todos los métodos.

Es importante destacar que, al analizar los resultados, se observa que a diferencia de otros métodos, el método de regresión logística otorga una importancia considerable a la variable "velocidad tangencial" ( $I_{\text{combined}} = 0,250$ ). Esto sugiere que dicha variable es relevante en ese métodos y puede desempeñar un papel significativo en la predicción o explicación del fenómeno en estudio.

Debido a estas variaciones en la importancia de las variables, decidimos llevar a cabo un experimento donde entrenamos todos los modelos de machine learning utilizando únicamente dos variables: la separación angular y la diferencia de movimiento propio escalada. Esto nos permitió comparar el rendimiento de los modelos al utilizar solo estas dos variables en lugar de las siete mencionadas anteriormente, excepto en el caso de la regresión logística.

En este análisis, observamos que la exclusión de las cinco variables adicionales resultó en una disminución de 0.02 en el puntaje F1 para todos los modelos, excepto para el modelo de regresión logística. Tras esta etapa, se procedió a examinar diversos modelos de machine learning para separar las dos clases identificadas. Después de un extenso análisis comparativo, se determinó que el modelo de regresión logística ofrecía el mejor rendimiento.

La regresión logística es un modelo de machine learning que se caracteriza por su simplicidad y facilidad de interpretación. Puede manejar datos categóricos y numéricos y puede trabajar con datos de alta dimensionalidad. Una ventaja adicional de este modelo es que puede generar una probabilidad de pertenencia a cada clase, lo que puede ser útil en la toma de decisiones. Sin embargo, una limitación es que este modelo puede no manejar adecuadamente estructuras de datos complejas y puede ser sensible a valores atípicos en los datos.

Al contrastar KDE y regresión logística en el contexto de la astronomía y la gestión de grandes conjuntos de datos, la regresión logística resultó ser una opción superior a KDE. Esto se debe a que proporciona una solución computacionalmente más eficiente y de interpretación más sencilla.

En todos los modelos analizados, se seleccionó el que maximizaba el puntaje F1, la precisión en el conjunto mock y en las binarias reales, minimizando simultáneamente el tiempo de procesamiento tanto en el conjunto mock como en las binarias.

$$\text{precisión} = \frac{TP}{TP + FP} \quad (7.1)$$

$$\text{exhaustividad (recall)} = \frac{TP}{TP + FN} \quad (7.2)$$

$$F1 = 2 \times \frac{\text{precisión} \times \text{exhaustividad}}{\text{precisión} + \text{exhaustividad}} \quad (7.3)$$

El puntaje F1 es una medida que se utiliza para evaluar la precisión de un modelo de clasificación, combinando las medidas de precisión y exhaustividad en un único valor. La precisión se refiere a la capacidad del modelo para identificar correctamente las observaciones positivas, en este caso, las estrellas binarias reales. Por otro lado, la exhaustividad o recall se refiere a la capacidad del modelo para encontrar todas las observaciones positivas posibles.

El **puntaje F1** se calcula utilizando la media armónica de la precisión y la exhaustividad. Al usar la media armónica en lugar de la media aritmética, se penaliza más las diferencias grandes entre la precisión y la exhaustividad. Por lo tanto, se busca un modelo que tenga un puntaje F1 alto, dado que esto indica que el modelo tiene tanto una buena precisión como una buena exhaustividad. Esto quiere decir que el modelo sea capaz de identificar correctamente las estrellas binarias (precisión) pero también que sea capaz de identificar la mayor cantidad posible de estas (exhaustividad), sin dejar de lado la eficiencia computacional y el tiempo requerido en su implementación.

En el proceso de evaluación de nuestros modelos, no se consideró necesario dar un énfasis particular a la desviación estándar de la validación cruzada. Esto se debe a que los modelos presentaron resultados sólidos y consistentes, reflejando un buen desempeño en sus predicciones. Además, las desviaciones estándar observadas fueron bajas, lo que indica una variabilidad mínima en los resultados de la validación cruzada.

Es relevante aclarar que “**Mock**” se refiere al catálogo proporcionado por el Observatorio Virtual Astrofísico Alemán ([Center, 2008](#)), producido por Rybizki ([Rybicki y cols., 2021](#)). Este catálogo, de naturaleza artificial, se fundamenta en una versión del modelo Besançon de la Vía Láctea ([Robin, A. C., Reylé, C., Derrière, S., y Picaud, S., 2003](#)), generado mediante Galaxia ([Sharma, Bland-Hawthorn, Johnston, y Binney, 2011](#)). Dado que el catálogo no incluye estrellas binarias, se puede inferir que todas las estrellas binarias obtenidas de este son en realidad alineaciones.

En contraposición, “Binarias reales” se refiere a un conjunto de datos derivados del catálogo de estrellas binarias de El-Badry ([El-Badry y cols., 2021](#)), que ha sido filtrado para retener únicamente aquellos candidatos binarios con menos del 10 % de probabilidad de ser una alineación.

En este contexto, el “tiempo” en los respectivos catálogos se refiere al lapso que cada algoritmo de aprendizaje automático requiere para discernir entre candidatos que son binarias reales y los que son alineaciones en cada catálogo.

En el cuadro 7.4 se presenta una lista detallada de los parámetros óptimos encontrados para cada uno de los modelos de clasificación utilizados en este estudio. Estos parámetros fueron determinados mediante un proceso de optimización que buscaba maximizar el rendimiento del modelo en nuestros datos de prueba.

Los parámetros presentados en la tabla son útiles no solo para entender cómo se han configurado los modelos en este estudio, sino también como un punto de partida para futuros trabajos que busquen aplicar o mejorar estos métodos de clasificación.

Cabe señalar que la selección de estos parámetros no garantiza el mismo rendimiento en todos los conjuntos de datos y contextos, ya que dependen intrínsecamente de las características y distribución de los datos específicos utilizados en este estudio.

CUADRO 7.4: Tabla de Parámetros Óptimos para Modelos de Clasificación Utilizados

Modelo	Mejores Parámetros
Naive Bayes	{var_smoothing: 1e-09}
LDA	{shrinkage: None, solver: svd}
QDA	{reg_param: 0.0, store_covariance: True, tol: 0.0001}
Logistic Regression	{'C': 1, 'class_weight': None, 'fit_intercept': True, 'max_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
Neural Network	{activation: 'tanh', alpha: 0.0001, hidden_layer_sizes: (50,), learning_rate: 'adaptive', max_iter: 200, solver: 'adam'}
Random Forest	{'max_depth': 5, 'min_samples_split': 10, 'n_estimators': 100}
KNN	{algorithm: 'ball_tree', n_neighbors: 12, weights: 'uniform'}
SVM	{C: 1, kernel: 'linear'}
Gradient Boosting	{learning_rate: 0.1, max_depth: 3, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 300, subsample: 0.8}
AdaBoost	{algorithm: 'SAMME.R', learning_rate: 0.1, n_estimators: 200}
XGBoost	{colsample_bytree: 0.8, gamma: 1, learning_rate: 0.1, max_depth: 7, n_estimators: 200, subsample: 0.8}
LightGBM	{learning_rate: 0.05, max_depth: 3, min_child_samples: 5, n_estimators: 300, num_leaves: 10, subsample: 0.8}
CatBoost	{depth: 5, iterations: 300, l2_leaf_reg: 1, learning_rate: 0.1, subsample: 0.9}
Stochastic Gradient Descent	{alpha: 0.001, l1_ratio: 0, loss: 'hinge', max_iter: 1000, penalty: 'elasticnet', tol: 1e-06}
Perceptron	{alpha: 0.001, eta0: 0.01, max_iter: 1000, penalty: 'elasticnet', tol: 0.0001}

Excluyendo la fase de generación de candidatos binarios, el algoritmo de regresión logística demostró una eficiencia impresionante, logrando clasificar estrellas binarias y alineaciones en apenas un segundo. Por el contrario, el método KDE, que implica el cálculo del espacio paramétrico en siete dimensiones para ambos catálogos -real y con estrellas desplazadas-, requirió más de 100 minutos. Esta diferencia considerable en los tiempos de procesamiento pone de relieve la superioridad de la regresión logística en términos de eficiencia computacional para este conjunto de tareas.

CUADRO 7.5: Importancia de las variables en el modelo

Variable	Porcentaje de importancia
angularsepscaled	44.81 %
parallax_diff_error	13.04 %
parallax	10.16 %
tang_vel	9.47 %
scaled_pm_diff	8.30 %
norm_parallax_diff	7.35 %
g18_local_source_den	6.87 %

Los porcentajes de importancia de cada variable fueron obtenidos utilizando una metodología basada en los coeficientes del modelo de regresión logística. Primero, se calcularon los coeficientes del modelo mediante el código proporcionado. A continuación, se calculó el valor absoluto máximo de los coeficientes sumando los valores absolutos de todos los coeficientes. Luego, se iteró sobre los coeficientes y se calculó el porcentaje dividiendo el valor absoluto de cada coeficiente entre el valor absoluto máximo y multiplicándolo por 100. Esto permitió obtener una medida relativa de la importancia de cada variable en el modelo, expresada como un porcentaje. Los porcentajes indican la contribución relativa de cada variable, donde un porcentaje más alto se asocia con una mayor importancia en el modelo.

## 7.2. Regresión Logística

Un resultado innovador en esta comparación de modelos de aprendizaje automático, así como de métodos para estimar la importancia de las variables en cada enfoque, destaca que, a diferencia de todos los demás métodos, la penalización L1 en la regresión logística es el único enfoque que mostró importancia en variables poco comunes, como la velocidad tangencial. Este hallazgo abre una puerta para un análisis más detallado, el cual se abordará a continuación.

Evaluamos la regresión logística utilizando las mismas dos variables que evaluamos en todos los otros modelos de aprendizaje automático: la separación angular y la diferencia de movimiento propio escalado. Este modelo obtuvo un puntaje F1 de 0.9810, con los siguientes parámetros óptimos: 'C': 1, 'class\_weight': None, 'fit\_intercept': True, 'max\_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'. Los resultados indicaron la siguiente importancia en las variables:

Variable	Importancia
angularsepscaled	-6.439352
scaled_pm_diff	-1.869843

Posteriormente, evaluamos el modelo utilizando las siete características, lo que resultó en un aumento mínimo de 0.0005 en el puntaje F1. Al igual que con los otros modelos, parece que estas variables aportan muy poca información.

Sin embargo, al analizar la importancia que el modelo otorga a las variables, se observa lo siguiente:

Variable	Importancia
angularsepscaled	-9.986735
parallax_diff_error	-3.007828
parallax	-2.390179
tang_vel	2.156318
scaled_pm_diff	-1.982720
norm_parallax_diff	-1.536997
g18_local_source_den	-1.321022

Optamos por transformar las variables para explorar si podían aportar más información. Con el fin de evitar transformaciones lineales, decidimos aplicar escalas logarítmicas a todas las variables, excepto la separación angular, que ya se encontraba en esa escala. Tras esta transformación, volvimos a evaluar el modelo, y se obtuvo un puntaje F1 de 0.9925 en el conjunto de prueba y un F1 score de 0.9923 en el conjunto de entrenamiento, lo que representa consistencia del rendimiento en ambos conjuntos, generalizando adecuadamente los nuevos datos evitando el sobreajuste y una mejora respecto al resultado anterior.

Posteriormente, realizamos un análisis de interacciones entre las variables, donde probamos varias combinaciones multiplicando distintas parejas de variables. Sin embargo, la mejor combinación de variables resultó en una disminución del puntaje F1 a 0.9920. Esta combinación consistía en la multiplicación de la separación angular con la diferencia de movimiento propio escalado, además de la multiplicación del paralaje con la velocidad tangencial y la multiplicación de la diferencia del error del paralaje con la velocidad tangencial.

CUADRO 7.6: Resultados de la Regresión Logística con las 7 variables en escalas logarítmicas.

Dep. Variable	No. Observations	Model	Df Residuals	Df Model
r_ca	700,000	Logit	699,992	7
		coef	std err	z
				P> z  [0.025 0.975]
const	65.3279	0.686	95.224	0.000 63.983 66.673
angularsepscaled	-10.5793	0.079	-134.025	0.000 -10.734 -10.425
parallax	-4.9962	0.054	-92.544	0.000 -5.102 -4.890
parallax_diff_error	-2.6688	0.030	-89.081	0.000 -2.728 -2.610
g18_local_source_den	-17.1670	0.237	-72.566	0.000 -17.631 -16.703
tang_vel	1.2770	0.022	59.255	0.000 1.235 1.319
norm_parallax_diff	-1.7192	0.022	-76.760	0.000 -1.763 -1.675
scaled_pm_diff	-1.8192	0.019	-98.144	0.000 -1.856 -1.783

En el siguiente análisis, exploraremos y explicaremos las diferentes columnas presentadas en la tabla de resultados de la regresión logística. Cada columna tiene una información valiosa para comprender el modelo y su capacidad para predecir la variable objetivo. Discutiremos el significado de cada columna, desde los coeficientes y su interpretación, hasta la importancia de los valores de z, p y los intervalos de

confianza. Al finalizar, obtendremos una comprensión más completa de cómo estas métricas nos permiten evaluar la relevancia de las variables predictoras y la robustez del modelo en la clasificación binaria.

1. **coef**: El coeficiente representa la estimación del cambio en el logaritmo de la odds (razón de probabilidades) de la variable dependiente (*r\_ca*) asociado a cada variable predictora. Es una medida de la contribución de cada variable en la predicción de la variable objetivo. Un coeficiente positivo indica que un incremento en la variable predictora está asociado con un aumento en la probabilidad de la variable dependiente, mientras que un coeficiente negativo indica lo contrario.
2. **std err**: El error estándar representa la incertidumbre asociada a la estimación del coeficiente. Mide cuánto varía el coeficiente de una muestra a otra. Valores pequeños de error estándar indican una estimación más precisa y confiable del coeficiente.
3. **z**: El valor z (Z-score) es una medida estandarizada que indica cuántas desviaciones estándar se encuentra el coeficiente por encima o por debajo de cero. Un valor z grande (positivo o negativo) indica que el coeficiente es significativamente diferente de cero, lo que sugiere que la variable predictora es relevante para el modelo.
4. **P>| z |**: Es el valor p asociado al valor z, que representa la probabilidad de obtener un valor z tan extremo o más extremo que el observado, asumiendo que el coeficiente es realmente cero. Un valor p pequeño (generalmente menor a 0.05) sugiere que el coeficiente es estadísticamente significativo y que la variable predictora es relevante para la predicción del modelo.
5. **[0.025, 0.975]**: Estos valores representan los intervalos de confianza al 95 % para cada coeficiente. Indican el rango dentro del cual se espera que esté el coeficiente verdadero con un nivel de confianza del 95 %. Si el intervalo contiene el valor cero, significa que no podemos estar seguros de que el coeficiente sea diferente de cero y, por lo tanto, la variable podría no ser significativa para el modelo.

Las variables predictoras que tienen mayor importancia en la predicción de la variable objetivo (*r\_ca*) son ‘angularsepscaled’, y ‘g18\_local\_source\_den’. Estas variables tienen los coeficientes más grandes en valor absoluto, lo que indica que ejercen una influencia significativa en la log-odds de la variable dependiente.

Significado de los signos:

- Un coeficiente positivo indica que un incremento en la variable predictora está asociado con un aumento en la log-odds de la variable dependiente (*r\_ca*). En este caso, la variable ‘tang\_vel’ tiene un coeficiente positivo de 1.2770, lo que sugiere que un aumento en ‘tang\_vel’ está relacionado con un aumento en la probabilidad de tener el valor de *r\_ca* en la categoría positiva.
- Un coeficiente negativo indica que un incremento en la variable predictora está asociado con una disminución en la log-odds de la variable dependiente. Varias variables como ‘angularsepscaled’, ‘parallax’, ‘parallax\_diff\_error’, ‘g18\_local\_source\_den’, ‘norm\_parallax\_diff’ y ‘scaled\_pm\_diff’ tienen coeficientes negativos, lo que sugiere que un aumento en estas variables se relaciona con

una disminución en la probabilidad de tener el valor de *r\_ca* en la categoría positiva.

Los errores estándar son pequeños en general, lo que indica que las estimaciones de los coeficientes son precisas y confiables. Esto es importante porque un error estándar pequeño sugiere que las estimaciones de los coeficientes no varían significativamente de una muestra a otra, lo que brinda mayor confianza en la robustez de las conclusiones obtenidas del modelo.

Los valores de *z* para todos los coeficientes son altos, lo que sugiere que los coeficientes son significativamente diferentes de cero, y las variables predictoras son relevantes para el modelo.

Todos los valores *p* son muy bajos (0.000), lo que indica que los coeficientes son estadísticamente significativos y las variables predictoras tienen un impacto significativo en la predicción del modelo.

Los intervalos de confianza al 95 % para todos los coeficientes no incluyen el valor cero, lo que respalda aún más la significancia de las variables predictoras. Es por esto que hemos seleccionado como mejor modelo la regresión logística con las 7 variables en escalas logarítmicas.

Los parámetros óptimos obtenidos deben ser utilizados para ajustar el modelo de regresión logística, asegurando así los mejores resultados.

CUADRO 7.7: Grid de Hiperparámetros para Regresión Logística

<b>param_grid</b>
<b>penalty:</b> ['l1', 'l2']
<b>C:</b> [0.01, 0.1, 1, 10, 100]
<b>fit_intercept:</b> [True, False]
<b>max_iter:</b> [100, 200, 500]
<b>class_weight:</b> [None, 'balanced']
<b>solver:</b> ['lbfgs', 'liblinear', 'saga']

Hemos utilizado una grilla de hiperparámetros para encontrar los valores óptimos porque en el proceso de entrenamiento de modelos de aprendizaje automático, los hiperparámetros son parámetros externos al modelo que deben ajustarse antes de entrenar el modelo en sí. Estos hiperparámetros controlan diferentes aspectos del modelo y pueden afectar significativamente su rendimiento y capacidad de generalización.

La grilla de hiperparámetros es una técnica común de búsqueda sistemática que nos permite probar diferentes combinaciones de valores para los hiperparámetros, de manera exhaustiva o utilizando un muestreo estratégico, y evaluar el rendimiento del modelo con cada combinación. Esto se hace típicamente mediante validación cruzada, donde se divide el conjunto de datos en varios conjuntos más pequeños para entrenar y evaluar el modelo de forma más robusta.

En nuestro caso, hemos utilizado la métrica F1-score para evaluar el rendimiento del modelo en cada combinación de hiperparámetros. Seleccionar los hiperparámetros con el puntaje de F1-score más alto nos permite obtener un modelo con un mejor rendimiento en términos de la capacidad de clasificación y la capacidad para generalizar a datos no vistos. La grilla de hiperparámetros nos ayuda a evitar la selección

manual de hiperparámetros, lo que podría ser subjetivo y no óptimo, y nos permite encontrar la mejor configuración para nuestro modelo de manera sistemática y automatizada.

Después de explorar exhaustivamente las 360 combinaciones con los hiperparámetros mencionados, se identificó la combinación óptima que logró el mejor F1 score. Los resultados más destacados son los siguientes:

- C: 1
- class\_weight: None
- fit\_intercept: True
- max\_iter: 100
- penalty: l2
- solver: newton-cg

Al aplicar la regresión logística a nuestro modelo de clasificación, se obtuvo una puntuación media de validación cruzada (mean\_test\_score) de 0.9925. Esta métrica proporciona una estimación confiable del rendimiento del modelo durante la fase de pruebas.

Además, la desviación estándar asociada a los puntajes de validación cruzada fue relativamente baja, con un valor de 0.002162, lo que sugiere que nuestro modelo muestra una buena consistencia en diferentes conjuntos de datos de prueba.

Al obtener los coeficientes de esta regresión logística, observamos cambios drásticos en la importancia de las características. Durante el análisis, notamos que la variable g18\_local\_source\_den se ha destacado como una de las variables con mayor coeficiente. Este hallazgo tiene sentido desde el punto de vista físico, ya que, considerando su signo, es lógico inferir que a medida que la densidad de la zona de origen del par de estrellas disminuye, es más probable que se trate de un par binario genuino en lugar de un simple alineamiento fortuito.

En el contexto de la regresión logística, el parámetro C se refiere a la inversa de la fuerza de regularización, que controla la cantidad de ajuste permitido en el modelo. Un valor menor de C produce un modelo más regularizado, lo que implica que se otorga mayor importancia a la penalización de los coeficientes en la regresión logística. Por otro lado, un valor mayor de C reduce la regularización y permite un ajuste más complejo del modelo.

En particular, cuando C=1 en la regresión logística, significa que la regularización no se ha intensificado ni disminuido. Es decir, la fuerza de regularización es moderada, lo que permite cierta complejidad en el modelo mientras se intenta evitar el sobreajuste. Este valor de C se considera comúnmente como un punto de partida razonable para el ajuste de la regresión logística.

Un valor demasiado bajo de C puede resultar en un modelo demasiado simplista y subajustado, mientras que un valor demasiado alto de C puede resultar en un modelo demasiado complejo y sobreajustado.

class\_weight=None significa que no se aplica ningún peso a las clases en la regresión logística y se trata a todas las clases por igual. Esto puede ser adecuado si las clases están equilibradas en los datos, pero si hay un desequilibrio de clase, es posible que

desee considerar la asignación de pesos personalizados a las clases para mejorar el rendimiento del modelo.

`Fit_intercept=True` significa que se ajustará un término de intercepción en el modelo de regresión logística. Es decir, el modelo tendrá en cuenta una constante adicional en la ecuación de la regresión logística, además de los coeficientes de las características de entrada. Este término de intercepción se utiliza para ajustar la línea de regresión logística para que pase a través de la media de los valores de la variable de respuesta en los datos de entrenamiento.

El parámetro `max_iter` se refiere al número máximo de iteraciones permitidas para que el algoritmo de optimización converja y alcance una solución para los coeficientes de la regresión logística. Específicamente, `max_iter=100` significa que el algoritmo de optimización realizará hasta 100 iteraciones para encontrar los coeficientes óptimos de la regresión logística. Si el algoritmo no converge después de 100 iteraciones, el modelo utilizará los coeficientes encontrados en la última iteración.

Es importante ajustar el valor de `max_iter` correctamente, ya que si se establece demasiado bajo, es posible que el algoritmo no tenga suficiente tiempo para converger y alcance una solución subóptima. Por otro lado, si se establece demasiado alto, el algoritmo puede tardar más en converger y aumentar el tiempo de entrenamiento del modelo.

El parámetro `penalty` se refiere al tipo de regularización que se aplica a los coeficientes de la regresión logística. Específicamente, `penalty='l2'` significa que se aplica una regularización de norma L2 a los coeficientes de la regresión logística. La regularización L2 agrega una penalización al cuadrado de los coeficientes al costo de la función de la regresión logística, lo que tiene el efecto de disminuir los valores de los coeficientes, reduciendo así la complejidad del modelo y evitando el sobreajuste.

Otra opción posible para el parámetro `penalty` es `penalty='l1'`, lo que significa que se aplica una regularización de norma L1 a los coeficientes de la regresión logística. En este caso, la regularización L1 agrega una penalización en valor absoluto a los coeficientes al costo de la función de la regresión logística, lo que tiene el efecto de reducir algunos de los coeficientes a cero, lo que puede ayudar a seleccionar las características más importantes y reducir la complejidad del modelo.

El parámetro `solver` se refiere al algoritmo de optimización utilizado para encontrar los coeficientes óptimos de la regresión logística. Específicamente, `solver='newton-cg'` significa que se utiliza el método de Newton conjugado para optimizar la función de costo de la regresión logística. El método de Newton conjugado es un algoritmo de optimización de segundo orden que utiliza la información de la matriz Hessiana de la función de costo para encontrar una solución óptima. Este algoritmo puede ser eficiente para conjuntos de datos pequeños y medianos.

La fórmula general de la regresión logística es:

$$p(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 - \beta_1 \cdot x_1 - \beta_2 \cdot x_2 - \beta_3 \cdot x_3 - \beta_4 \cdot x_4 + \beta_5 \cdot x_5 - \beta_6 \cdot x_6 - \beta_7 \cdot x_7)}}$$

Donde:

- $p(y = 1|x)$  es la probabilidad condicional de que la variable dependiente ( $y$ ) sea igual a 1 dado un conjunto de variables independientes ( $x$ ).

- $e$  es la base del logaritmo natural (aproximadamente 2.71828).
- $\beta_0 = 65,3279$
- $\beta_1 = -10,5793$
- $\beta_2 = -4,9962$
- $\beta_3 = -2,6688$
- $\beta_4 = -17,1670$
- $\beta_5 = 1,2770$
- $\beta_6 = -1,7192$
- $\beta_7 = -1,8192$
- $x_1$  corresponde a angularsepscaled
- $x_2$  corresponde a parallax
- $x_3$  corresponde a parallax\_diff\_error
- $x_4$  corresponde a g18\_local\_source\_den
- $x_5$  corresponde a tang\_vel
- $x_6$  corresponde a norm\_parallax\_diff
- $x_7$  corresponde a scaled\_pm\_diff

Para una comprensión más profunda y detallada de la Regresión Logística, le invitamos a consultar el Apéndice C. Allí, hemos preparado una revisión exhaustiva y meticulosa de este método de clasificación, incluyendo detalles adicionales que podrían ser de interés para el lector en busca de un conocimiento más completo en la materia.

Como se puede observar en la figura 7.1 y 7.2, se examina un par binario específico derivado de la aplicación de nuestro algoritmo de clasificación. Este par binario es el más luminoso del catálogo generado, identificadas por sus respectivas source\_id: 376603455732600832 y 376603455732601344. Las magnitudes fotométricas G medias (Phot\_G\_mean\_mag) de estas estrellas son 6.079968 y 6.943458, respectivamente.

La primera imagen es una vista óptica de la estrella Gaia DR3 376603455732601344, obtenida de la base de datos de Gaia en ESA's Gaia Archive con colores DSS2. La segunda imagen representa el mismo objeto estelar, pero en un rango de longitud de onda cercano al infrarrojo, utilizando la codificación de color JHK de la base de datos 2MASS. Estas representaciones permiten una mejor apreciación de las características de la estrella, ya que cada rango de longitud de onda revela detalles diferentes, brindando así una visión más completa del objeto en estudio.

### 7.3. Validación y Precisión de la Regresión Logística en la Detección de Alineaciones Estelares: Un Estudio Basado en Simulaciones

Para validar el algoritmo propuesto, se generó un catálogo de prueba conformado exclusivamente por alineaciones estelares. Para ello, se desplazaron las estrellas



**Figura 7.1:** Vista óptica de Gaia DR3 376603455732601344



**Figura 7.2:** Representación infrarroja de Gaia DR3 376603455732601344

reales en su ascensión recta, sumando o restando 0.5 veces la secante de la declinación de la estrella en radianes, según cuál ascensión recta fuese mayor. Este proceso se llevó a cabo antes de crear pares de estrellas, asegurando que todas las parejas en el catálogo de prueba representaran alineaciones. No obstante, se preservaron las estadísticas de alineación aleatoria, ya que la densidad de las estrellas en nuestro volumen de búsqueda de 1 kpc no experimenta variaciones significativas en escalas de 0.5 grados.

Posteriormente, se aplicó el algoritmo de clasificación de regresión logística en este conjunto de datos, que contenía un millón de estrellas. Como resultado, se identificaron 33.185 pares de estrellas, de los cuales, debido al desplazamiento aplicado, todos deberían representar alineaciones. Sin embargo, el algoritmo detectó 1 estrella binaria real, lo que constituye un error en la clasificación. De esta manera, se puede afirmar que el algoritmo alcanzó una precisión del 99,9969 % en 8 minutos con 4 segundos.

Este procedimiento puede ser ilustrado mediante una analogía con un salón de baile ([Lépine y Bongiorno, 2007](#)). Imagina una multitud de hombres y mujeres que asisten a un evento de baile en un amplio salón. Supón que algunos de los asistentes son parejas casadas, mientras que el resto son solteros. Inicialmente, todos los bailarines se distribuyen al azar por el salón, pero todas las parejas casadas se mantienen tomadas de la mano. Las personas se encontrarían en parejas, algunas porque son parejas casadas, y otras simplemente por casualidad. En un momento específico, un anunciador ordena a todos los hombres que den 10 pasos a su derecha. Después de esto, se formarían nuevas parejas, pero en esta ocasión ninguna sería una pareja casada, todas serían emparejamientos por casualidad. Si calculáramos cuántas parejas había inicialmente y restáramos la cantidad de parejas después del movimiento, entonces tendríamos una buena estimación de cuántas parejas casadas están asistiendo al baile.

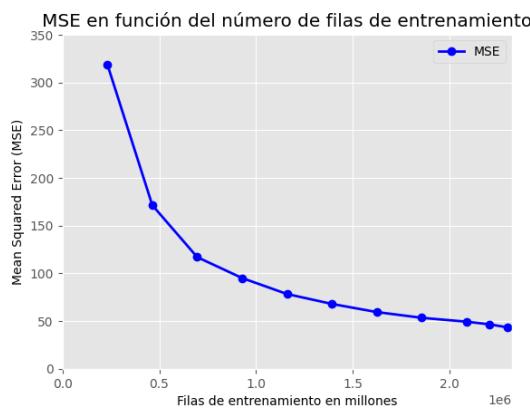
Este mismo principio es el que se aplica en nuestro estudio. La reubicación de las estrellas en nuestra simulación es análoga a los hombres dando pasos a la derecha en el salón de baile. Al hacer esto, podemos identificar cuántas estrellas están realmente alineadas (las ‘parejas casadas’) en comparación con las que simplemente aparecen en pares debido a su proximidad aleatoria (los ‘solteros’).

## 7.4. Mejora en la Eficiencia y Precisión: Aplicación de K-Neighbors Regressor para la Detección de Estrellas Binarias

En este estudio, abordamos la complejidad computacional asociada con el cálculo de  $\Sigma_{18}$ , una medida cuantitativa de la densidad local de las estrellas. Esta métrica, a pesar de su importancia en el análisis de densidad estelar, presenta un desafío debido a su alto coste computacional y la dependencia de su precisión con el tamaño del catálogo estelar. Como alternativa, propusimos la creación de un “mapa de densidad” celeste, similar a una carta náutica, que refleja la densidad de estrellas en diferentes regiones del cielo.

Para implementar esta idea, recurrimos a la regresión de machine learning, planteando la hipótesis de que un modelo de este tipo podría estimar la densidad local de cada estrella basándose únicamente en sus coordenadas celestes. Después de varias pruebas con diferentes modelos, seleccionamos el K-Neighbors Regressor (KNR) por su eficiencia en tiempo de computación y almacenamiento, incluso cuando se sobreajusta deliberadamente para mejorar la precisión del mapa de densidad.

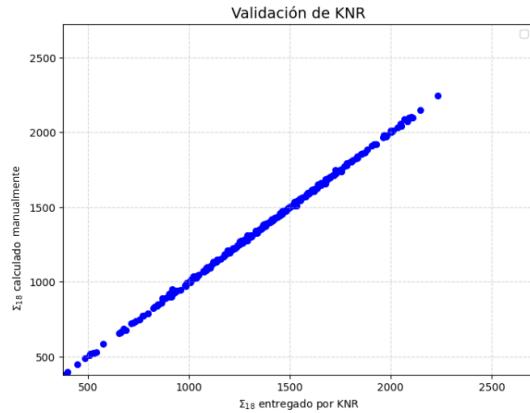
Para corroborar la eficacia del método Kernel Nearest Neighbor Regression (KNR) en la estimación del parámetro  $\Sigma_{18}$ , se llevó a cabo un proceso de validación manual. De un catálogo compuesto por 64 millones de estrellas, se seleccionó una muestra de 300 para el cálculo independiente de  $\Sigma_{18}$ . Los resultados se compararon luego con las predicciones suministradas por el modelo KNR, lo que se representó en un diagrama de dispersión que manifestó una correlación lineal de tipo  $y = x$ . Esta observación respalda la precisión de KNR al generar resultados en intervalos de tiempo notablemente cortos. Es importante subrayar que el entrenamiento del modelo KNR se llevó a cabo únicamente con 2.1 millones de estrellas. No obstante, evidencia una precisión consistente en la estimación de  $\Sigma_{18}$  para cualquier estrella que se encuentre dentro del catálogo de 64 millones. Esto es debido a que al aumentar el entrenamiento haría variar muy poco el MSE, como se puede ver en el Figura 7.3.



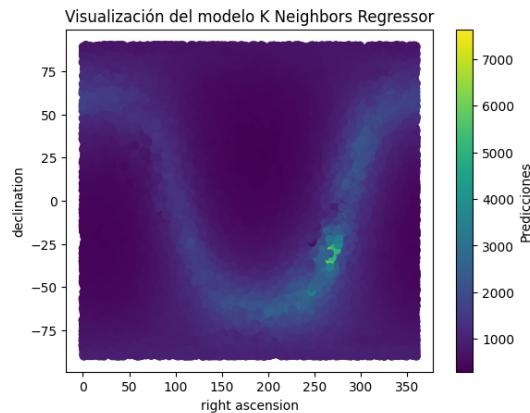
**Figura 7.3:** Variación del Mean Squared Error (MSE) en función del número de filas de entrenamiento. Se observa una tendencia decreciente a medida que aumenta el tamaño del conjunto de entrenamiento.

Esta optimización permitió que nuestro algoritmo, al procesar un millón de estrellas, identificara 22.423 estrellas binarias en solo 7 minutos y 13 segundos. En comparación, el código original identificaba 22.054 estrellas binarias en un tiempo de 43

minutos. La eficiencia mejorada y la precisión comparable sugieren que nuestro enfoque de "mapa de densidad" utilizando KNR puede ser una herramienta valiosa para el análisis de la densidad estelar en grandes conjuntos de datos astronómicos.



**Figura 7.4:** El gráfico corresponde al método de valiación del modelo de KNR



**Figura 7.5:** Mapa de densidad de estrellas obtenido con el modelo K-Neighbors Regressor (KNR). El eje X representa la ascensión recta (RA), mientras que el eje Y muestra la declinación (Dec). La densidad estelar en diferentes regiones del cielo se refleja en la variación de colores en el gráfico, proporcionando una representación visual y cuantitativa de la distribución de estrellas en el cielo observable.

## Capítulo 8

# Conclusiones

### 8.1. Conclusiones generales

En términos de los objetivos propuestos al inicio de esta investigación, podemos afirmar con confianza que se han cumplido con éxito. A través de un análisis cuidadoso y exhaustivo, hemos probado y comparado varios algoritmos en términos de precisión y eficiencia en la clasificación de estrellas binarias. Este esfuerzo nos ha permitido evaluar de manera precisa y objetiva el rendimiento de estos algoritmos y determinar si pueden superar a la Estimación de Densidad Kernel (KDE) en este aspecto. Los datos del proyecto GAIA eDR3 han sido de gran utilidad para este propósito, proporcionando una base sólida y robusta para nuestras investigaciones.

Respecto a la hipótesis que planteamos inicialmente, nuestros resultados la validan. Hemos encontrado que existen algoritmos que pueden superar a KDE en términos de precisión y eficiencia para la clasificación de estrellas binarias. Entre los algoritmos evaluados, la regresión logística mostró resultados particularmente sobresalientes. Este hallazgo respalda y valida nuestra hipótesis, demostrando que, de hecho, hay alternativas a KDE que pueden brindar mejoras significativas en el ámbito de la clasificación de estrellas binarias.

Los catálogos estelares, especialmente los que se refieren a estrellas binarias, representan un recurso valioso para la comunidad científica. No sólo proporcionan una base sólida para futuros estudios, sino que también pueden impulsar descubrimientos innovadores y permitir una comprensión más profunda de nuestra galaxia y del universo en su conjunto. El catálogo que hemos desarrollado no es una excepción a esto, y su importancia radica en su capacidad para proporcionar datos exhaustivos y precisos de una amplia gama de estrellas binarias. Además de la mera recopilación de datos, nuestro enfoque se ha centrado en asegurar la calidad y la precisión de la información contenida en nuestro catálogo. En este sentido, la creación del catálogo tiene las siguientes funcionalidades:

- Calibración de edades estelares: Un catálogo de estrellas binarias reales proporcionaría información valiosa sobre la evolución estelar y las relaciones entre las propiedades físicas de las estrellas. Esto permitiría calibrar de manera más precisa las estimaciones de las edades estelares en diversas regiones del espacio.
- Relaciones de masa inicial-final: Las estrellas binarias tienen una relación compleja entre sus masas iniciales y finales, que está influenciada por procesos de evolución estelar, interacciones y transferencia de masa. Un catálogo de estrellas binarias real permitiría establecer relaciones más precisas entre las masas

iniciales y finales de las estrellas, lo cual es fundamental para comprender la evolución estelar y la formación de objetos compactos como las enanas blancas.

- Cálculo de masas de enanas blancas: El corrimiento al rojo gravitacional es un fenómeno que ocurre en sistemas binarios donde una enana blanca, debido a su alta densidad, produce un desplazamiento hacia el rojo en la luz emitida por su compañera. Un catálogo de estrellas binarias reales permitiría utilizar este efecto para calcular con mayor precisión las masas de las enanas blancas y comprender mejor sus propiedades.
- Estimación de la abundancia de estrellas progenitoras de enanas blancas: Al estudiar las características de las estrellas binarias reales, se puede obtener información sobre las etapas previas a la formación de enanas blancas. Esto incluye la estimación de la abundancia de estrellas progenitoras de enanas blancas, lo cual es importante para comprender la evolución estelar y la formación de estos objetos.
- Estudio de las masas de estrellas: El catálogo de estrellas binarias reales proporcionaría una muestra amplia y diversa de sistemas estelares con información precisa sobre sus masas. Esto permitiría investigar las propiedades y distribución de las masas estelares en el universo y mejorar nuestro conocimiento sobre la formación y evolución estelar.
- Determinación de los radios de las estrellas involucradas: En ciertos casos favorables, las estrellas binarias reales podrían proporcionar datos que permitan determinar los radios de las estrellas involucradas en los sistemas binarios. Esto sería útil para comprender mejor la estructura y propiedades físicas de las estrellas, así como para validar y mejorar los modelos teóricos.

## 8.2. Trabajo futuro

Aunque este trabajo ha dado pasos significativos en el campo de la identificación de estrellas binarias, reconocemos que todavía hay mucho que explorar y perfeccionar. La ciencia evoluciona continuamente y la astronomía no es una excepción a este hecho. Con miras al futuro, tenemos varias líneas de investigación y mejoras que estamos considerando. A continuación, presentamos algunas de las áreas que consideramos para el trabajo futuro:

Este estudio no se centró detalladamente en el análisis de la variable 'bp\_rp', un indicador esencial del color estelar. Este parámetro, que representa la diferencia de color BP - RP, podría ser clave para entender la composición intrínseca de cada sistema binario. Específicamente, nuestro interés reside en determinar si el sistema binario está compuesto por estrellas enanas blancas (WD), estrellas de la secuencia principal (MS), o cualquier combinación posible entre estas. Por lo tanto, el análisis de 'bp\_rp' podría proporcionar una perspectiva crucial para la discriminación precisa de estas categorías estelares.

Con este fin, en un estudio futuro, proponemos utilizar el siguiente enfoque para inferir esta información. Primero, se recuperan los valores de 'bp\_rp', 'phot\_g\_mean\_mag' y 'parallax' para cada estrella del catálogo limpio. Estas variables nos permiten calcular la magnitud absoluta de las estrellas,  $M_g$ , utilizando la relación:

$$Mg = G + 5 \times \log_{10} \left( \frac{\text{parallax}}{100} \right) \quad (8.1)$$

donde  $G$  es la magnitud promedio en la banda G. Luego, se establece una serie de condiciones para clasificar cada estrella como WD o MS basándose en sus propiedades de color y magnitud absoluta. Específicamente, una estrella se clasifica como WD si su magnitud absoluta es mayor que el producto de su índice de color y 3.25, sumado a 9.625.

Una vez clasificadas las estrellas, podemos determinar la naturaleza del sistema binario en base a sus componentes. Los sistemas pueden ser 'WDWD' (ambas estrellas son enanas blancas), 'WDMS' (una estrella enana blanca y una estrella de la secuencia principal), 'MSMS' (ambas estrellas son de la secuencia principal) entre otras combinaciones. Los sistemas con estrellas para las que no se tiene información de color son etiquetados como desconocidos.

Finalmente, esta nueva información es agregada al catálogo y guardada para su uso en futuras investigaciones. Este análisis amplía nuestra comprensión de los sistemas binarios y nos permitirá hacer predicciones más precisas sobre su comportamiento.

Hacia el futuro, este algoritmo, además, podría abrir nuevas vías para el refinamiento y la validación de los catálogos existentes de estrellas binarias. Este enfoque computacionalmente avanzado ofrece la posibilidad de corroborar y desambiguar los múltiples sistemas de estrellas incluidos en los catálogos actuales, como GaiaDR2, el catálogo del Grupo Binario Kepler y la Lista de Identificación de Binarias (ILB), mencionados en el párrafo 3. La capacidad de distinguir de manera concluyente entre binarias verdaderas y alineaciones fortuitas permitiría una mayor precisión en la catalogación, al mismo tiempo que eliminaría la incertidumbre asociada con las identificaciones ambiguas. Este trabajo podría aportar una contribución significativa para garantizar la exactitud y la coherencia de estos valiosos repositorios de datos, permitiendo así un mayor avance en el entendimiento de los sistemas binarios en nuestra galaxia.

### 8.3. Código y Uso

El código completo de nuestro algoritmo se encuentra disponible para su consulta pública en nuestro repositorio GitHub, cuya dirección URL es:

[github.com/Lucianogodoi/find\\_binaries](https://github.com/Lucianogodoi/find_binaries).

En este repositorio se incluye un archivo README.md con instrucciones detalladas para la identificación de estrellas binarias a partir de su archivo .csv personalizado.

Con el método de interpolación de KNR calculamos el  $\Sigma_{18}$  de las 64.407.853 estrellas del catálogo GAIA eDR3 en tan solo 2 minutos con 43 segundos, disponibles también para descargar.

## Referencias

- Abate, C., Pols, O. R., Karakas, A. I., y Izzard, R. G. (2015). Carbon-enhanced metal-poor stars: a window on agb nucleosynthesis and binary evolution - i. detailed analysis of 15 binary stars with known orbital periods. *A&A*, 576, A118. Descargado de <https://doi.org/10.1051/0004-6361/201424739> doi: 10.1051/0004-6361/201424739
- Andrews, J. J., Chanamé, J., y Agüeros, M. A. (2017, 08). Wide binaries in Tycho-Gaia: search method and the distribution of orbital separations. *Monthly Notices of the Royal Astronomical Society*, 472(1), 675-699. Descargado de <https://doi.org/10.1093/mnras/stx2000> doi: 10.1093/mnras/stx2000
- Backurs, A., Indyk, P., y Wagner, T. (2019). Space and time efficient kernel density estimation in high dimensions. En H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, y R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Descargado de [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/a2ce8f1706e52936dfad516c23904e3e-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/a2ce8f1706e52936dfad516c23904e3e-Paper.pdf)
- Barbierato, E., Gribaudo, M., y Iacono, M. (2013, 01). Performance evaluation of nosql big-data applications using multi-formalism models. *Future Generation Computer Systems, to appear*. doi: 10.1016/j.future.2013.12.036
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. Descargado de <http://dx.doi.org/10.1023/A%3A1010933404324> doi: 10.1023/A:1010933404324
- Center, G. D. (2008). *ADQL query*. VO resource provided by the GAVO Data Center. Descargado de [https://dc.zah.uni-heidelberg.de/\\_system\\_/adql/query/info](https://dc.zah.uni-heidelberg.de/_system_/adql/query/info)
- Chen, T., y Guestrin, C. (2016). XGBoost: A scalable tree boosting system. , 785–794. Descargado de <http://doi.acm.org/10.1145/2939672.2939785> doi: 10.1145/2939672.2939785
- Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- Cover, T., y Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27. doi: 10.1109/TIT.1967.1053964
- De Sa, C., Feldman, M., Ré, C., y Olukotun, K. (2017, jun). Understanding and optimizing asynchronous low-precision stochastic gradient descent. *SIGARCH Comput. Archit. News*, 45(2), 561–574. Descargado de <https://doi.org/10.1145/3140659.3080248> doi: 10.1145/3140659.3080248
- El-Badry, K., y Rix, H.-W. (2018, 08). Imprints of white dwarf recoil in the separation distribution of Gaia wide binaries. *Monthly Notices of the Royal Astronomical Society*, 480(4), 4884-4902. Descargado de <https://doi.org/10.1093/mnras/sty2186> doi: 10.1093/mnras/sty2186
- El-Badry, K., Rix, H.-W., y Heintz, T. M. (2021, 02). A million binaries from Gaia eDR3: sample selection and validation of Gaia parallax uncertainties. *Monthly Notices of the Royal Astronomical Society*, 506(2), 2269-2295. Descargado de <https://doi.org/10.1093/mnras/stab323> doi: 10.1093/mnras/stab323
- FISHER, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179-188. Descargado de <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x> doi: <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- Freund, Y., y Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System*

- Sciences*, 55(1), 119-139. Descargado de <https://www.sciencedirect.com/science/article/pii/S002200009791504X> doi: <https://doi.org/10.1006/jcss.1997.1504>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189 – 1232. Descargado de <https://doi.org/10.1214/aos/1013203451> doi: 10.1214/aos/1013203451
- Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., Brown, A. G. A., Vallenari, A., Babusiaux, C., ... Zschocke, S. (2016). The gaia mission. *A&A*, 595, A1. Descargado de <https://doi.org/10.1051/0004-6361/201629272> doi: 10.1051/0004-6361/201629272
- Hartman, Z. D., y Lépine, S. (2020, apr). The SUPERWIDE catalog: A catalog of 99,203 wide binaries found inigaia/iand supplemented by the SUPERBLINK high proper motion catalog. *The Astrophysical Journal Supplement Series*, 247(2), 66. Descargado de <https://doi.org/10.3847%2F1538-4365%2Fab79a6> doi: 10.3847/1538-4365/ab79a6
- Hosmer Jr, D. W., Lemeshow, S., y Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.
- Isaeva, A. A. (2012, marzo). On the selection effects in catalogues of binary stars. *Advances in Astronomy and Space Physics*, 2, 25-27.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. En I. Guyon y cols. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Descargado de [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf)
- Lépine, S., y Bongiorno, B. (2007, marzo). New Distant Companions to Known Nearby Stars. II. Faint Companions of Hipparcos Stars and the Frequency of Wide Binary Systems. , 133(3), 889-905. doi: 10.1086/510333
- Malkov, O., Karchevsky, A., Kaygorodov, P., y Kovaleva, D. (2016, enero). Identification list of binaries. *Baltic Astronomy*, 25, 49-52. doi: 10.1515/astro-2017-0109
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., y Gulin, A. (2019). *Catboost: unbiased boosting with categorical features*.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Riquelme Santos, J. C., Ruiz, R., y Gilbert, K. (2006). Minería de datos: Conceptos y tendencias. *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial*, 10 (29), 11-18..
- Robin, A. C., Reylé, C., Derrière, S., y Picaud, S. (2003). A synthetic view on structure and evolution of the milky way. *A&A*, 409(2), 523-540. Descargado de <https://doi.org/10.1051/0004-6361:20031117> doi: 10.1051/0004-6361:20031117
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6, 386-408.
- Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.
- Russell, S. J., y Norvig, P. (2002). *Artificial intelligence: A modern approach* (2nd edition). Prentice Hall. Hardcover. Descargado de <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0137903952>
- Rybicki, J., Green, G. M., Rix, H.-W., El-Badry, K., Demleitner, M., Zari, E., ... Gould, A. (2021, 12). A classifier for spurious astrometric solutions in Gaia eDR3.

- Monthly Notices of the Royal Astronomical Society*, 510(2), 2597-2616. Descargado de <https://doi.org/10.1093/mnras/stab3588> doi: 10.1093/mnras/stab3588
- Sapozhnikov, S. A., Kovaleva, D. A., Malkov, O. Y., y Sytov, A. Y. (2020a, septiembre). Binary Star Population with Common Proper Motion in Gaia DR2. *Astronomy Reports*, 64(9), 756-768. doi: 10.1134/S1063772920100078
- Sapozhnikov, S. A., Kovaleva, D. A., Malkov, O. Y., y Sytov, A. Y. (2020b, octubre). VizieR Online Data Catalog: Binary Star Pop. with Common Proper Motion (Sapozhnikov+, 2020). *VizieR Online Data Catalog*, J/AZh/97/733.
- Sharma, S., Bland-Hawthorn, J., Johnston, K. V., y Binney, J. (2011, feb). Galaxia: A code to generate a synthetic survey of the milky way. *The Astrophysical Journal*, 730(1), 3. Descargado de <https://dx.doi.org/10.1088/0004-637X/730/1/3> doi: 10.1088/0004-637X/730/1/3
- Vanbeveren, D. (2001). *The influence of binaries on stellar population studies* (Vol. 264). doi: 10.1007/978-94-015-9723-4
- Zhang, J., Qian, S.-B., Wu, Y., y Zhou, X. (2019, octubre). Unbiased Distribution of Binary Parameters from LAMOST and Kepler Observations. , 244(2), 43. doi: 10.3847/1538-4365/ab442b
- Zhang, J., Qian, S. B., Wu, Y., y Zhou, X. (2020, marzo). VizieR Online Data Catalog: Binary stars parameters from LAMOST & Kepler obs. (Zhang+, 2019). *VizieR Online Data Catalog*, J/ApJS/244/43.

## Apéndice A

# Selección de la metología

Existe una variedad de metodologías en el campo de la ciencia de datos, cuyo enfoque varía dependiendo del tipo de problema que se busca resolver. Algunas de las más destacadas incluyen:

- **CRISP-DM** (Cross-Industry Standard Process for Data Mining): Provee una estructura para el desarrollo de proyectos de minería de datos, incluyendo la comprensión del negocio y los datos, preparación de los datos, modelado, evaluación y despliegue.
- **SEMMA** (Sample, Explore, Modify, Model, and Assess): Centrada en el modelado predictivo, facilita la aplicación de diversas técnicas de minería de datos para el descubrimiento de conocimiento.
- **TDSP** (Team Data Science Process): Proporciona una estructura sistemática y repetible para el desarrollo de proyectos de ciencia de datos, desde la definición del problema hasta la implementación y el mantenimiento del modelo.
- **Data Vault**: Se centra en la disponibilidad, trazabilidad, y consistencia de la información, lo que lo hace ideal para proyectos de integración de datos de múltiples fuentes.
- **Agile Data Science**: Aplica los principios del desarrollo ágil al proceso de la ciencia de datos, enfocándose en la iteración rápida, la adaptabilidad y la colaboración entre equipos interdisciplinarios.

En el presente estudio, se ha optado por la implementación de la metodología del Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD), un enfoque estructurado en la ciencia de datos para extraer conocimientos significativos de grandes volúmenes de información. En general, la metodología KDD comprende las siguientes etapas: selección de datos, preprocessamiento, transformación, minería de datos e interpretación y evaluación. Dentro del campo de la astronomía, este enfoque resulta particularmente útil debido a la vastedad de los catálogos astronómicos disponibles, permitiendo la identificación de patrones y correlaciones que enriquecen nuestra comprensión del universo.

Optamos por esta metodología debido a las siguientes consideraciones:

- **Enfoque en la extracción de conocimiento:** KDD se centra en descubrir patrones útiles y conocimientos en los datos, en lugar de simplemente construir modelos predictivos. Esta extracción de conocimiento puede ser especialmente útil en campos como la astronomía, donde los descubrimientos pueden dar lugar a nuevas teorías o la validación de las existentes.

- **Interdisciplinariedad:** KDD enfatiza la interacción entre los aspectos de la ciencia de datos y el campo de aplicación. En un proyecto de astronomía, KDD promueve la estrecha colaboración entre astrónomos y científicos de datos para entender mejor los datos y los resultados de los análisis.
- **Flexibilidad:** Aunque KDD proporciona una guía estructurada para el análisis de datos, también es flexible y permite la personalización de acuerdo a las necesidades del proyecto. Esto puede ser útil en proyectos que requieren un enfoque más adaptativo o exploratorio.
- **Manejo de grandes volúmenes de datos:** KDD fue diseñada con el propósito de manejar grandes bases de datos, lo que la hace adecuada para proyectos de ciencia de datos que implican grandes cantidades de información, como es frecuentemente el caso en la astronomía.

## Apéndice B

# Complejidad

A continuación vamos a desglosar las complejidades de los algoritmos que se utilizan en astrofísica para encontrar y categorizar estrellas vecinas y binarias. Veremos cómo estos algoritmos manejan grandes cantidades de datos y requieren recursos sustanciales para hacerlo. Nuestro objetivo es entender mejor cómo funcionan estas herramientas y cómo podrían mejorar para ser más eficientes y precisas en nuestra exploración del espacio.

En la búsqueda de vecinos:

Esta función tiene una complejidad temporal que depende de varios factores, principalmente del tamaño del conjunto de datos y de las operaciones realizadas en cada ciclo. Analicemos el código línea por línea para determinar su complejidad:

La primera línea solo imprime algunos valores y no agrega complejidad a la función.

La segunda línea realiza una operación de división y multiplicación que toma tiempo constante. En el peor de los casos, donde la longitud de coords es mayor a Nblock, la complejidad de esta línea es O(1).

La tercera línea utiliza la librería psutil para obtener información sobre el uso de memoria. Como esto no depende de los datos de entrada, podemos asumir que su complejidad es constante.

La cuarta línea establece el índice inicial y final para un subconjunto de los datos. Esto involucra un cálculo de mínimo y una operación de slicing, ambas con complejidad O(1).

La quinta línea realiza una consulta a un objeto BallTree para encontrar las estrellas que están dentro de un radio dado. La complejidad de esta operación depende de la cantidad de estrellas y el radio de búsqueda. Si tenemos  $n$  estrellas y un radio de búsqueda  $r$ , la complejidad de esta operación es de  $O(n \log n + k_1)$ , donde  $k_1$  es el número de estrellas dentro del radio de búsqueda, el cual es 5 parsec proyectado.

Las líneas seis a doce copian los valores de las variables de astrometría de las estrellas dentro del subconjunto actual. Esto es una operación de asignación y tiene una complejidad de O(1).

Las líneas 13 a 22 utilizan un bucle for para realizar operaciones en cada estrella del subconjunto actual. En cada iteración del bucle, se calcula la distancia angular y el paralaje de las estrellas vecinas, así como el error en la medición del paralaje. También se calcula la diferencia en las coordenadas de movimiento propio de cada estrella y su error. Todas estas operaciones tienen una complejidad  $O(k_1)$ , donde  $k_1$  es el número de estrellas dentro del radio de búsqueda.

La línea 23 calcula el valor de  $\mu_{\max}$ , que es una constante multiplicada por el paralaje de la estrella actual. Por lo tanto, su complejidad es  $O(1)$ .

Las líneas 24 a 27 utilizan los valores previamente calculados para determinar cuántas estrellas vecinas tienen un paralaje y movimiento propio consistentes con el de la estrella actual. Esto implica operaciones matemáticas simples y comparaciones, todas con complejidad  $O(1)$ . Sin embargo, este proceso se realiza para cada estrella del subconjunto actual, lo que implica una complejidad total de  $O(k_1^2)$ , donde  $k_1$  es el número de estrellas dentro del radio de búsqueda.

La última línea simplemente devuelve el número de estrellas vecinas que cumplen las condiciones especificadas. Como esto solo implica una operación de asignación, su complejidad es  $O(1)$ .

Por lo tanto, la complejidad total de la función `query_this_j(j)` depende del tamaño de los datos y el radio de búsqueda, pero podemos decir que es aproximadamente de  $O(n \log n + k_1^2)$ , donde  $n$  es la cantidad total de estrellas y  $k$  es el número de estrellas dentro del radio.

Selección de candidatos binarios:

Al igual que en la función anterior, la complejidad de esta función depende del tamaño de los datos y las operaciones realizadas en cada ciclo. Veamos cada línea de la función para determinar su complejidad:

La primera línea simplemente imprime algunos valores y no agrega complejidad a la función.

La segunda línea realiza una operación de división y multiplicación que toma tiempo constante. En el peor de los casos, donde la longitud de `coords` es mayor a `Nblock`, la complejidad de esta línea es  $O(1)$ .

La tercera línea utiliza la librería `psutil` para obtener información sobre el uso de memoria. Como esto no depende de los datos de entrada, podemos asumir que su complejidad es constante.

La cuarta línea establece un índice booleano para seleccionar los elementos dentro de un bloque de datos. Esto implica una operación de comparación y una de asignación, ambas con complejidad  $O(1)$ .

La quinta línea realiza una consulta a un objeto `BallTree` para encontrar las estrellas que están dentro de un radio dado. La complejidad de esta operación depende de la cantidad de estrellas y el radio de búsqueda. Si tenemos  $n$  estrellas y un radio de búsqueda  $r$ , la complejidad de esta operación es de  $O(n \log n + k_2)$ , donde  $k_2$  es el número de estrellas dentro del radio de búsqueda, el cual es una separación angular correspondiente a 1 parsec.

Las líneas seis a trece copian los valores de las variables de astrometría de las estrellas dentro del subconjunto actual. Esto es una operación de asignación y tiene una complejidad de  $O(1)$ .

Las líneas 14 a 39 utilizan un bucle `for` para realizar operaciones en cada posible pareja de estrellas dentro del subconjunto actual. En cada iteración del bucle, se calculan varias magnitudes relacionadas con la separación angular, el paralaje y la medición del movimiento propio de las dos estrellas. La complejidad de estas operaciones es  $O(k_2^2)$ , donde  $k_2$  es el número de estrellas dentro del radio de búsqueda.

La última línea simplemente devuelve dos listas con las parejas de estrellas que cumplen las condiciones especificadas. Esta operación implica una asignación y, por lo tanto, tiene una complejidad de  $O(1)$ .

Por lo tanto, podemos decir que la complejidad total de la función `query_this_j(j)` es aproximadamente de  $O(n \log n + k_2^2)$ , donde  $n$  es la cantidad total de estrellas y  $k_2$  es el número de estrellas dentro del radio de búsqueda.

Búsqueda de vecinos sobre candidatos binarios:

Al igual que en las dos funciones anteriores, la complejidad de esta función depende del tamaño de los datos y las operaciones realizadas en cada ciclo. Veamos cada línea de la función para determinar su complejidad:

La primera línea simplemente imprime algunos valores y no agrega complejidad a la función.

La segunda línea realiza una operación de división y multiplicación que toma tiempo constante. En el peor de los casos, donde la longitud de `coords_bin` es mayor a `Nblock`, la complejidad de esta línea es  $O(1)$ .

La tercera línea utiliza la librería `psutil` para obtener información sobre el uso de memoria. Como esto no depende de los datos de entrada, podemos asumir que su complejidad es constante.

La cuarta línea establece un índice booleano para seleccionar los elementos dentro de un bloque de datos. Esto implica una operación de comparación y una de asignación, ambas con complejidad  $O(1)$ .

La quinta línea realiza una consulta a un objeto `BallTree` para encontrar las estrellas que están dentro de un radio dado. La complejidad de esta operación depende de la cantidad de estrellas y el radio de búsqueda. Si tenemos  $n$  estrellas y un radio de búsqueda  $r$ , la complejidad de esta operación es de  $O(c \log c + k_3)$ , donde  $k_3$  es el número de estrellas dentro del radio de búsqueda, el cual es 5 parsec.

Las líneas seis a doce copian los valores de las variables de astrometría de las estrellas dentro del subconjunto actual. Esto es una operación de asignación y tiene una complejidad de  $O(1)$ .

Las líneas 13 a 20 utilizan un bucle `for` para realizar operaciones en cada estrella del subconjunto actual. En cada iteración del bucle, se calcula la distancia angular y el paralaje de las estrellas vecinas, así como el error en la medición del paralaje y la diferencia en las coordenadas de movimiento propio. Todas estas operaciones tienen una complejidad  $O(k_3)$ , donde  $k_3$  es el número de estrellas dentro del radio de búsqueda.

La línea 21 calcula el valor de `mu_max`, que es una constante multiplicada por el paralaje de la estrella actual. Por lo tanto, su complejidad es  $O(1)$ .

Las líneas 22 a 24 utilizan los valores previamente calculados para determinar cuántas estrellas vecinas tienen un paralaje y movimiento propio consistentes con el de la estrella actual. Esto implica operaciones matemáticas simples y comparaciones, todas con complejidad  $O(1)$ . Sin embargo, este proceso se realiza para cada estrella del subconjunto actual, lo que implica una complejidad total de  $O(k_3^2)$ , donde  $k_3$  es el número de estrellas dentro del radio de búsqueda.

La última línea simplemente devuelve un arreglo que indica cuántas estrellas vecinas cumplen las condiciones especificadas. Esta operación implica una asignación y, por lo tanto, tiene una complejidad de  $O(1)$ .

Por lo tanto, podemos decir que la complejidad total de la función `query_this_j(j)` es aproximadamente de  $O(c \log c + k_3^2)$ , donde  $c$  es la cantidad total de candidatos binarios y  $k$  es el número de estrellas dentro del radio de búsqueda.

Complejidad del cálculo del  $\Sigma_{18}$ :

En la etapa final de nuestra evaluación, el cálculo de  $\Sigma_{18}$  ha experimentado una modificación significativa. En lugar de la metodología previa, ahora hemos incorporado un modelo de Regresión basado en K Vecinos Cercanos (KNR) para realizar este cálculo. Este modelo se entrena con dos parámetros específicos: la ascensión recta y la declinación. Este cambio ha permitido que el proceso de cálculo de  $\Sigma_{18}$  se realice en un lapso de tiempo significativamente corto, apenas unos segundos.

La incorporación de un algoritmo de aprendizaje automático como el KNR en nuestra metodología introduce una nueva dimensión a la consideración de la complejidad computacional. La complejidad exacta de los algoritmos de aprendizaje automático puede variar en función de varios factores, incluyendo el número de características y el volumen de datos. En general, la complejidad del entrenamiento de un modelo de Regresión KNR se puede expresar como  $O(Dn)$ , donde  $D$  es el número de dimensiones y  $n$  es el número total de puntos de datos. En nuestro caso, dado que estamos utilizando solo dos dimensiones, este factor puede considerarse prácticamente constante.

Además, para la consulta o predicción, la complejidad del KNR es de  $O(Dnk)$ , donde  $k$  es el número de vecinos más cercanos. Como estamos considerando un  $k$  relativamente pequeño para nuestro cálculo de  $\Sigma_{18}$ , este término tampoco presenta un impacto significativo en la complejidad total.

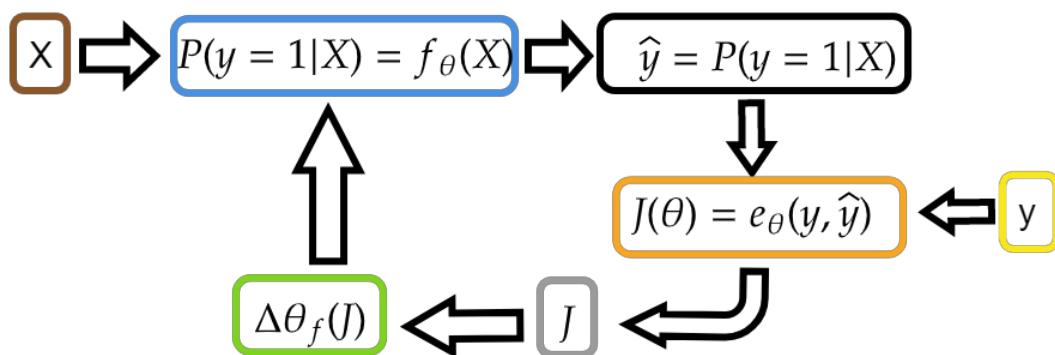
La complejidad total del algoritmo se puede expresar ahora como  $O(2n \log n + k_1^2 + k_2^2 + c \log c + Dnk)$ . Para aclarar, en este contexto,  $n$  es el número total de estrellas,  $c$  es el número de candidatos binarios,  $k_1, k_2, k_3$  son los números de estrellas dentro de los respectivos radios de búsqueda y  $Dnk$  representa la complejidad de la consulta con el modelo de Regresión KNR.

## Apéndice C

# Regresión Logística

A lo largo de nuestra investigación, hemos analizado y comparado varios modelos de machine learning para abordar la problemática de clasificación de estrellas en sistemas binarios. Sin embargo, entre todos los métodos considerados, un enfoque ha demostrado ser particularmente efectivo y robusto para nuestro conjunto de datos: la Regresión Logística. Esta técnica estadística, aunque de naturaleza simple, ha logrado proporcionar resultados precisos y consistentes, superando a otros modelos más complejos. A continuación, nos centraremos exclusivamente en el análisis y la interpretación de los resultados obtenidos a través de la Regresión Logística, explorando su potencial y discutiendo las razones de su rendimiento excepcional en nuestro estudio.

El valor esencial de nuestro modelo de Regresión Logística reside en su robustez y rendimiento aceptable incluso cuando los datos no siguen una distribución Gaussiana. Nuestro enfoque primordial es la funcionalidad y efectividad del modelo al enfrentar nuevos datos, los cuales no se utilizaron durante la estimación de parámetros. Es importante subrayar que nuestro mecanismo de validación es el conjunto de prueba. Aunque las hipótesis teóricas sobre las que se construye el modelo son importantes, no realizamos un control estricto para verificar si los datos siguen efectivamente la distribución Gaussiana. Lo que nos importa en última instancia es que el modelo funcione de manera efectiva con nuevos conjuntos de datos.



**Figura C.1:** Visualización del proceso iterativo de ajuste en la Regresión Logística

La Figura C.3 ilustra el proceso iterativo de ajuste inherente a la Regresión Logística. En este diagrama, los datos de entrada son representados por el nodo marrón 'X'. Estos datos se introducen en el modelo de Regresión Logística, que se representa con un rectángulo azul. Posteriormente, se calcula la función de costo, identificada con el nodo naranja. Finalmente, la optimización del modelo se realiza a través del

optimizador, simbolizado por el rectángulo verde, que busca minimizar la función de costo y perfeccionar el modelo de Regresión Logística.

Cabe mencionar que los datos de entrada se definen como:  $(X_i, y_i), 0 < i < N - 1$

Los datos que manejamos proporcionan las clasificaciones necesarias. Recibiremos un dato de entrada  $x$ , que será procesado por un modelo y mapeado a una estimación de la salida  $\hat{Y}$ , utilizando una función definida parametrizada por  $\theta$ . En base a la salida del modelo y a las anotaciones (la salida deseada), evaluaremos el modelo. Para esto, buscamos una medida de similitud entre la salida del modelo  $\hat{Y}$  y las anotaciones  $y$ . Esta medida de similitud es la función de costo. A menor diferencia entre  $\hat{Y}$  e  $y$ , menor será la función de costo, proporcionando una medida del error. Finalmente, contamos con un optimizador que busca minimizar la función de costo mediante la modificación de los valores de  $\theta$ .

La regresión logística se define de la siguiente manera:

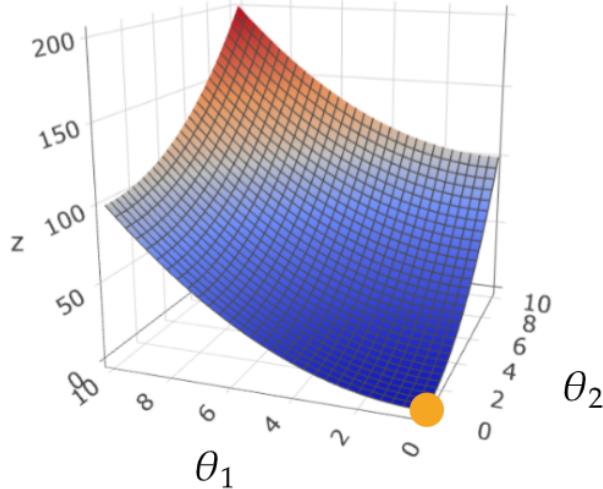
$$P(y = 1|X) = \frac{1}{1 + e^{-(\vec{\theta}^T \cdot \vec{x} - \theta_0)}} \quad (\text{C.1})$$

Esta ecuación describe la probabilidad de que  $y = 1$  dado  $X$ , en función de los parámetros del modelo  $\theta$ ,  $\theta_0$ , y las características de entrada  $X$ .

En la regresión logística, los parámetros  $\theta$  se determinan utilizando el método de máxima verosimilitud. En esencia, este método busca encontrar los parámetros  $\theta$  que maximicen la probabilidad (o verosimilitud) de los datos observados dados estos parámetros. En la práctica, esto a menudo implica iterar sobre diferentes valores de  $\theta$  hasta encontrar aquellos que proporcionen el mejor ajuste a los datos, un proceso que a menudo se realiza a través de un algoritmo de optimización como el descenso de gradiente.

Por otro lado, en el análisis discriminante lineal (LDA), los parámetros  $\theta$  se calculan directamente a partir de los datos. LDA asume que los datos de cada clase se distribuyen normalmente y tienen la misma varianza (o covarianza en el caso multivariante). Con base en estas suposiciones, LDA estima los parámetros de las distribuciones Gaussianas de cada clase a partir de los datos y utiliza estas estimaciones para calcular los parámetros  $\theta$ .

Consideremos la función de costo  $J(\theta)$ , que se define en el espacio de parámetros, específicamente en términos de  $\theta_1$  y  $\theta_2$ . El objetivo principal en el proceso de optimización es encontrar el punto mínimo de esta función, es decir, el conjunto de parámetros que minimizan el error de predicción. Recordemos que definimos el error como la discrepancia entre las anotaciones verdaderas  $y$  y las predicciones de nuestro modelo  $\hat{y}$ .



**Figura C.2:** Visualización en 3D de la función  $z(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$ . El círculo naranja denota el punto de mínimo global en el espacio de parámetros  $(\theta_1, \theta_2)$ , el lugar donde se minimiza la función de costo.

El modelo en estudio se describe mediante la ecuación:

$$P(y=1|X) = \frac{1}{1 + e^{-(\vec{\omega}^T \cdot x + \omega_0)}} \quad (\text{C.2})$$

En este caso, los parámetros relevantes son  $(\vec{\omega}, \omega_0)$ .

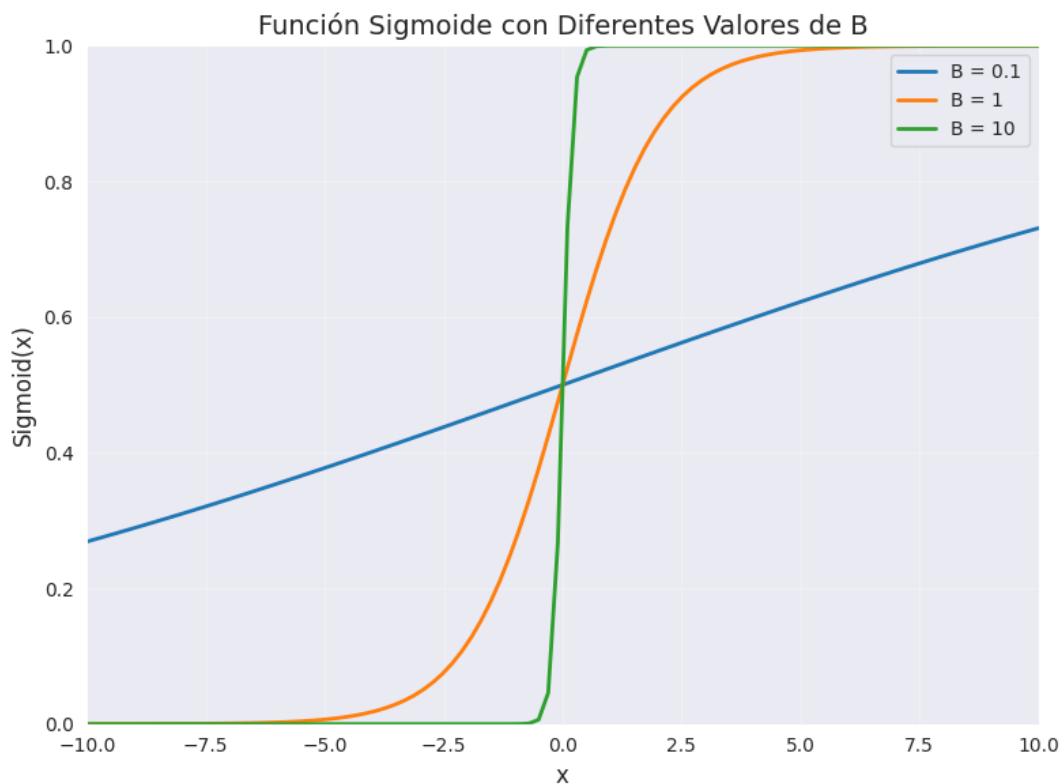
### C.1. Equilibrio entre confianza y volumen de datos

La determinación de la clasificación mediante este modelo ocurre cuando la probabilidad evaluada alcanza el valor de 0.5. Este punto se conoce como el umbral de decisión.

Además, este modelo incluye una característica interesante relacionada con la temperatura,  $T$ . Se define un factor  $B = \frac{1}{T}$ . Aquí, la temperatura representa una medida de la incertidumbre del modelo. Si incrementamos  $T$ , el valor de  $B$  disminuye, lo que indica una mayor incertidumbre en las predicciones del modelo. En otras palabras, al aumentar la temperatura, el modelo se torna menos "seguro".

En el gráfico presentado, se observa que al incrementar el valor de  $B$  (equivalente a disminuir la temperatura), la confianza del modelo aumenta, evidenciado por una transición más abrupta entre las clases en la función sigmoidal.

No obstante, en el caso de la regresión logística, existe una posible problemática cuando los datos son linealmente separables. En estas circunstancias, el modelo puede de volverse extremadamente confiado, provocando que tanto  $\omega_0$  como la norma de  $\vec{\omega}$  tiendan al infinito,  $\lim_{\omega_0 \rightarrow \infty}$  y  $\lim_{|\omega| \rightarrow \infty}$ , respectivamente. Esta situación podría llevar a una confianza excesiva del modelo en sus predicciones, y problemas al trabajar con numeros excesivamente altos. Una estrategia comúnmente adoptada para contrarrestar este comportamiento es la regularización.



**Figura C.3:** Comparación de la función sigmoide para diferentes valores del parámetro  $B$ : En la gráfica se muestran tres curvas de la función sigmoide, correspondientes a  $B = 0,1$  (en azul),  $B = 1$  (en naranja) y  $B = 10$  (en verde). Se puede observar cómo el valor de  $B$  influye en la transición entre clases y, por ende, en la confianza del modelo en sus predicciones.

La regularización introduce un término de penalización en la función de costo, que incrementa conforme el valor de los parámetros aumenta. Este mecanismo de control efectivamente evita que los parámetros del modelo crezcan desmedidamente, siendo particularmente útil en escenarios de alta dimensionalidad o ante presencia de correlación entre las características. Dentro de Scikit-learn ([Pedregosa y cols., 2011](#)), se implementan diversos tipos de regularización, tales como L1 (Lasso) y L2 (Ridge). La elección entre estos depende de la naturaleza específica del problema y de los datos.

Adicionalmente, en situaciones donde los datos son linealmente separables, otra alternativa viable es emplear un clasificador de vectores de soporte (SVM, por sus siglas en inglés) con un kernel lineal. Esta opción puede ser más adecuada para tratar dichos escenarios, proporcionando un manejo más efectivo y robusto del modelo.

## C.2. Binary Cross-Entropy

Uno de los aspectos más críticos en la implementación de este modelo es la elección de una función de costo adecuada para optimizar los parámetros del modelo. Una de las funciones de costo más utilizadas en este contexto es la Entropía Cruzada Binaria, también conocida como pérdida logarítmica. Esta función, cuando se aplica a la regresión logística, tiene propiedades atractivas que la hacen particularmente efectiva para la modelación de datos de naturaleza binaria. A continuación, exploraremos a fondo la Entropía Cruzada Binaria en el contexto de la Regresión Logística, describiendo su formulación matemática y discutiendo sus propiedades y ventajas.

$$\mathcal{L} = P(y = 1 | X_1) \cdot P(y = 1 | X_2) \cdot (1 - P(y = 1 | X_3)) \cdot P(y = 1 | X_4) \cdot (1 - P(y = 1 | X_5)) \rightarrow 1 \quad (\text{C.3})$$

$$\mathcal{L} = \prod_{i=1}^N \left[ \log P(y = 1 | X_i)^{y_i} \cdot (1 - P(y = 1 | X_i))^{1-y_i} \right] \quad (\text{C.4})$$

$$\log(\mathcal{L}) = \frac{1}{N} \sum_{i=1}^N [y_i \cdot \log P(y = 1 | X_i) + (1 - y_i) \cdot \log(1 - P(y = 1 | X_i))] \quad (\text{C.5})$$

$$\mathcal{L}(\omega, \omega_0) = -\frac{1}{N} \sum_{i=1}^N \left[ y_i \log \left( \frac{1}{1 + e^{-(X_i^T \vec{\omega} + \omega_0)}} \right) + (1 - y_i) \log \left( 1 - \frac{1}{1 + e^{-(X_i^T \vec{\omega} + \omega_0)}} \right) \right] \quad (\text{C.6})$$

En el modelo de regresión logística, la verosimilitud  $\mathcal{L}$  se define como el producto de las probabilidades predichas para las clases positivas y la complementaria para las clases negativas de cada observación, como se puede observar en la ecuación C.3. El objetivo de la maximización de la verosimilitud es hacer que este producto sea lo más cercano posible a uno. Sin embargo, a medida que aumenta el número de observaciones, el producto tiende a cero debido a la naturaleza de la multiplicación de probabilidades, lo cual complica el problema de optimización.

Para solucionar este inconveniente, se aplica el logaritmo a la verosimilitud en la ecuación C.5, transformando el producto en una suma. Este enfoque simplifica la maximización a un problema de optimización más manejable. Aún así, trabajar con

log-verosimilitud puede depender del número de observaciones. Para evitar esta dependencia, normalizamos por el número de observaciones,  $N$ , obteniendo un promedio en lugar de una suma, como se observa en la tercera ecuación.

Finalmente, dado que estamos en un contexto de aprendizaje supervisado, deseamos comparar las predicciones del modelo con los datos verdaderos. Para ello, se utiliza la entropía cruzada binaria. Al multiplicar la log-verosimilitud por  $-1$ , transformamos el problema de maximización de la verosimilitud en un problema de minimización de la entropía cruzada binaria. La ecuación C.6 muestra este promedio de las entropías cruzadas para cada observación, que es la función de pérdida que queremos minimizar en el proceso de entrenamiento del modelo.

Recordemos que la entropía cruzada se define por:

$$H(p, q) = H(p) + D_{KL}(p||q) \quad (\text{C.7})$$

donde

- $H(p)$  es la entropía de  $p$  y mide la incertidumbre en la distribución de probabilidad  $p$ .
- $D_{KL}(p||q)$  es la divergencia de Kullback-Leibler de  $p$  con respecto a  $q$ , y mide cuánto difiere la distribución de probabilidad  $p$  de una distribución de referencia  $q$ .

La entropía cruzada  $H(p, q)$ , expresada en la ecuación, es una medida clave en el contexto del aprendizaje supervisado, particularmente en la clasificación binaria. Se puede descomponer en dos componentes: la entropía  $H(p)$  y la divergencia de Kullback-Leibler  $D_{KL}(p||q)$ .

La entropía  $H(p)$  cuantifica la incertidumbre inherente en la distribución de probabilidad de los datos,  $p$ . En el caso ideal de datos determinísticos, la distribución es perfectamente predecible, y la entropía es cero. Sin embargo, en la práctica, los datos suelen presentar alguna forma de incertidumbre, y la distribución puede variar de forma significativa, de  $(0,1)$  a  $(1,0)$  en el caso binario.

La divergencia de Kullback-Leibler,  $D_{KL}(p||q)$ , por otro lado, mide cuánto difiere nuestra distribución de probabilidad modelo  $q$  de la distribución de probabilidad verdadera  $p$ . En términos de optimización, queremos minimizar esta divergencia. Cuando el modelo refleja perfectamente la verdadera distribución de los datos, la divergencia de Kullback-Leibler será cero, indicando que nuestro modelo ha convergido.

En términos prácticos, utilizamos la entropía cruzada binaria como nuestra función de costo en la regresión logística, que es simplemente  $H(p, q)$  en el caso de la clasificación binaria. Sin embargo, es importante tener en cuenta que si los datos no son linealmente separables, es improbable que  $D_{KL}(p||q)$  llegue a cero. Además, si  $D_{KL}(p||q)$  es cero, esto implica que nuestra incertidumbre sobre los datos es muy alta, lo cual es un indicador de que nuestro modelo puede no estar funcionando correctamente.

### C.3. Optimizadores

En la regresión logística, la entropía cruzada binaria se utiliza comúnmente para cuantificar la discrepancia entre las predicciones del modelo y los valores verdaderos de la variable objetivo. Esta función de costo presenta ciertas propiedades que favorecen la eficiencia de los algoritmos de optimización.

En primer lugar, estas funciones de costo son convexas. Esta característica asegura la existencia de un mínimo global hacia el que los algoritmos de optimización pueden converger. Esto simplifica en gran medida el proceso de optimización, ya que evita los mínimos locales que podrían atrapar el proceso de optimización.

En segundo lugar, estas funciones de costo tienen regiones "planas", también conocidas como plateaus, donde la derivada se acerca a cero. Finalmente, es importante tener en cuenta que estas funciones de costo son sensibles al rango de las entradas. Esta sensibilidad puede ser una ventaja, ya que proporciona una mayor discriminación entre las entradas, pero también puede presentar desafíos si las entradas no están correctamente normalizadas o si varían ampliamente en magnitud.

Uno de los métodos de optimización que ha demostrado ser muy eficaz es el método de Newton CG (Conjugate Gradient). Este método es una variante del método de Newton que se utiliza para minimizar funciones de varias variables. En lugar de utilizar la matriz Hessiana completa, como lo hace el método de Newton estándar, Newton CG utiliza aproximaciones de la Hessiana para ahorrar en costos computacionales.

El método de Newton CG tiene varias ventajas sobre otros métodos de optimización disponibles en el paquete de regresión logística de sklearn ([Pedregosa y cols., 2011](#)). En primer lugar, Newton CG tiene una convergencia cuadrática en lugar de la convergencia lineal de otros métodos como el descenso de gradiente. Esto significa que Newton CG puede converger al mínimo de la función de costo más rápido que estos otros métodos. Además, Newton CG es menos sensible a la escala de las características, lo que lo hace más robusto frente a la necesidad de normalizar las entradas.

Un punto crucial para destacar es que el rendimiento de Newton CG puede variar dependiendo del problema específico y de la inicialización del modelo. Sin embargo, si se realiza una búsqueda exhaustiva de los parámetros utilizando una técnica como la búsqueda en malla (grid search), se puede determinar que el método de Newton CG es la opción más eficiente para la regresión logística. Este resultado respalda la elección de este método como la principal técnica de optimización en la práctica de la regresión logística.

## Apéndice D

# Cortes Astrofísicos

A continuación se mencionarán los cortes astrofísicos propuestos por K. El-Badry en ([El-Badry y Rix, 2018](#)).

### D.1. Candidatos binarios

Se definirán como candidatos a binarios aquellos pares de estrellas que satisfagan los siguientes criterios:

- *Distancia proyectada menor a 1 parsec:* la separación angular entre ambas estrellas, denominada  $\theta$ , debe cumplir con

$$\frac{\theta}{\text{arcsec}} \leq 206,265 \times \frac{\omega_1}{\text{mas}}, \quad (\text{D.1})$$

donde  $\omega_1$  es la paralaje de la estrella más brillante en magnitud  $G$ . Elegimos un radio máximo de búsqueda de 1 pc (correspondiente a un período orbital de alrededor de  $10^8$  años), ya que es improbable encontrar binarios ligados más allá de esta distancia, donde el campo de marea galáctico se equipara con la atracción gravitacional entre las estrellas. La distancia a partir de la cual el campo de marea galáctico supera la aceleración interna de un binario se conoce como radio de Jacobi. En las proximidades del sistema solar, este radio se estima como  $r_J \approx 1,35 \text{ pc} \times (M_{\text{tot}}/M_{\odot})$ , donde  $M_{\text{tot}}$  es la masa total del binario. A separaciones ligeramente inferiores a  $r_J$ , los binarios son alterados eficientemente por influencias gravitacionales de entidades como otras estrellas y nubes moleculares.

- *Paralajes consistentes en un rango de 3 (o 6) sigma:* los paralajes de ambos componentes,  $\omega_1$  y  $\omega_2$ , deben respetar

$$|\omega_1 - \omega_2| < b \sqrt{\sigma_{\omega,1}^2 + \sigma_{\omega,2}^2}, \quad (\text{D.2})$$

donde  $\sigma_{\omega,i}$  es la incertidumbre en la paralaje del componente  $i$ , y  $b = 3$  para pares con  $\theta > 4$  segundos de arco, o  $b = 6$  para pares con  $\theta < 4$  segundos de arco. El corte menos restrictivo en  $\theta < 4$  segundos de arco se elige debido a la baja probabilidad de alineación accidental y las subestimaciones significativas en las incertidumbres de la paralaje en las separaciones angulares cercanas.

- *Movimientos propios congruentes con una órbita kepleriana:* Las dos estrellas en un binario amplio mostrarán movimientos propios similares, pero no serán exactamente iguales debido al movimiento orbital. Exigimos que

$$\Delta\mu \leq \Delta\mu_{\text{orbit}} + 2\sigma_{\Delta\mu}, \quad (\text{D.3})$$

donde  $\Delta\mu$  es la diferencia observada de los movimientos propios escalares,  $\sigma_{\Delta\mu}$  es su incertidumbre, y  $\Delta\mu_{\text{orbit}}$  es la máxima diferencia de movimiento propio esperada por el movimiento orbital. Las dos primeras cantidades se obtienen como:

$$\Delta\mu = \left[ (\mu_{\alpha,1}^* - \mu_{\alpha,2}^*)^2 + (\mu_{\delta,1} - \mu_{\delta,2})^2 \right]^{1/2},$$

$$\sigma_{\Delta\mu} = \frac{1}{\Delta\mu} \left[ \left( \sigma_{\mu_{\alpha,1}^*}^2 + \sigma_{\mu_{\alpha,2}^*}^2 \right) \Delta\mu_{\alpha}^2 + \left( \sigma_{\mu_{\delta,1}}^2 + \sigma_{\mu_{\delta,2}}^2 \right) \Delta\mu_{\delta}^2 \right]^{1/2},$$

donde  $\Delta\mu_{\alpha}^2 = (\mu_{\alpha,1}^* - \mu_{\alpha,2}^*)^2$  y  $\Delta\mu_{\delta}^2 = (\mu_{\delta,1} - \mu_{\delta,2})^2$ . En esta ecuación,  $\mu_{\alpha,i}^* \equiv \mu_{\alpha,i} \cos \delta_i$ , donde  $\alpha$  y  $\delta$  son la ascensión recta y la declinación, respectivamente, y  $\mu_{\alpha}$  y  $\mu_{\delta}$  representan los movimientos propios en las direcciones de la ascensión recta y la declinación.

$$\Delta\mu_{\text{orbit}} = 0,44 \text{ mas yr}^{-1} \times \left( \frac{\omega}{\text{mas}} \right)^{3/2} \left( \frac{\theta}{\text{arcsec}} \right)^{-1/2}$$

## D.2. Búsqueda de vecinos cercanos

Depuramos la lista preliminar de candidatos a binarios a través de varias fases. Primero, empezamos con todas las fuentes recolectadas a partir de nuestra consulta ADQL inicial. Para cada fuente, calculamos el número de vecinos en el espacio de fase que superen una luminosidad de  $G = 18$  y sean compatibles con las características de tamaño y dispersión de velocidad de un cúmulo estándar. Definimos a los vecinos como aquellos que cumplen con las siguientes condiciones:

- Tienen una separación proyectada menor a 5 pc, o sea,  $\theta \leq 17,19 \text{ arcmin} \times (\omega/\text{mas})$ .
- Sus movimientos propios están dentro de  $5 \text{ km s}^{-1}$ , lo que equivale a una diferencia de movimiento propio  $\Delta\mu \leq 1,05 \text{ mas yr}^{-1} \times (\omega/\text{mas})$ , permitiendo una tolerancia de  $2\sigma$ .
- Sus paralajes son consistentes en el margen de 2 sigma, lo cual se traduce en  $\Delta\omega \leq 2\sqrt{\sigma_{\omega,1}^2 + \sigma_{\omega,2}^2}$ .

Finalmente, descartamos de nuestra selección de posibles binarios todos los pares en los que uno de los componentes tenga más de 30 vecinos conforme a la definición anterior.

## D.3. Búsqueda de vecinos sobre candidatos binarios

En esa parte del algoritmo, se procede con la eliminación de todos los conjuntos de estrellas superpuestos. Esto significa que si uno de los miembros de un par candidato a ser binario forma parte de otro par también candidato, ambos pares son excluidos. De esta manera, se filtran los tríos genuinos resueltos, que son efectivamente identificados en nuestra búsqueda inicial y cuya presencia no es inusual.

En la etapa final, buscamos identificar y eliminar miembros de pequeños conglomerados o grupos móviles que pudieran haber pasado desapercibidos en la primera

etapa de depuración. Asignamos las coordenadas en el espacio de fase del componente más luminoso de cada par para representar dicho par y procedemos a contar el número de pares adyacentes para cada candidato, utilizando los mismos tres criterios previamente empleados para contar las fuentes adyacentes (sin hacer ningún corte de magnitud). Eliminamos todos los candidatos que tengan más de un par adyacente.

Es relevante mencionar que durante este proceso de filtrado de tríos resueltos, cúmulos y grupos móviles, se perderán algunos pares binarios reales. Sin embargo, se puede estimar un límite superior de aproximadamente un 15 % en la fracción de binarios auténticos que se perderán durante este proceso de depuración ([El-Badry y Rix, 2018](#)).