UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Repositorio Digital USM **https://repositorio.usm.cl**

Tesis USM TESIS de Pregrado de acceso ABIERTO

2019

AUTOMATIZACIÓN DEL PROCESO DE DETECCIÓN Y CLASIFICACIÓN DE ESTRELLAS VARIABLES DE LARGO PERÍODO

HORTH MEDINA, PATRICIO IGNACIO

https://hdl.handle.net/11673/46760

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAISO - CHILE

“AUTOMATIZACIÓN DEL PROCESO DE DETECCIÓN Y CLASIFICACIÓN DE ESTRELLAS VARIABLES DE LARGO PERÍODO”

PATRICIO HORTH MEDINA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Mauricio Araya
Profesor Correferente: Hector allende

Mayo - 2019

RESUMEN

Resumen— Las estrellas variables han sido estudiadas ampliamente durante las últimas décadas, logrando grandes avances que han ayudado a entender la evolución estelar y generar una taxonomía bien definida para estas. Sin embargo, existen estrellas variables de baja amplitud que sólo son identificables en el largo período. Las cuales han sido poco estudiadas, debido a las dificultades de adquisición y procesamiento de series tiempo a escalas temporales altas, quedando así una gran cantidad de información sin estudiar y la interrogante de por qué ocurre este comportamiento. Por otro lado, el uso de técnicas de aprendizaje automático ha sido ampliamente aceptado para el estudio de estrellas variables de corto periodo automatizando su clasificación y facilitando el trabajo de astrónomos y astrofísicos. Por lo tanto, se espera que este tipo de técnicas puede ser de gran utilidad para ayudar a estudiar este comportamiento en largo periodo. En este trabajo se propone un proceso que automatice la clasificación de estrellas variables de largo periodo. Este proceso consta de dos grandes etapas, donde la primera de ellas consiste en identificar las estrellas que presenten este comportamiento, mientras que la segunda parte, consiste en realizar una clasificación en base al comportamiento de las diversas características que puedan tener. Los algoritmos de clustering pueden ser de utilidad para realizar esta clasificación, encontrando patrones según el comportamiento de estas. Utilizando el algoritmo K-means y considerando como base las características que son utilizadas para la elaboración del diagrama H-R, se consiguió encontrar un total de 6 grupos claramente definidos.

Palabras Clave— Estrellas variables de largo periodo; Aprendizaje automático; Clustering; Análisis de componentes principales; K-means;

ABSTRACT

Abstract— Variable stars have been studied extensively during the last decades, achieving great advances that have helped to understand stellar evolution and generate a well-defined taxonomy for these. However, there are variable stars of low amplitude that are only identifiable in the long period. Which have been little studied, due to the difficulties of acquisition and processing of time series at high temporal scales, thus leaving a large amount of information without studying and the question of why this behavior occurs. On the other hand, the use of machine learning techniques has been widely accepted for the study of short-period variable stars by automating their classification and facilitating the work of astronomers and astrophysicists. Therefore, it is expected that this type of techniques can be very useful to help study this behavior in a long period. In this work a process is proposed that automates the classification of variable stars of long period. This process consists of two major stages, where the first consists of identifying the stars that exhibit this behavior, while the second part consists of making a classification based on the behavior of the various characteristics that they may have. Clustering algorithms can be useful to perform this classification, finding patterns according to the behavior of these. Using the K-means algorithm and considering as a basis the characteristics that are used for the elaboration of the H-R diagram, it was possible to find a total of 6 clearly defined groups.

Keywords— Long-term variable stars; Machine learning; Principal component analysis; K-means

GLOSARIO

ASAS: All sky automated survey. Proyecto astronómico dedicado al estudio de estrellas variables.
CSS: Catalina sky survey. Proyecto astronómico dedicado a la deteccion de asteroides y objetos cercanos a la tierra.
PCA: Principal component analysis. Método estadístico utilizado para la reducción de dimensiones y selección de características.
t-SNE: t-distributed stochastic neighbor embedding. Método estadístico utilizado para la visualización de datos en altas dimensiones.
VOSA: Virtual observatory Sed Analyzer. Observatorio virtual enfocado en la obtención teórica de modelos para la estimación de parámetros físicos y comparación de resultados fotométricos, entre otras funciones.

ÍNDICE DE CONTENIDOS

RESUMEN III

ABSTRACT IV

GLOSARIO V

ÍNDICE DE FIGURAS IX

ÍNDICE DE TABLAS X

INTRODUCCIÓN 1

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 2
1.1 Antecedentes 2
1.2 Objetivos de la memoria 3
1.2.1 Objetivo general 3
1.2.2 Objetivos específicos 4

CAPÍTULO 2: MARCO CONCEPTUAL ASTRONÓMICO 5
2.1 Series de tiempo 5
2.2 Estrellas variables 6
2.3 Diagrama Hertzsprung- Russell 8
2.4 All Sky Automated Survey 9
2.5 Cross-Matching 9
2.6 Observatorios Virtuales 11

CAPÍTULO 3: MARCO CONCEPTUAL: CIENCIAS DE LA COMPUTACIÓN 12
3.1 Aprendizaje supervisado 12
3.1.1 k vecinos más cercanos (KNN) 13
3.1.2 Arboles de decisión 13
3.1.3 Support vector machines (SVM) 14
3.2 Aprendizaje semi supervisado 14
3.3 Clustering 15
3.3.1 Funciones de proximidad 15
3.3.2 K-Means Clustering 17
3.3.3 Expectation-Maximization clustering 17
3.3.4 Clustering jerárquico aglomerativo 20
3.3.5 DBSCAN 21
3.4 Medidas de desempeño 23
3.4.1 Índice de Calinski-Harabasz 23
3.4.2 Índice de Davies-Bouldin 24
3.4.3 Índice Silhouette 25

3.5 Reducción de dimensionalidad . . . 26
3.5.1 Maldición de la dimensionalidad . . . 26
3.5.2 Análisis de componentes principales . . . 27
3.5.3 t-distributed stochastic neighbor embedding (t-SNE) . . . 29
3.6 Metodología CRISP-DM . . . 31

CAPÍTULO 4: PROPUESTA DE SOLUCIÓN . . . 33
4.1 Parte 1: Detección de estrellas variables de largo periodo . . . 33
4.1.1 Obtención de datos . . . 34
4.1.2 Preprocesamiento . . . 35
4.1.3 Extracción de características . . . 36
4.1.4 Clasificación . . . 38
4.2 Parte 2: Clasificación de las estrellas variables de largo periodo en base a sus propiedades físicas . . . 39
4.2.1 Cross-matching . . . 39
4.2.2 Selección de características y reducción de dimensionalidad . . . 40
4.2.3 Clustering . . . 41

CAPÍTULO 5: RESULTADOS EXPERIMENTALES . . . 43
5.1 Consideraciones previas . . . 43
5.1.1 Ambiente de desarrollo . . . 43
5.1.2 Datos Utilizados . . . 43
5.2 Clasificación . . . 44
5.3 Selección de características y reducción de dimensionalidad . . . 45
5.3.1 Selección de características . . . 46
5.3.2 Reducción de dimensionalidad . . . 48
5.4 Clustering . . . 50
5.4.1 DBSCAN . . . 50
5.4.2 K-means . . . 52
5.4.3 Clustering jerárquico aglomerativo . . . 53
5.4.4 Elección del algoritmo a utilizar . . . 53
5.5 Grupos encontrados . . . 55
5.5.1 Cluster 1 . . . 57
5.5.2 Cluster 2 . . . 58
5.5.3 Cluster 3 . . . 59
5.5.4 Cluster 4 . . . 60
5.5.5 Cluster 5 . . . 60
5.5.6 Cluster 6 . . . 61

CAPÍTULO 6: VALIDACIÓN DE LA SOLUCIÓN . . . 64
6.1 Opinión experta sobre resultados obtenidos . . . 64
6.2 Preguntas hacia las expertas . . . 65
6.3 Conclusiones sobre las observaciones . . . 66

CAPÍTULO 7: CONCLUSIONES 67
7.1 Consideraciones para el trabajo futuro 68
7.2 Conclusiones generales 69

ANEXOS 70
7.1 Algoritmos de clasificación supervisada 70
7.1.1 Arboles de decisión 70
7.1.2 Support vector machines 71
7.2 Código y jupyter notebooks 73

REFERENCIAS BIBLIOGRÁFICAS 74

ÍNDICE DE FIGURAS

1 Taxonomía de las estrellas variables. 6

2 Curva de de luz de una estrella púlsate tipo Mira. 7

3 Curva de de luz de una estrella binaria eclipsante. 7

4 Diagrama H-R mostrando la diversidad de estrellas de la Vía Láctea. 8

5 Ejemplo de DBSCAN. 22

6 Ciclo de vida de la metodología CRISP DM. 31

7 Pipeline completo del proceso propuesto. 33

8 Extracto de uno de los datasets de la estrella 045914-6935.8 35

9 Algoritmos de clustering disponibles en scikit learn 41

10 Diagrama de la esfera celeste con los campos utilizados. 44

11 Diagrama H-R para las estrellas variables de largo periodo 45

12 Matriz de correlaciones usando las variables preseleccionadas 47

13 Varianza explicada acumulada e independiente por componente principal 49

14 Porcentaje de cada variable por cada componente principal 49

15 Visualización de los datos usando t-SNE 50

16 Gráficas comparativas para el valor del parámetro épsilon, tomando en cuenta el índice de silhouette y la cantidad de clusters encontrados por el algoritmo. 51

17 Gráfica comparativas para los algoritmos K-menas y K-medians. 52

18 Clustering jerárquico aglomerativo con diferentes métricas de distancia y estrategias de vinculación 53

19 Distribución de los grupos encontrados 55

20 Boxplots para cada característica por cluster 56

21 Valores promedio para cada característica por cluster. 56

22 Gráficos para cluster 1 57

23 Gráficos para cluster 2 58

24 Gráficos para cluster 3 59

25 Gráficos para cluster 4 60

26 Gráficos para el cluster 5 61

27 Gráficos para cluster 6 62

28 Diagrama que muestra la aplicación del algoritmo SVM. 71

ÍNDICE DE TABLAS

1 Estructuras de datos usadas en para hacer cross-matching 10

2 Funciones de distancia 16

3 Valores índice de silhouette 26

4 Estructura de los datasets entregados por ASAS 34

5 Criterios de clasificación 39

6 Librerías utilizadas. 43

7 Características disponibles 46

8 Errores para las variables radio y masa 47

9 Resultados obtenidos por DBSCAN 51

10 Comparación entre algunos algoritmos de clustering 54

INTRODUCCIÓN

A través del tiempo el estudio de las estrellas variables ha permitido generar una gran cantidad de información que ha sido de utilidad para la comprensión de nuestra galaxia y el universo, permitiendo descubrir propiedades de cuerpos celestes que de otra forma hubiese sido difícil conseguir. Sin embargo hay un tipo de estrella variable que ha sido poco estudiada debido a que su variabilidad es percibible en el largo periodo y a baja amplitud, desconociendo los motivos que pueden generar este comportamiento y las implicancias que estas pueden tener para la astrofísica y astronomía.

Esta memoria se propone un proceso basado en la adaptación de la metodología CRISP-DM, el cual consta de dos grandes partes: La identificación de estrellas variables de largo periodo y la clasificación de estas en base a las diversas propiedades físicas que puedan tener. Particularmente, si bien se propone el proceso completo, el desarrollo de esta memoria esta enfocado principalmente en la resolución de la segunda parte mediante algoritmos de aprendizaje no supervisado, donde se buscan patrones en base a las diversas características presentes en las estrellas variables de largo periodo con el fin de comprender qué es lo que las lleva a presentar este comportamiento.

Dentro de los capítulos presentes en este documento, podemos encontrar la definición del problema y los objetivos que persigue esta memoria en el capitulo 1. El marco teórico utilizado desde el punto de vista astronómico se expone en el capitulo 2, mientras que el marco teórico referente a las ciencias de la computación se expone en el capitulo 3. Por otro lado las diferentes etapas que conforman el proceso propuesto junto con sus objetivos se exponen en el capitulo 4. Los resultados obtenidos producto de aplicar el pipeline completo del proceso propuesto son detallados en el capitulo 5, mientras que en el capitulo 6 se exponen las opiniones de dos astrónomos acerca de los resultados obtenidos , lo que servirá para la validación de este trabajo. Finalmente en el capítulo 7 se presentan las conclusiones y aprendizajes obtenidos tras la realización de este trabajo.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

1.1. Antecedentes

El estudio de las estrellas en astronomía y astrofísica es de vital importancia ya que estas son consideradas el principal motor de la evolución cósmica, particularmente para la creación de elementos más pesados que el hidrógeno y el helio, los cuales están presentes prácticamente en todo nuestro entorno. En particular el estudio de las estrellas variables ha sido bastante fértil en las últimas décadas, permitiendo obtener información acerca de propiedades estelares importantes, como son la estructura interna y externa de una estrella, su radio, temperatura, composición, brillo, y evolución. Así también se han logrado realizar clasificaciones dependiendo de las características presentes en cada estrella variable, dividiéndolas en dos grandes categorías y diversos subtipos. Muchas de estas propiedades hubiesen sido muy difíciles de obtener mediante otros métodos ya que generalmente la naturaleza de la variabilidad de la estrella proporciona las respuestas a los diversos fenómenos del universo, como, por ejemplo, el estudio de las supernovas ha ayudado a comprender que la expansión del universo se está acelerando, mientras que el estudio de las variables cefeidas[1]. ha jugado un papel importante en la determinación de las distancias entre galaxias y la edad del universo.

Por otra parte, en las últimas décadas se han hecho descubrimientos en el comportamiento de estrellas previamente catalogadas como constantes, al ser estudiadas en el largo período, observando variaciones en sus curvas de luz a una baja amplitud, abriendo así la posibilidad de encontrarnos frente a un nuevo tipo de estrella variable. Lo anterior hace probable el descubrimiento de nuevas propiedades o características que contribuyan al entendimiento de los fenómenos presentes en el universo. Una primera aproximación sobre el comportamiento de las estrellas variables de largo periodo fue descrito en el trabajo realizado por Nikolaus Vogt [42], el cual se llevo a cabo usando una muestra de datos de 216 curvas de luz catalogadas como constantes seleccionadas de forma aleatoria provistas por el observatorio de Sonneberg. En este estudio se descubrió que en grandes períodos de tiempo para la escala humana (1000 -8000 días), un total de 17 estrellas se comportaban como variables en una muy baja amplitud, las cuales representaban aproximadamente el 7 % de la muestra total. Esta investigación expuso la existencia del comportamiento variable en el brillo de las estrellas en el largo periodo, sin embargo, este se realizo para una cantidad muy reducida de datos y de forma manual sin el apoyo de métodos computacionales que permitieran hacer el análisis escalable para una cantidad mayor de estrellas.

En la actualidad gracias a los avances tecnológicos, el perfeccionamiento técnicas en el uso de sensores CCD y los diversos proyectos astronómicos que mapean todo el cielo disponible

[1]Tipo de estrella variable que pulsa radialmente, variando tanto en temperatura como diámetro para producir cambios de brillo con un periodo y amplitud estables muy regulares

como ASAS o CSS, se ha logrado generar bases de datos del orden de las decenas de terabytes, esperando incluso llegar a los petabytes en algunos años más[7], por lo que es esperable que dentro de esta gran cantidad de información sea posible encontrar una buena cantidad de estrellas que no figuran dentro de ningún catalogo de estrellas variables y que al ser analizadas en el largo periodo puedan presentar el comportamiento descrito anteriormente. Es por esto que el uso de métodos provistos por las ciencias de la computación, como por ejemplo técnicas basadas en aprendizaje automático y minería de datos son indispensable en el análisis de datos astronómicos. Esto permite realizar clasificaciones de grandes volúmenes de datos de forma ágil y eficiente, facilitando enormemente los posteriores estudios referentes al tema que los expertos puedan realizar.

Finalmente podemos destacar que ya existen un sin numero de trabajos relacionados con la clasificación de estrellas variables ”tradicionalesütilizando técnicas de machine learning. Algunos ejemplos de lo mencionado anteriormente lo podemos encontrar en el trabajo titulado ”Meta-classification for variable stars” [33], donde se propone un nuevo meta-modelo que integra automáticamente los modelos de clasificación existentes de estrellas variables. Otro ejemplo interesante puede verse reflejado los resultados obtenidos del proyecto BIGMACC [37], el cual haciendo uso de algoritmos supervisados logro construir un catálogo de clasificación probabilísticos preciso y bien calibrado basado en el catalogo de estrellas variables de ASAS. Si bien trabajos como los mencionados anteriormente tratan la problemática de las estrellas variables desde las ciencias de la computación, estos solo se centran en estrellas variables de corto periodo que ya cuentan con una gran cantidad de estudios, lo que hace posible el uso de estas técnicas supervisadas. Por otro lado a la fecha no existen trabajos que traten el problema de las estrellas variables de largo periodo desde el punto de vista de las ciencias de la computación mediante el uso de algoritmos no supervisados, por lo que este trabajo ayudara a resolver un problema que actualmente no cuenta con una solución que automatice el proceso de detección y clasificación de este tipo de estrellas.

1.2. Objetivos de la memoria

1.2.1. Objetivo general

El objetivo general de esta memoria es abarcar el problema de la clasificación de estrellas variables de largo período desde el punto de vista de las ciencias de la computación, con el fin de proponer un proceso automatizado de análisis que integre de manera apropiada, elementos astronómicos y estadísticos junto con técnicas de aprendizaje automático no supervisadas.

1.2.2. Objetivos específicos

- Estudiar la literatura existente referente a la clasificación computacional de estrellas variables con el fin de conocer el abanico de técnicas disponibles que pueden ser utilizadas.
- Identificar grupos de estrellas y determinar patrones en común que ayuden a realizar un análisis exploratorio de datos por cada grupo.
- Agilizar el análisis de grandes volúmenes de datos.

CAPÍTULO 2
MARCO CONCEPTUAL ASTRONÓMICO

En esta sección se presentarán conceptos astronómicos que son fundamentales para comprender el desarrollo de esta memoria, donde se dará especial énfasis en la definición y taxonomía de las estrellas variables, junto con algunas de las herramientas más útiles usadas para el análisis y obtención de datos.

2.1. Series de tiempo

Una serie temporal se define como una secuencia de n observaciones x_t ordenadas cronológicamente, sobre una característica (serie univariante o escalar) o varias características (serie multivariante o vectorial) de una unidad observable en intervalos regulares de tiempo.

La principal utilidad de estudiar las series de tiempo es para encontrar la relación causal entre diversas variables que cambian con el tiempo e influyen entre sí. Dentro de una serie de tiempo existen cuatro tipos básicos de variabilidad, los cuales sobrepuestos o actuando en forma conjunta, contribuyen a los cambios observados en un periodo de tiempo y otorgan a la curva su comportamiento errático. Estas cuatro componentes son las siguientes:

1. Tendencia secular, la cual indica la marcha general y persistente del fenómeno observado, es una componente de la serie que refleja la evolución a largo plazo.
2. La variación estacional la cual corresponde al movimiento periódico de corto plazo, es decir se trata de una componente causal debido a la influencia de ciertos fenómenos que se repiten de manera periódica en una estación definida (día, fin de semana, mes, año etc).
3. La variación cíclica la cual es el componente de la serie que recoge las oscilaciones periódicas de amplitud superior a un año. movimientos normalmente irregulares alrededor de la tendencia, en las que, a diferencia de las variaciones estacionales, tiene un período y amplitud variables, pudiendo clasificarse como cíclicos, cuasi-cíclicos o recurrentes.
4. La variación irregular la cual se debe a factores a corto plazo, imprevisibles y no recurrentes que afectan a la serie de tiempo.

En astronomía las series de tiempo más estudiadas son las “Curvas de luz”, las cuales contienen información acerca del brillo estelar versus el tiempo y se caracterizan por estar muestreadas de forma irregular, poseer espacios sin datos (gaps) y ser ruidosas. El análisis de este tipo de curvas es importante pues permite estudiar el comportamiento del objeto en cuestión, detectar eventos de interés, ayudar con la clasificación estelar y obtener parámetros del objeto que de otra forma serían difíciles de obtener.

2.2. Estrellas variables

Las estrellas son consideradas variables cuando estas presentan un cambio observable en la intensidad de su brillo a través del tiempo, pudiendo ser de forma periódica o irregular. Este cambio puede ser medido y representado mediante una curva de luz el cual puede variar desde una milésima de magnitud[2] hasta unas veinte magnitudes. Actualmente se estima que existen más de 150.000 estrellas catalogadas como variables y se sospecha que muchas más pueden serlo [25].

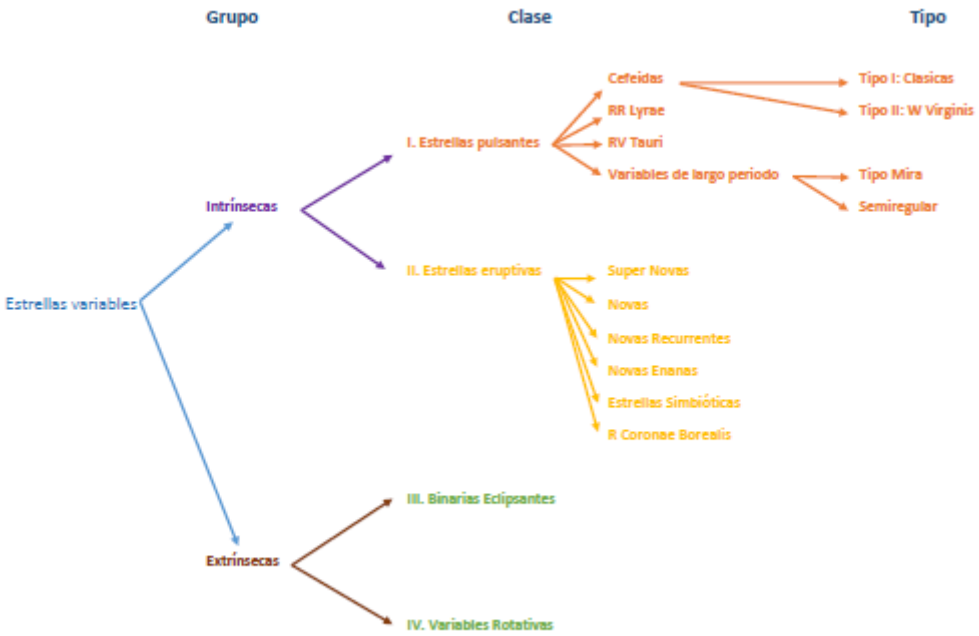

Figura 1: Taxonomía de las estrellas variables.
Fuente: Australia Telescope National Facility [13]

Podemos encontrar dos grupos principales de estrellas variables: Intrínsecas y extrínsecas (en la figura 1 se puede ver la taxonomía completa). Las estrellas intrínsecas se caracterizan debido a que los cambios en su brillo son producidos por cambios físicos o del sistema estelar donde se encuentra presente. Un ejemplo para este tipo de estrella puede ser observado en la figura 2.

[2]En astronomía magnitud es una medida del brillo de una estrella.

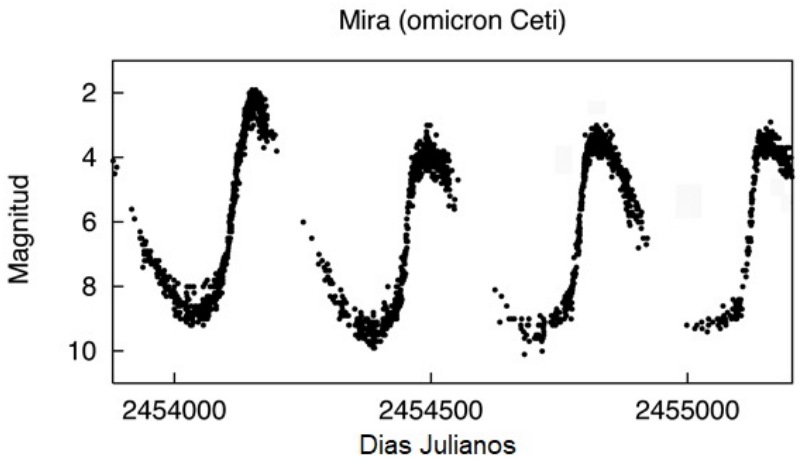

Figura 2: Curva de de luz de una estrella púlsate tipo Mira.
Fuente: American Asociation of Variable Stars Observers[8]

Esta tipo de estrellas se caracterizan por ser súper gigantes rojas que varían con periodos en un rango de 80 a 1000 días y la luz visible llega a variaciones de al menos 2.5 magnitudes [8].

Por otro lado, las extrínsecas presentan cambios en su brillo debido a factores externos que afectan la observación de esta, el ejemplo más claro de este tipo de estrella son las binarias eclipsantes las cuales son estrellas que se eclipsan mutuamente cada cierto periodo de tiempo. En la figura 3 podemos observar la curva de luz de este tipo de estrella.

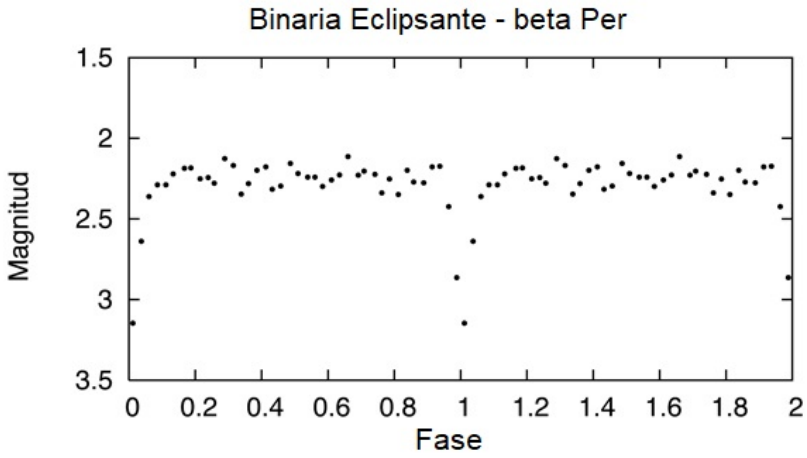

Figura 3: Curva de de luz de una estrella binaria eclipsante.
Fuente: American Asociation of Variable Stars Observers[8]

Este tipo de estrella se caracteriza por formar un sistema binario junto con otra estrella, donde las órbitas de estas se superponen eclipsándose la una a la otra de forma periódica,

provocando una disminución en el brillo observable. Esto puede variar en un periodo de minutos a años.[8]

Pero ¿por qué es importante estudiar este tipo de estrellas? Debido a que con la información obtenida de estas estrellas es posible determinar propiedades físicas importantes de estos cuerpos celestes y del universo en general, como por ejemplo, la composición química, radios, evolución y etapas de las estrellas, distancias a galaxias lejanas y la evolución misma del universo, así como su edad. Dentro de los dos grandes grupos que se definieron anteriormente existen varios tipos, dentro de las cuales centraremos nuestro interés en las de largo periodo.

2.3. Diagrama Hertzsprung- Russell

El diagrama de Hertzsprung- Russell, conocido de forma abreviada como diagrama H-R constituye una pieza central de la astrofísica y es una herramienta fundamental para estudiar las estrellas. Debe su nombre a los trabajos de los astrónomos Ejnar Hertzsprung y Henry Norris Russell [30]. Este diagrama básicamente es un gráfico de dispersión que puede ser representado de variadas formas, siendo en su forma original descrito por la magnitud absoluta de una estrella (Luminosidad en rango visible) frente a su tipo espectral (Temperatura). Por convención, el eje horizontal del diagrama recorre las temperaturas de mayor a menor, mientras que el eje vertical recorre los brillos de menos brillante a más brillante.

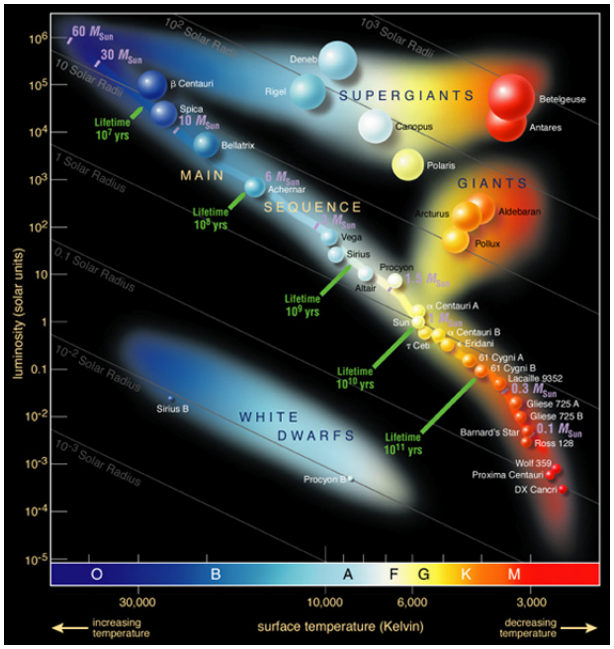

Figura 4: Diagrama H-R mostrando la diversidad de estrellas de la Vía Láctea.
Fuente: European Southern Observatory [11].

La mayoría de las estrellas presentes en este diagrama se agrupan en la curva que denominamos **secuencia principal**, la cual corresponde a la etapa más larga en la vida de una estrella. Las características de las estrellas que habitan este segmento del diagrama varía dependiendo de la posición en la que se encuentren, siendo las denominadas enanas rojas las que se encuentran en la parte baja, las cuales son poco masivas y frías, mientras que en la parte alta podemos encontrar las gigantes azules las cuales son estrellas bastante calientes, luminosas y supermasivas. Otro tipo de estrella que describe este diagrama son las llamadas **gigantes rojas** las cuales son estrellas sustancialmente más grandes y luminosas que las encontradas en la secuencia principal, teniendo un radio entre 10 y 100 veces mayor a estas estrellas. Estás estrella se sitúan por encima de la secuencia principal y su formación se debe a que la estrella agotó todo el hidrógeno disponible para la fusión teniendo como resultado un aumento progresivo de tamaño. Finalmente tenemos el grupo de las **enanas blancas** las cuales son un remanente estelar que se genera cuando una estrella de masa menor a las 10 masas solares se queda sin combustible nuclear.

Como podemos apreciar el diagrama H-R permite visualizar cómo se agrupan los diferentes tipos de estrellas según sus características físicas (Luminosidad, temperatura efectiva, radio, masa, color, etc) y permite entender cómo es la evolución de las mismas a lo largo del tiempo, por lo que será una herramienta a considerar en el desarrollo de esta memoria.

2.4. All Sky Automated Survey

All Sky Automated Survey (ASAS) es un proyecto continuo a largo plazo dedicado a la detección y monitoreo de la variabilidad fotométrica en el brillo de las estrellas en todo el cielo disponible al sur de $\delta < +28°$ del Observatorio Las Campanas, Chile [34]. A la fecha ASAS posee un catálogo cercano a las 17 millones de estrellas en la banda V entre las 8 y 14 magnitudes, con datos recolectados en un periodo de nueve años. Dentro de la base de datos disponible en ASAS podemos encontrar estrellas catalogadas como variables, donde se han identificado un total de 50124 y han sido publicadas en el catálogo de estrellas variables (ACVS) [35], así como también podemos encontrar estrellas que han sido catalogadas como constantes.

2.5. Cross-Matching

En la actualidad se encuentran en funcionamiento una gran cantidad de surveys [28], proyectos astronómicos que mapean el cielo generando grandes catálogos con diversos tipos de información. Gracias a estos ha sido posible generara intersecciones sobre distintos tipos de estudio referentes a objetos que se encuentran en una misma región del cielo. El realizar este cruce de datos se conoce cross-matching [26]. Cuando consideramos la posición del objeto en el cielo (LAT,LONG, RA-DEC, etc.) hablamos de cross-match espacial. Este procedimiento no es sencillo debido a las múltiples fuentes de error, como por ejemplo problemas derivados del proceso de calibración y adquisición de datos o las propiedades físicas y sensibilidad del instrumento las cuales dependen de la longitud de onda del espectro electromagnéti-

co que se esté observando, por lo que es común que dependiendo del catálogo se tengan diferentes coordenadas para un mismo objeto generando imprecisiones que complican el procedimiento.

Para facilitar el cruce de de información es necesario contar con estructuras de datos que permitan hacer una indexación de los objetos en el cielo usando muchas veces basándose en la idea de la esfera celeste[3] con el fin de facilitar su localización y la de sus vecinos. Algunas de las estructuras de indexación espacial más populares se detallan en la tabla 5

Estructura	Descripción
Hierarchical Triangular Mesh (HTM)	En la estructura HTM, la esfera celeste es representada inicialmente por un octaedro, donde cada triángulo se divide en cuatro y se va numerando para conservar la proximidad espacial. Este procedimiento es repetido de forma recursiva hasta alcanzar el nivel de recursión deseado [22].
Hierarchical Equal Area iso-Latitude Pixelisation (HEALPix)	En HEALPix la superficie esférica se subdivide en cuadriláteros curvilíneos, de modo que cada píxel posea la misma área. La partición de resolución más baja está formada por 12 píxeles. En cada subdivisión, todos los píxeles se subdividen en cuatro nuevos píxeles de áreas iguales [15].
Quad Tree Cube	La estructura Q3C proyecta la esfera celeste en un cubo. Cada posición de un astro se asigna a un valor binario de 64 bits a través de la asignación de curva de orden-Z , añadiendo en los tres primeros bits el número de la cara que varía de 0 a 5 [20].
Zones	La estructura ZONES divide la esfera celeste en zonas (anillos de DEC constantes). En general son miles de zonas y cada una de ellas tiene un número. Los objetos se agrupan entonces en sus respectivas zonas, de acuerdo con su posición [16].

Tabla 1: Estructuras de datos usadas en para hacer cross-matching
Fuente: Elaboración Propia.

Finalmente podemos mencionar que la la mayoría de los algoritmos usados para realizar el cruce de información, estos utilizan la posición de los objetos usando sus coordenadas RA (right ascension) y DEC (declination) , en conjunto con alguna de las estructuras de indexación mencionadas anteriormente.

[3]Es una esfera ideal, sin radio definido, concéntrica con el globo terrestre, en la cual aparentemente se mueven los astros, permitiendo representar la posición en que se hallan los objetos celestes

2.6. Observatorios Virtuales

El origen del concepto de Observatorio Virtual surge de una propuesta científica-tecnológica que tiene la intención de dar respuesta al enorme problema de operar adecuadamente la gran cantidad de información producida por diferentes medios que se utilizan para estudiar el Universo (telescopios, modelos numéricos, etc). En términos generales un observatorio virtual puede ser definido como una herramienta computacional que permite almacenar, acceder y procesar información de manera eficiente en grandes bases de datos. El éxito de los Observatorios Virtuales radica en su capacidad de poder realizar interacciones entre las distintas bases de datos de manera sencilla gracias a la existencia de la Alianza Internacional de Observatorios Virtuales, (IVOA por sus siglas en inglés), la cual es la entidad responsable de dictar los estándares sobre la forma en la que deben operar estos. Los estándares van desde la manera en la que se generan los metadatos hasta la implementación de herramientas computacionales que se aplicarán sobre ellos

CAPÍTULO 3
MARCO CONCEPTUAL: CIENCIAS DE LA COMPUTACIÓN

El aprendizaje automático o Machine Learning puede ser entendido como una subrama de la inteligencia artificial que apunta a desarrollar algoritmos capaces de aprender a través de los datos, identificando patrones complejos, es decir, estos algoritmos usan modelos que son capaces de adaptarse a un conjunto de datos de entrenamiento para poder realizar predicciones sobre datos futuros. En la literatura podemos encontrar varias definiciones de Machine Learning, una de las primeras fue dada por Arthur Samuel la cual dice "Es un campo de estudio que da la capacidad de aprender a las computadoras sin ser explícitamente programadas" otra definición interesante fue entregada por Tom Mitchell[24] "Un sistema aprende de la experiencia E con respecto a un conjunto de tareas T y una medida de rendimiento R, si su rendimiento en T, medido según R, mejora con la experiencia E". El uso de este tipo de técnicas, actualmente es de gran ayuda para procesar y analizar de manera más eficiente la gran cantidad de datos entregados actualmente por los Sky Surveys [7]. También en el futuro con la construcción del The Large Synoptic Survey Telescope se tendrá el desafío de analizar la detección de cerca de un millón de eventos astronómicos por noche que este telescopio será capaz de realizar [43, 7], siendo parte de su proyecto científico, el contar con algoritmos que usen técnicas de aprendizaje automático. Dentro de este tipo de algoritmos podemos encontrar, dos enfoques principales: Aprendizaje supervisado y no supervisado. Durante el desarrollo de este capitulo se presentan las características de ambos enfoques en conjunto con algunos de los algoritmos más utilizados para cada uno de estos. También se abarcan otras técnicas y metodologías inherentes al proceso de aprendizaje automático y data mining tales como métricas de desempeño, similaridad o distancia y reducción de dimensionalidad.

3.1. Aprendizaje supervisado

El aprendizaje supervisado consiste en intentar deducir una función que sea capaz de etiquetar nuevos datos sin que se conozca previamente la etiqueta de estos. Esto se logra mediante el uso de un conjunto de entrenamiento que contenga los ejemplos clasificados previamente. Dentro de la estructura del conjunto de entrenamiento encontramos que estos poseen por un lado sus atributos, ya sean numéricos o categóricos y por otro lado tenemos sus etiquetas que indican a que grupo pertenece cada uno de estos. Entonces la idea principal en la fase de entrenamiento es determinar una relación entre los diferentes atributos y el valor de la etiqueta, para luego poder predecir datos no categorizados.

Uno de los algoritmos mas populares y sencillos de clasificación no supervisada es **K vecinos más cercanos**, el cual se presenta en la sección 3.1.1.

3.1.1. k vecinos más cercanos (KNN)

Este algoritmo consiste en clasificar un nuevo objeto en base a la clase más frecuente de sus K vecinos más cercanos. Su funcionamiento consiste en almacenar en un vector el conjunto de entrenamiento, junto a la clase asociada para cada muestra de este conjunto. Como primer paso se calculara la distancia euclidiana entre el nuevo objeto sin clasificar y cada elemento del conjunto de entrenamiento, quedándonos con las k muestras más cercanas y clasificando la nueva muestra de entrenamiento en la clase más frecuente a la que pertenecen los k vecinos obtenidos anteriormente.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{k,i} - x_{k,j})^2} \tag{1}$$

Una vez hecho esto, es necesario realizar el mismo proceso pero con un conjunto de datos de validación, donde se calcularan porcentajes de clasificación sobre los ejemplos de este conjunto para poder estimar el poder de generalización del clasificador.

Las principales ventajas de este algoritmo son que su coste de aprendizaje es nulo, es un clasificador robusto y versátil tolerante al ruido de los datos y además puede ser usado para hacer regresiones. Sus principales desventajas radican en que el costo de encontrar los k mejores vecinos es alto, no hay un método para decidir el valor óptimo para k y si el número de descriptores crece su rendimiento disminuye.

3.1.2. Arboles de decisión

Esta técnica sirve para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema. La estructura del árbol conociste principalmente en nodos interiores y nodos terminales (hojas) unidos mediante ramas. Cada nodo interior tiene características, mientras que cada hoja un valor distinto del atributo. Si seguimos el árbol desde el nodo raíz hacia abajo, cada ruta posible terminara en una hoja creando así una segmentación de los datos.

Dentro de las ventajas que podemos encontrar para usar este algoritmo es que es un modelo simple e intuitivo, siendo capaces de manejar datos continuos y discretos. Así también sólo se utilizan las características que son necesarias para separar el conjunto de entrenamiento en grupos homogéneos o casi homogéneos en términos de la variable de clase. Como principales desventajas que podemos encontrar en los arboles de decisión es que las pruebas que se realizan dentro estos solo se basan en atributos únicos, también son incapaces de aprender combinaciones aritméticas de características que son útiles en cualquier clasificador.

Un trabajo interesante donde se usa este algoritmo es en [33], en donde se construyo un meta-modelo basado en un árbol de decisión para la reutilización de otros modelo que clasificaban estrellas variables de forma automática bajo diversos contextos y diferentes representaciones de datos.

El detalle de este algoritmo puede ser revisado en la sección 7.1.1 de los anexos.

3.1.3. Support vector machines (SVM)

La técnica SVM está basada en la minimización del riesgo estructural (SRM), la cual consiste en elegir un conjunto de funciones de clasificación para un con conjunto de datos de tal forma que se minimice el error de entrenamiento. [41].

Las ventajas de este método radican en que resuelve un problema de optimización que asegura entregar una solución óptima para el conjunto de entrenamiento, a pesar de que las SVM pueden ser no lineales en el espacio original. Además esta técnica puede ser usada con una gran cantidad de características de entrada y trabajar sobre varios dominios a la vez. Como principal desventaja es que este método posee un gran costo computacional, y como puede trabajar con varias clases a la vez, las cuales las divide en $|C|$ SVMs que trabajan con dos clases, y si dos clases son mutuamente excluyentes y si además cierto dato de la clase de entrenamiento aparece clasificado dentro de dos clases distintas, es imposible saber a cual corresponde realmente.

El detalle de este algoritmo puede ser revisado en la sección 7.1.2 de los anexos.

3.2. Aprendizaje semi supervisado

Como se pudo ver en la sección anterior, para poder utilizar técnicas de aprendizaje supervisado, es necesario contar previamente con un conjunto que posea una buena cantidad de datos etiquetados, para ser utilizado como conjunto de entrenamiento y obtener un buen desempeño en los algoritmos. Lo anterior muchas veces no es posible, debido a que la clasificación y etiquetado depende muchas veces del tiempo y esfuerzo de especialistas.

Para solucionar esta problemática se ha planteado el enfoque de aprendizaje semi-supervisado. En este enfoque los datos se dividen en dos conjuntos: $L = \{x_1, \cdots, x_l\}$ para los cuales se conocen sus etiquetas $Y = \{y_1, \cdots, y_l\}$, y el conjunto $U = \{x_{l+1}, \cdots, x_{l+u}\}$, para los cuales no se conocen sus etiquetas. Debido a esto, tenemos que este se encuentra entre los enfoques supervisado y no supervisado.

La idea general de este enfoque, es aprender tanto de los datos etiquetados como de los no etiquetados, combinando ambas técnicas de aprendizaje (supervisado y no supervisado). En esencia, los métodos semi-supervisados utilizan los datos no etiquetados para producir modificaciones o reformular las hipótesis de los datos ya clasificados [40].

El aprendizaje semi-supervisado se puede dividir en dos clases principales, aprendizaje transductivo y aprendizaje inductivo.

- **Aprendizaje transductivo**

El objetivo en esta clase de aprendizaje es predecir la etiqueta correcta en la fracción de datos no etiquetada que es utilizada para el aprendizaje semi-supervisado.

- **Aprendizaje inductivo**

  En esta clase se intenta obtener la función de aprendizaje que permita predecir la etiqueta correcta en datos nuevos no etiquetados que no se han utilizado para el aprendizaje semi-supervisado.

3.3. Clustering

Al contrario de los algoritmos supervisados, para este caso no se disponen de datos previamente etiquetados, sino que se buscan agrupaciones (clusters) según similitudes en las características de los datos que están siendo estudiados. Una vez determinados los grupos, para que estos sean de utilidad, pueden ser analizados detenidamente con el objetivo de descubrir las claves o propiedades que han producido tal agrupamiento. La aparición de esta estructura en los datos, puede llevar a aprender aspectos o propiedades de los elementos de la muestra que de otro modo habrían pasado inadvertidos.

Desde el punto de vista de la astronomía este tipo de algoritmos han contribuido a las investigaciones a identificar subgrupos no reconocidos previamente dentro de los conjuntos de datos, detectar observaciones anómalas y permitir el análisis de datos que contienen características faltantes. Estos métodos son una adición útil a la caja de herramientas de cualquier investigador y se espera que ayuden en futuros descubrimientos. [43].

En el desarrollo de esta sección se presentarán algunas medidas de similariadad, junto con los algoritmos K-means, EM, Jerárquico aglomerativo y DBSCAN.

3.3.1. Funciones de proximidad

Como se menciono anteriormente al aplicar algoritmos de clustering, estos no disponen de etiquetas que sirvan para realizar un entrenamiento previo y determinar los patrones subyacentes que ayuden a una posterior clasificación de datos no etiquetados, por lo que es necesario contar con una alternativa que nos entregue una medida de similaridad entre los datos para poder encontrar los distintos grupos.

Dentro de las alternativas disponibles la más común es la distancia entre dos individuos x e y a una medida, indicada por $d(x, y)$, que mide el grado de semejanza, entre ambos, en relación a un cierto número de características cuantitativas y/o cualitativas. El valor de $d(x, y)$ es siempre un valor no negativo, y cuanto mayor sea este, mayor será la diferencia entre x e y.

Toda distancia debe verificar, al menos, las siguientes propiedades:

1. $d(x, y) > 0$ (no negatividad).
2. $d(x, x) = 0$
3. $d(x, y) = d(y, x)$

Existe una gran cantidad de distancias e indicadores de similaridad y no se puede disponer de una regla general que permita definir una métrica que sea conveniente para todo tipo de análisis ya que la elección de una u otra depende del tipo de análisis que se quiera realizar y la naturaleza de las variables que se estén estudiando.

A continuación se presentaran algunas de las métricas de distancia mas conocidas.

Métrica	Función	Descripción
Minkowski	$d(X,Y) = \left(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^d\right)^{\frac{1}{p}}$	Familia de distancias. A medida que varia su valor d, se genera una nueva métrica de distancia.
Manhattan	$d(X,Y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$	Caso especial de Minkowski para p = 1. Tiende a formar clusters hiperrectangulares
Euclidiana	$d(X,Y) = \left(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^2\right)^{\frac{1}{2}}$	Caso especial de Minkowski para p = 2. Tiende a formar clusters hiperesféricos.
Mahalanobis	$d(X,Y) = \sqrt{(X-Y)^T \Sigma (X-Y)}$	Σ representa la matriz de covarianzas entre los datos. Es invariante ante los cambios de escala por lo que no depende de las unidades de medida. Tiende a formar clusters hiperelipsoidales.
Pearson	$\rho(X,Y) = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$	No es una métrica. Se deriva de un coeficiente de correlación. No mide magnitud.
Coseno	$s_{cos}(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$	Es eficiente, especialmente en vectores dispersos (independiente de la cantidad de componentes en el vector).

Tabla 2: Funciones de distancia

3.3.2. K-Means Clustering

K-Means es uno de los métodos más simples para particionar el conjunto de datos en K subconjuntos C_k que contienen N_k puntos. Este algoritmo busca reducir al mínimo la suma de las distancias cuadradas desde la media dentro de cada cluster minimizando la siguiente función objetivo:

$$\sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2 \tag{2}$$

donde $\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i$, es el promedio de los puntos del conjunto C_k y $C(x_i) = C_k$ denota que la clase de x_i es C_k. El algoritmo se presenta a continuación.

Algorithm 1 K-Means [2]

1: Seleccionar K puntos como centroides μ_k iniciales
2: **repeat**
3: Formar K Clusters C_k mediante la asignación de puntos x_i al centroide μ_k más cercano
4: Recalcular cada centroide $\mu_k \in C_k$
5: **until** Se cumpla el criterio de convergencia

Este algoritmo no asegura que se encuentre el óptimo global, sin embargo la suma de las distancias cuadradas siempre se mejorará, además este algoritmo se ejecuta varias veces para distintos valores iniciales de μ_k quedándose con el valor que entregue el menor error en la suma de distancias cuadradas. Su principal ventaja radica en su simplicidad y eficiencia, mientras que sus desventajas son principalmente, la sensibilidad al ruido, outliers, la elección de la cantidad de clusters y el centroide inicial.

3.3.3. Expectation-Maximization clustering

Este algoritmo proporciona una estrategia para llevar el método de optimización de k-means a un entorno probabilístico. A diferencia de K-means donde cada cluster C_k es una ubicación en el espacio de características, EM usa distribuciones probabilísticas como por ejemplo la normal(Gaussiana) para construir los clusters C_k. Cuando se usan distribuciones Gaussianas este método es conocido como **Gaussian Mixture (GMM)**, método que según [43] es uno de los más famosos en el análisis de datos astronómicos. Este algoritmo asume que todos los puntos de la data son generados por una mezcla de distribuciones Gaussianas (conocida también como distribución normal) con parámetros desconocidos. En el caso de una variable única x la distribución Gaussiana puede ser escrita de la siguiente forma.

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3}$$

Donde μ representa la media aritmética y σ^2 representa la varianza. Para el caso donde tenemos vectores x de datos de dimensión D, la distribución Gaussiana multivariada se puede expresar como:

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \cdot e^{-\frac{1}{2}\left((x-\mu_i)^T\Sigma^{-1}(x-\mu)\right)} \tag{4}$$

Donde μ es un vector de medias, Σ es una matriz cuadrada de covarianzas de dimensiones DxD y $|\Sigma|$ es el determinante de dicha matriz.

La función de log-verosimilitud para el caso multivariado puede ser escrita de la siguiente forma

$$l(x_i,\mu_k,\Sigma_k) = \sum_{i=1}^{N} log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k) \tag{5}$$

Para encontrar soluciones de máxima verosimilitud es necesario derivar la función log-verosimilitud respecto de de los parámetros μ_k, Σ_k, π_k. Aquí solo serán presentados los resultados finales omitiendo todo el trabajo algebraico necesario para despejar los términos, si el lector considera pertinente la revisión de estos puede consultar [2] donde se encuentra un desarrollo más completo.

Derivando respecto a μ_k encontramos

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_{i=1}^{K} \gamma_{ik}} \tag{6}$$

donde tenemos que la media μ_k esta explicada como una medida ponderada de los puntos pertenecientes al conjunto de datos, donde el factor de ponderación esta dado por la probabilidad posterior γ_{ik} de que el componente K fuera el responsable de generar x.

Con la derivada anterior también es posible encontrar una expresión para la probabilidad posterior γ_{ik}.

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k,\sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j,\sigma_j)} \tag{7}$$

De manera análoga usando la derivada de la función log-verosimilitud con respecto a Σ_k obtenemos

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_{i=1}^{N} \gamma_{ik}} \tag{8}$$

Para este caso también podemos notar que cada punto es ponderado por la probabilidad posterior γ_{ik} generada por el componente correspondiente y con el denominador dado por el número efectivo de puntos asociados con el componente correspondiente. Finalmente encontramos la probabilidad de mezcla derivando respecto a π_k.

$$\pi_k = \sum_{i=1}^{N} \frac{\gamma_{ik}}{N} \tag{9}$$

A continuación se presenta un algoritmo para encontrar soluciones de máxima verosimilitud para modelos con variables latentes.

Algorithm 2 EM para mezclas Gaussianas [2]

1: Dado un conjunto de datos y un modelo de mezclas Gaussianas, se busca maximizar la log-verosimilitud en base a los parámetros μ_k, Σ_k, π_k
2: Inicializar las medias $\mu_k^0, \Sigma_k^0, \pi_0^k$
3: **Expectación (E)**: Se calcula la probabilidad posterior de que el componente K fuera responsable de generar x_n según (7)
4: **Maximización (M)**: Se actualizan los parámetros usando el valor actual de γ_{ik} siguiendo el siguiente orden

- Se actualizan las medias μ_k según (6)
- Se actualizan las matrices de covarianzas Σ_k según (8)
- se actualizan las probabilidades de mezcla π_k según (9)

5: Calcular la log-verosimilitud según (5)
6: **if** Se cumple el criterio de convergencia establecido **then**
7: **return** Los parámetros finales
8: **else**
9: Se repiten los pasos entre las lineas 3-5
10: **end if**

Comparado este algoritmo con K-means, tenemos que le toma bastantes más iteraciones para converger [6, 31]. También existirán muchos máximos locales para la log-verosimilitud, los cuales dependerán de las distintas inicializaciones para los parametros μ, Σ, π, y el algoritmo EM no garantiza encontrar el óptimo global. Una solucion para encontrar una inicialización apropiada y agilizar el proceso de convergencia para un modelo de mezclas de Gaussianas, consiste en ejecutar el algoritmo K-means [6] y elegir las medias y covarianzas de los clusters, y las fracciones de los puntos asignados a los clusters respectivos, para inicializar a los parámetros $\mu_k^0, \Sigma_k^0, \pi_0^k$ respectivamente.

3.3.4. Clustering jerárquico aglomerativo

En un algoritmo de clustering jerárquico, se usara una estructura tipo árbol binario, donde la raíz representa el nodo donde se encuentran presentes todo el conjunto que contiene a todos los datos que posteriormente serán agrupados, formando así la cúspide de la jerarquía. En cada nivel, los nodos serán hijos del nivel anterior representando un subconjuntos de datos del nodo padre, los cuales corresponden a clusters. Las entradas asociadas a cada uno de estos pueden ser determinadas bajando por el árbol, desde el nodo actual hasta las hojas que contiene los puntos de datos singulares. A esta tipo de estructura se le denomina dendograma. Por otro lado los métodos aglomerativos comienzan determinando clusters singulares (que contienen sólo un objeto de datos por clusters) en el nivel más bajo, y continúan uniendo dos clusters cada vez para construir una jerarquía bottom-up (de abajo hacia arriba) de los clusters. Para poder realizar el agrupamiento es necesario saber las distancias entre clusters. Para esto existen diferentes tecnicas de vinculación las que pueden generar diferentes dendogramas.

Por ejemplo el método de enlace simple define la distancia entre dos clusters C_i, C, j como:

$$d(C_i, C_j) = \underset{x_a \in C_i, x_b \in C_j}{min} d(x_a, x_b) \tag{10}$$

Por otro lado el método del enlace completo (Complete linkage) la define como

$$d(C_i, C_j) = \underset{x_a \in C_i, x_b \in C_j}{max} d(x_a, x_b) \tag{11}$$

Finalmente el método del enlace promedio (Averange linkage) define la distancia como

$$d(C_i, C_j) = \frac{1}{(n_i + n_j)} \frac{1}{(n_i + n_j - 1)} \sum_{x_a \in C_i} \sum_{x_b \in C_j} d(x_a, x_b). \tag{12}$$

Finalmente tenemos el método ward, el cual Minimiza la suma de las diferencias cuadradas dentro de todos los grupos. Es un enfoque que minimiza la variación y en este sentido, es similar a la función objetivo de k-means, pero se aborda con un enfoque jerárquico.

Algorithm 3 Clustering Jerárquico Aglomerativo [43]

1: **INPUT:** Data set $\mathcal{X}$ con n datos
2: **OUTPUT:** Particiones jerárquicas $\mathcal{H} = \{p_1, ..., p_n\}, p_i = \{c_1, ..., c_i\}$.
3: Sea p_n una partición para un cluster formado por un ítem. Se forman n clusters
4: **for** $i = n - 1$ **to** 1 **do**
5: Se crea p_i mezclando los dos clusters más cercanos
6: **end for**

Este algoritmo al contrario del los dos presentados anteriormente es determinista, lo que quiere decir que siempre se obtendrá el mismo resultado para un conjunto de datos dado. Por otro lado la principal ventaja de tener un método de agrupamiento jerárquico es que éste

permite cortar la jerarquía en cualquier nivel determinado y obtener los clusters correspondientes. Esta característica lo diferencia significativamente de los métodos de agrupamiento divisivos como K-Means, ya que no requiere de un parámetro predefinido K (número de clusters) especificado por el usuario. Como desventaja este algoritmo es algo costoso computacionalmente hablando, ya que obtener todas las distancias se tiene una complejidad de $\mathcal{O}(n^2)$ y se requieren hacer operaciones adicionales para construir la jerarquía y determinar las distancias mínimas entre los n niveles lo que tiene una complejidad de $\mathcal{O}(n^3)$.

3.3.5. DBSCAN

DBSCAN es un algoritmo de clustering que tiene un enfoque basado en la densidad, modelando cada cluster como cúmulos de observaciones, por lo que si un punto pertenece o no a un cluster en particular, debe encontrarse cercano a varios puntos de este [12]. Este algoritmo define dos parámetros importantes, un número épsilon positivo y un número natural que será la cantidad mínima de puntos para formar un grupo. El algoritmo se presentara a continuación

Algorithm 4 Algoritmo DBSCAN [18]

Require: Data set $\mathcal{X}$ con n datos, ϵ: radio de la vecindad, MinPts: cantidad mínima de puntos.
Ensure: Conjunto de clusters basados en densidad

```
Marcar todos los datos como NO visitados
while Existan datos sin visitar do
    Seleccionar un elemento p que no este visitado de forma aleatoria;
    Marcar p como visitado;
    if dentro del radio ϵ de p existen al menos MinPts puntos then
        Crear un nuevo cluster C y agregar p a C;
        Considerar N como la cantidad de puntos dentro del radio de p;
        for cada punto p' en N do
            if p' no está visitado then
                Marcar p' como visitado;
                if dentro del radio de p' existen al menos MinPts puntos then
                    Agregar estos puntos a N;
                end if
            end if
            if p' aún no es miembro de algún cluster then
                Agregar p' a C;
            end if
        end for
    else
        Marcar p como ruido
    end if
end while
```

En síntesis lo que este algoritmo hace como se pudo ver, se elige un punto arbitrario en

el conjunto de datos. Si hay una cantidad de puntos mayor o igual a n dentro del radio ϵ se consideran todos los puntos como parte del mismo cluster. A continuación, se realiza una comprobación de todos los nuevos puntos, revisando si ellos también tienen más puntos que cumplan la condición anterior, haciendo que el cluster crezca de forma recursiva. A medida que se generan los distintos clusters, puede darse el caso que el punto arbitrario escogido tenga menos de n puntos en su círculo de radio ϵ , y tampoco forme parte de otra agrupación. Si esto es así se considera un ”punto de ruido”que no pertenecen a ningún grupo.

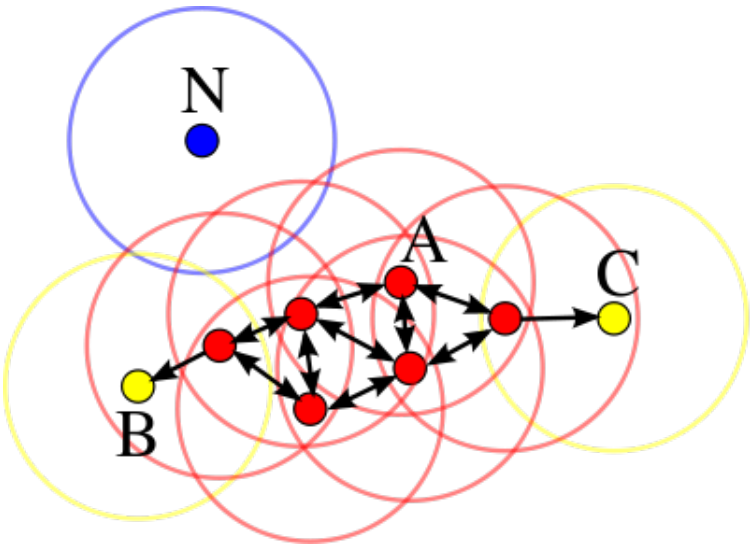

Figura 5: Ejemplo de DBSCAN.

Entonces los distintos tipos de puntos que podemos observar en la construcción de un cluster como podemos apreciar en la imagen son los siguientes

1. Puntos pertenecientes al núcleo: Estos son los puntos centrales del cluster que cumplen con la cantidad mínima de n puntos en su vecindario de radio ϵ .
2. Puntos pertenecientes a la frontera: Estos puntos pertenecen al cluster sin embargo en su vecindario de radio ϵ poseen menos de n puntos. Estos puntos pertenecen a la vecindad de algún punto del núcleo.
3. Puntos de ruido: Cualquier punto que no forma parte de un cluster, es decir que no se encuentra ni en su núcleo ni en sus frontera .

Ventajas de usar este algoritmo

- Encuentra clusters no separables linealmente.
- No necesita asumir un número fijo de clusters.

- No depende de las condiciones de inicio.

Desventajas de usar este algoritmo

- Asume densidades similares en todos los clusters.
- Puede tener problemas al separar clusters.
- Encontrar el parámetro ϵ óptimo puede resultar complicado.

3.4. Medidas de desempeño

En general cuando trabajamos con algoritmos de clustering, estos buscan encontrar grupos donde sus elementos internos estén bien condensados y la separación entre los diversos grupos esté bien definida, es decir que la distancia intra-clusters sea mínima y que la distancia inter-cluster sea la máxima posible. A continuación presentaremos algunos de los métodos que podemos encontrar para medir el desempeño de estos algoritmos sin la necesidad de contar con un conjunto de entrenamiento previamente etiquetado.

3.4.1. Índice de Calinski-Harabasz

El método de Calinski-Harabasz [21] también conocido como criterio de relación de varianza, puede ser usado para evaluar el modelo, donde mientras más alto sea el puntaje obtenido, este indicará que los grupos encontrados estarán mejor definidos. Este índice está construido como la razón entre la dispersión interior de los clusters y la dispersión entre clusters. Formalmente para clusters este índice se obtiene de la siguiente forma:

$$CH(k) = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{N-1}{k-1}$$

B_k es la matriz de dispersión inter-clusters y W_k la matriz de dispersión intra-cluster, las cuales están definidas como sigue:

$$B_k = \sum_q n_q (c_q - c)(c_q - c)^T$$
$$W_k = \sum_{q=1} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

Con N como el número total de puntos de data, C_q como el conjunto de puntos perteneciente al cluster q y c_q como su centro, c el centro de E y n_q la cantidad de puntos del cluster.

Ventajas de usar este método

- El puntaje es rápido de calcular.
- El puntaje será alto cuando se tengan clusters densos y bien separados.

Desventajas de usar este método

- En general los puntajes son altos para clusters del tipo convexos por sobre otro tipo de como por ejemplo los basados en densidad que pueden ser obtenidos mediante un algoritmo como DBSCAN.

3.4.2. Índice de Davies-Bouldin

Otro criterio que podríamos considerar es el índice Davies-Bouldin [9], el cual consiste en minimizar la función de davies-bouldin

$$DB = \sum_{i=1}^{k} \underset{i \neq j}{max} R_{i,j}$$

donde $R_{i,j} = \frac{s_i + s_j}{d_{i,j}}$ con s_i como la media de las distancias de cada elemento de un cluster i a su centroide midiendo así la distancia intra-cluster, mientras que $d_{i,j}$ representa la distancia entre los centroides midiendo así que tan buena es la separación inter-cluster.

- Ventajas de usar este método
  - El cálculo del índice de Davies-Bouldin es simple y rápido.
  - El índice se calcula sólo en cantidades y características inherentes al conjunto de datos.
- Desventajas de usar este método
  - En general los puntajes son altos para clusters del tipo convexos por sobre otro tipo de como por ejemplo los basados en densidad que pueden ser obtenidos mediante un algoritmo como DBSCAN.
  - El uso de la distancia entre centroides limita la métrica de la distancia al espacio euclidiano.
  - Un buen valor reportado por este método no implica que los clusters encontrados están bien definidos.

3.4.3. Índice Silhouette

Dado un conjunto de datos definiremos un conjunto de particiones como $C = \{C_1, C_2, C_3, \ldots, C_k\}$, donde C_k denota el k-ésimo grupo en el conjunto de datos, con $k = 1, 2, 3, \ldots, K$. El índice silhouette [38] es calculado en base a la idea de silhouette width el cual puede ser expresado como:

$$S(x) = \frac{b(x) - a(x)}{max(a(x), b(x))}$$

Donde $a(x)$ es el promedio de la distancia intra-cluster, el cual puede ser expresado como:

$$a(x) = \frac{1}{n_k - 1} \sum_{x_k \in C_k} d(x, x_k)$$

con n_k como la cantidad de elementos del cluster C_k. Por otro lado $b(x)$ es la distancia promedio mínima entre los elementos de C_k con respecto de los elementos de otros clusters C_l, donde $l = 1, 2, 3 \ldots K$ y $l \neq K$ entonces la distancia mínima puede ser definida como:

$$b(x) = min_{l=1}^{K} d(x, x_l)$$

donde n_l es el número de elementos en cada grupo C_l. Usando las definiciones anteriores tenemos que el silhouette width de C_k puede ser obtenido como:

$$S(C_k) = \frac{1}{n_k} \sum_{x \in C_k} S(x)$$

Finalmente el índice silhouette queda expresado como:

$$Sil = \frac{1}{K} \sum_{k=1}^{K} S(C_k)$$

El puntaje entregado por este índice puede tomar valores comprendidos entre $[-1, 1]$ y su interpretación es la siguiente:

Valores	Interpretación
Valores en el rango $[1{,}0, 0{,}7)$	Los grupos encontrados poseen una estructura interna fuerte. Similaridad entre elementos de un mismo grupo es alta.
Valores en el rango $[0{,}7, 0{,}5)$	Los grupos encontrados presentan una estructura razonable. Similaridad entre elementos de un mismo grupo es buena.
Valores en el rango $[0{,}5, 0{,}25)$	Los grupos encontrados presentan una estructura débil.
Valores en el rango $[0{,}25, 0{,}0]$	Los grupos encontrados se encuentran superpuestos entre sí.
Valores entre el rango $(0{,}0, -1{,}0]$	Los datos se encuentran asignados a grupos incorrectos. Tener puntajes negativos implica que elementos de grupos diferentes son más similares que elementos pertenecientes al mismo grupo.

Tabla 3: Valores índice de silhouette
Fuente: Elaboración Propia

- Ventajas
  - Rango de puntajes bien definido y de fácil interpretación.
  - El puntaje será alto cuando se tengan clusters densos y bien separados.
- Desventajas
  - En general los puntajes son altos para clusters del tipo convexos por sobre otro tipo de como por ejemplo los basados en densidad que pueden ser obtenidos mediante un algoritmo como DBSCAN.

3.5. Reducción de dimensionalidad

3.5.1. Maldición de la dimensionalidad

En un inicio el concepto de maldición de la dimensionalidad fue descrito por Richard Bellman en el libro Adaptive control processes[5], donde hacía referencia a la imposibilidad de optimizar una función de varias variables por medio de una búsqueda mediante fuerza bruta en una grilla espacial discreta multidimensional ya que el número de puntos de esta aumentaba exponencialmente a medida que se tenían más dimensiones. Con el paso de los años este concepto ha pasado a usarse para describir cualquier tipo de problema que requiera hacer el análisis de datos en una gran número de atributos. En términos generales, los problemas

derivados de trabajar con datos de alta dimensión es que el número fijo de puntos se vuelven cada vez más dispersos a medida que aumenta la dimensionalidad. Para entender esto usaremos el siguiente ejemplo obtenido de [10]:

Supongamos que arrojamos una moneda en un estadio de fútbol y sólo puede caer en una línea de 100 metros de longitud. Para encontrarla solo bastaría con buscar sobre esa línea, donde podríamos incluso caminar sobre ella y encontrarla relativamente fácil. Ahora vamos a suponer que arrojamos la misma moneda, sólo que esta puede caer en la superficie cuadrada del estadio, por lo que para buscarla tendríamos que recorrer toda la superficie, dificultando esta tarea. Finalmente si suponemos que el área de búsqueda aumenta a tres dimensiones, podría darse el caso que la moneda puede caer en cualquier punto cúbico del estadio, es decir podría estar flotando en algún lugar dentro de ese estadio.

Este mismo comportamiento ocurre cuando tenemos una alta dimensionalidad en nuestros datos, donde estos se "pierden en el vacío del espacio" a medida que aumentamos la dimensionalidad.

Para propósitos de clustering el aspecto que más afecta de la maldición de la dimensionalidad, es que muchos de los algoritmos usan métricas de distancia o similaridad, donde se requieren que los objetos dentro de un mismo cluster están en general más cercanos entre sí y más alejados de los pertenecientes a otros. Si esto no es así los algoritmos encontrarán clusters sin sentidos, por lo que esta distorsión en el espacio puede afectar negativamente el desempeño de algunos algoritmos.

3.5.2. Análisis de componentes principales

El análisis de componentes principales es un método de aprendizaje no supervisado que tiene como objetivo aprender una nueva transformación más compacta de los datos dado un conjunto de n observaciones y p variables manteniendo la mayor cantidad de información relevante. En general las principales virtudes de PCA [32] son :

1. Permite representar en un espacio dimensionalmente más pequeño observaciones de un espacio p-dimensional manteniendo la información relevante de las variables originales.
2. Permite transformar las variables originales, en general correladas, en unas variables incorreladas, facilitando así la interpretación de los datos.

Obtención de las componentes principales

Sea un conjunto de datos representados por una matriz X de orden mxn. PCA encuentra una matriz ortonormal P que proyecta X a una representación Z de máxima varianza, a través de la transformación $Z = PX$, de tal forma que la matriz $C_z = \frac{1}{m}ZZ^T$, es una matriz

diagonal y las filas de la matriz son las componentes principales de X. Aplicando álgebra de matrices podemos reescribir la matriz C_z de la siguiente forma:

$$\begin{aligned} C_z &= \frac{1}{m} Z Z^T \\ &= \frac{1}{m}(PX)(PX)^T \\ &= \frac{1}{m} P X X^T P^T \\ &= P(\frac{1}{n} X X^T) P^T \\ &= P C_x P^T \end{aligned}$$

Donde la matriz C_x es simétrica y representa a la matriz de covarianzas de X. Considerando el siguiente teorema: Sea A una matriz simétrica, podemos reescribir esta como $A = VUV^T$, donde U es una matriz diagonal que contiene los valores propios de A y V es una matriz de vectores propios organizados como columnas. Usando este teorema podemos decir que cada una de las componentes principales de C_x están dadas por los valores propios de esta, mientras que la matriz C_z representa a la varianza de X.

En conclusión si se desea proyectar la matriz de datos X a un subespacio $k < n$ se deben obtener las primeras k vectores propios $p_1, p_2 \ldots p_k$ de C_x. Luego para representar x_i en el nuevo espacio se debe generar el siguiente vector:

$$z_i = \begin{bmatrix} p_1 x_i \\ \vdots \\ p_k x_i \end{bmatrix} \in \mathbb{R}^n$$

Finalmente podemos mencionar algunas consideraciones con respecto a este método

- El proceso de PCA identifica las direcciones en las que la varianza es mayor. La varianza de una variable se mide en su misma escala elevada al cuadrado,esto quiere decir que PCA es altamente sensible a la escala de los datos,por lo que es de suma importancia que las variables se normalicen para que tengan media 0 y desviación estándar 1, en caso contrario las variables que tengan una mayor escala, dominaran por sobre el resto.
- En general para poder determinar con cuantas componentes principales nos quedamos, se puede hacer uso de la varianza explicada por cada componente principal y la varianza explicada acumulada. Esto nos ayuda a ver que porcentaje de la estructura

original de los datos se conserva en determinada cantidad de componentes principales. La sumatoria de la proporción de varianza explicada acumulada de todas las componentes es siempre 1, y en general para determinar la cantidad correcta se tiende a escoger hasta la cantidad de componentes que no presenten un aumento sustancial en la varianza explicada acumulada.

3.5.3. t-distributed stochastic neighbor embedding (t-SNE)

Esta técnica esta enfocada a la visualización de datos en presencia de alta dimensionalidad, es una extensión de el algoritmo stochastic neighbor embedding (SNE) [23]. La idea central del algoritmo SNE es tratar de mapear objetos presentes en una alta dimensionalidad a una dimensionalidad menor preservando la estructura de parentesco entre la vecindad de elementos. Para esto lo que se hace es centrar una distribución Gaussiana para cada valor de entrada a lo largo del espacio de alta dimensionalidad, con el fin de poder usar su densidad para definir una función de probabilidad de todos los vecinos.

Formalmente el algoritmo funciona de la siguiente manera:

Como primer paso se convierten las distancias euclidianas multidimensionales entre pares de observaciones en probabilidades condicionales. Esto quiere decir que en lugar de medir la similitud entre dos observaciones x_i y x_j, se mide la probabilidad condicional de que x_j fuese seleccionada como observación vecina de x_i si las observaciones perteneciesen a una distribución de densidad Gaussiana centrada en x_i. Para observaciones cercanas, $p_{j|i}$ es alta, mientras que para observaciones alejadas la probabilidad es mínima.

$$p_{j|i} = \frac{e^{\frac{-||x_i - x_j||^2}{2\sigma_i^2}}}{\sum_{i \neq k} e^{-||x_i - x_k||^2}}$$

Donde σ_i es la varianza de la distribución Gaussiana en x_i

El segundo paso consiste en crear dos observaciones y_i, y_j homólogas a x_i, x_j pero en una dimensionalidad menor, cuya distancia se define como la probabilidad condicional $q_{j|i}$

$$q_{j|i} = \frac{e^{\frac{-||y_i - y_j||^2}{2\sigma_i^2}}}{\sum_{i \neq k} e^{-||y_i - y_k||^2}}$$

Como resultado de los pasos anteriores obtenemos la probabilidad condicional de la similitud entre pares de observaciones en el espacio multidimensional original y en un espacio de menor dimensión. Para que las observaciones y_i,y_j creadas en el espacio de menor dimensión representen de forma fiel a las observaciones en el espacio multidimensional original,

las probabilidades condicionales $p_{j|i}$ y $q_{j|i}$, deben ser exactamente las mismas, es decir la diferencia entre estas debe ser cero.

El algoritmo t-SNE se diferencia del algoritmo SNE en el modo en que se minimizan las diferencias de las probabilidades condicionales. Como se mencionó anteriormente es necesario fijar un valor para el parámetro σ_i, donde es muy poco probable que este que exista un valor para este que sea óptimo para todas las observaciones del set de datos, ya que las densidades suelen variar entre las diferentes regiones, por ejemplo, para regiones densas tenemos que un valor de σ_i bajo es más apropiado que para regiones más dispersas. Para solucionar este problema t-SNE tiene el parámetro perplexity, el cual es fijado por un usuario y puede entenderse como una medida del número de observaciones vecinas que tienen que emplearse en cada estimación local. Dentro delos valores recomendados para este parámetro tenemos que varían en un rango entre 5 y 50.

El algoritmo t-sne es una herramienta poderosa para la visualización de datos, sin embargo, tiene algunas consideraciones que deben ser tomadas en cuenta.

- Cuando se representan los resultados entregados por t-SNE, se observa un patrón de agrupaciones. Sin embargo, la interpretación de este patrón no debe hacerse de la misma forma que en un proceso de clustering. En t-SNE, ni el tamaño de los clusters, ni su densidad, ni su separación deben entenderse como indicadores de dispersión u otras características de la población con la que se está trabajando. El algoritmo de t-SNE actúa de tal forma que puede expandir clusters densos y compactar clusters dispersos, de ahí lo buenas que suelen ser sus visualizaciones.
- Este algoritmo calcula la probabilidad condicional de cada par de observaciones y trata de minimizar la suma de las diferencias entre las probabilidades de la dimensión superior e inferior. Esto implica una cantidad enorme de cálculos, lo que se traduce en un proceso muy lento cuando se trabaja con grandes cantidades de datos.
- Si bien t-SNE hace un trabajo excelente en la visualización, este funciona como una caja negra, esto quiere decir por ejemplo que no se tiene un set de ecuaciones que se puedan interpretar como lo es en el caso de PCA.
- Este algoritmo no es determinista, por lo que, para un mismo set de datos, existe la posibilidad de obtener resultados levemente diferentes.
- Finalmente si bien este algoritmo fue diseñado para la visualización de datos, este se puede usar para la reducción de dimensionalidad, pero depende mucho de las características de nuestros datos y de la cantidad de dimensiones que se usen, por lo que solo debe ser usado para reducir la dimensionalidad a 2 o 3 dimensiones, de lo contrario el comportamiento de este algoritmo se torna impredecible. En general no es muy recomendable usarlo para este fin.

3.6. Metodología CRISP-DM

La metodología CRISP DM [45] (Cross Industry Standard Process for Data Mining), es una de las más utilizadas para gestionar proyectos de minería de datos, proporcionando una visión general del ciclo de vida de este tipo de proyectos. Dentro del ciclo de vida que plantea este método tenemos 6 fases que definen tareas específicas y resultados esperados para cada una.

En la figura 6 las flechas entre fases representan las dependencias mas importantes y frecuentes que estas tienen entre si, mientras que el circulo de la parte exterior representa la naturaleza cíclica de la minería de datos. Cada una de las fases de detalla a continuación.

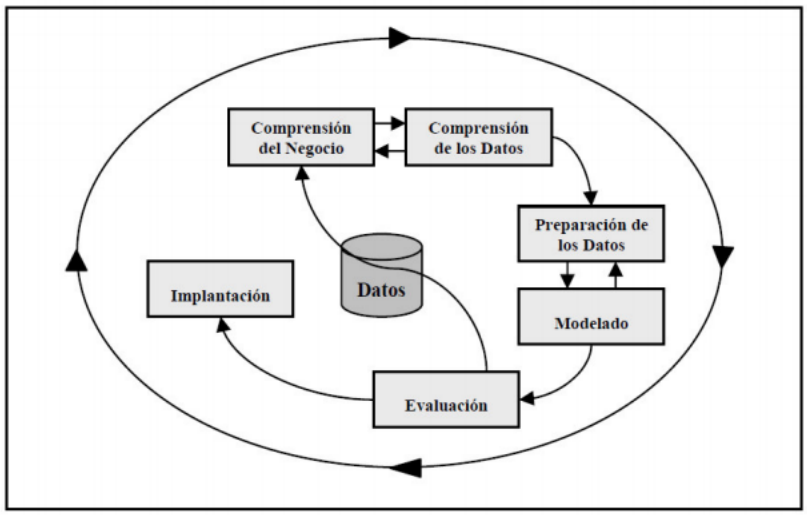

Figura 6: Ciclo de vida de la metodología CRISP DM.

1. Comprensión del negocio: Esta fase es la más importante del ciclo, y consiste en comprender los objetivos y requisitos del problema, además de crear un plan inicial para alcanzar los objetivos. La importancia de esta fase radica principalmente en que si no se tiene una idea clara del problema por muy sofisticados que sean los algoritmos permitirán obtener resultados fiables.

2. Comprensión de datos: En esta fase se realiza la recopilación de datos, se identifican problemas relacionados con la calidad de estos, se formulan las primeras hipótesis sobre estos en base a subconjuntos o patrones que se logren detectar.

3. Preparación de datos: En esta etapa se cubren todas las actividades para construir el conjunto de datos final, a partir del conjunto de datos en bruto. Dentro de las tareas más recurrentes tenemos, selección y transformación de atributos, limpieza de datos, utilización de herramientas de modelado, entre otras. Las tareas de preparación pueden realizarse varias veces y sin un orden en específico.

4. Modelado: En esta fase se aplican varias técnicas de modelado y se calibran sus parámetros para obtener óptimos resultados.

5. Evaluación: En esta etapa se han creado uno o varios modelos que parecen ser de alta calidad. Antes de realizar el despliegue final es importante evaluar mas a fondo el modelo seleccionado y revisar los pasos ejecutados para construirlo.

6. Despliegue: En esta fase se deberá definir como implementar los resultados. Dicha implementación puede tan simple como generar un informe o tan compleja como implementar un proceso repetible de minería de datos.

CAPÍTULO 4
PROPUESTA DE SOLUCIÓN

Para abordar el problema descrito en el capítulo 1, se plantea un proceso basado en la metodología **CRISP-DM** presentada en la sección 3.6. La elección de esta metodología como guía, se debe a que proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, junto con una flexibilidad para el paso entre fases, implicando que se permita movimiento hacia adelante y hacia atrás entre estas. Todo esto facilita la creación y aplicación de modelos para la resolución del problema.

El proceso propuesto consta de siete etapas, que van desde la obtención de los datos hasta la búsqueda de grupos. Debido a la complejidad del problema, se plantea que el proceso tenga dos grandes partes (figura 7) que solucionen de forma independiente la **Detección de estrellas variables de largo periodo** y la **clasificación según sus propiedades físicas**.

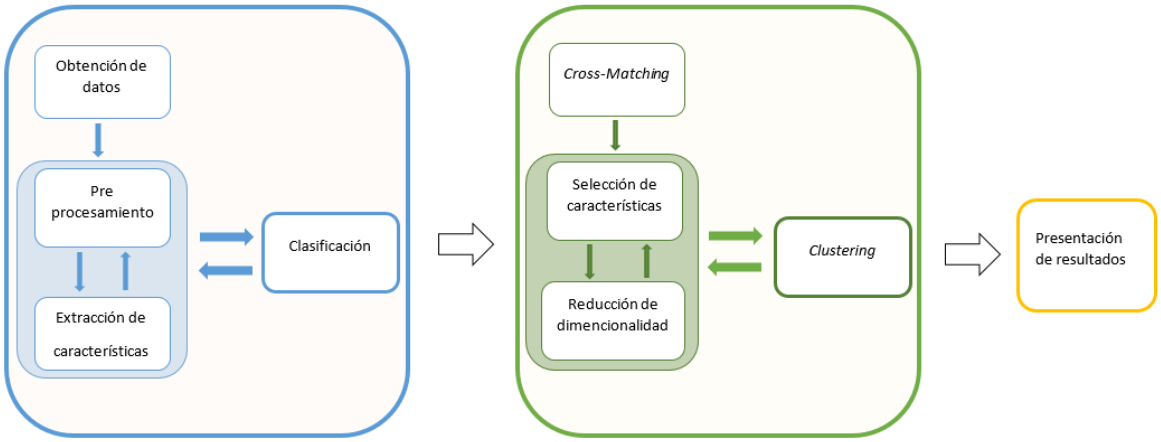

Figura 7: Pipeline completo del proceso propuesto.
Fuente: Elaboración propia.

Las técnicas usadas en este proceso pueden variar dependiendo de quien quiera replicar el pipeline y de la información disponible en el momento, es por esto que lo presentado en esta sección será el procedimiento general para cada etapa, junto con algunas consideraciones que se realizaron para su posterior implementación, además de proponer alternativas por cada una de estas.

4.1. Parte 1: Detección de estrellas variables de largo periodo

La primera parte tiene como objetivo final el realizar una clasificación para cada una de las estrellas disponibles en base a la información provista por las curvas de luz, es decir el output final de esta parte será un conjunto de estrellas que sean efectivamente variables en el largo

periodo. Las etapas que aquí se abarcan serán la obtención de datos preprocesamiento de estos, extracción de características y el proceso de clasificación.

4.1.1. Obtención de datos

EL objetivo de esta etapa, consiste en definir las fuentes de datos a utilizar, para luego identificar el formato y organización en que la data es entregada. Por lo anterior, esta presenta analogías con las etapa de comprensión de datos planteada por la metodología **CRISP-DM**.

Para poder estudiar el comportamiento variable en el largo periodo es necesario contar con una gran cantidad de curvas de luz que que estén construidas por observaciones a lo largo de varios años.

Cabecera	
ndata	Número de mediciones de cada dataset.
dataset	Identificador del dataset.
RA	Ascensión recta.
DEC	Declinación
desig	Designación que hace ASAS en el último dígito del RA y DEC por cada dataset (puede variar solo en 1 dígito).
Mediciones	
HJD	Indica la fecha en la que se hizo la medición en días julianos (HJD-2450000).
MAG	Es la magnitud y se posee una por cada apertura. Se tiene 5 magnitudes en total.
MER	Es el frame error que describen la calidad fotométrica promedio de cada Frame. Uno por cada apertura.
Frame	Número de frame.
Grade	Indica la calidad de la medición. Los valores van desde grade A a D

Tabla 4: Estructura de los datasets entregados por ASAS
Fuente: Elaboración Propia.

Uno de los proyectos astronómicos que cumple con estas características es ASAS, el cual cuenta con observaciones a lo largo de 9 años y su catalogo AASC (Asas All Stars Catalog) está constituido por cerca de 15 millones de estrellas, por lo que se perfila como la opción idónea para ser la fuente de datos que nos permita estudiar el comportamiento variable en

el largo periodo.

Para obtener los datos desde esta fuente, las alternativas de descarga que ofrece ASAS son bastante limitadas, por un lado tenemos a disposición el catalogo completo de estrellas variables (50124 curvas de luz), mientras para el resto de estrellas la única opción es hacer la descarga manual de forma individual usando su ASAS ID.

# HJD	MAG_1	MAG_0	MAG_2	MAG_3	MAG_4	MER_1	MER_0	MER_2	MER_3	MER_4	GRADE	FRAME
1868.69967	11.472	11.489	11.471	11.469	11.455	0.040	0.070	0.031	0.036	0.040	B	107
1869.68954	11.431	11.409	11.444	11.448	11.448	0.041	0.060	0.032	0.035	0.040	B	266
1870.69720	11.375	11.386	11.381	11.392	11.403	0.047	0.085	0.037	0.041	0.046	D	417
1871.70101	11.312	11.308	11.325	11.328	11.317	0.045	0.081	0.036	0.040	0.045	D	549
1872.69700	11.300	11.335	11.313	11.321	11.317	0.044	0.082	0.037	0.041	0.046	D	708
1874.68814	11.264	11.225	11.290	11.307	11.282	0.042	0.076	0.034	0.038	0.045	D	1028
1875.68822	11.284	11.233	11.284	11.289	11.297	0.040	0.060	0.031	0.035	0.040	B	1188
1876.68702	11.277	11.281	11.300	11.315	11.333	0.041	0.067	0.032	0.036	0.040	B	1349
1877.68740	11.295	11.258	11.314	11.334	11.343	0.036	0.053	0.027	0.030	0.034	A	1509
1878.67711	11.351	11.305	11.347	11.354	11.364	0.034	0.049	0.027	0.030	0.033	A	1662
1879.68046	11.335	11.339	11.339	11.341	11.335	0.036	0.047	0.027	0.030	0.034	A	1819
1881.67115	11.391	11.378	11.398	11.420	11.423	0.037	0.049	0.028	0.031	0.036	A	2129
1884.65046	11.478	11.457	11.490	11.506	11.530	0.039	0.053	0.030	0.034	0.038	A	2269
1885.66733	11.612	11.635	11.620	11.633	11.618	0.038	0.046	0.027	0.030	0.035	A	2436
1886.66870	11.679	11.613	11.691	11.704	11.726	0.038	0.049	0.028	0.031	0.035	A	2598
1887.65960	11.671	11.615	11.701	11.720	11.744	0.038	0.051	0.029	0.032	0.037	A	2753
1888.65902	11.652	11.651	11.679	11.709	11.718	0.039	0.058	0.030	0.035	0.040	A	2912
1889.65569	11.618	11.647	11.641	11.661	11.624	0.036	0.048	0.029	0.032	0.036	A	3072

Figura 8: Extracto de uno de los datasets de la estrella 045914-6935.8
Fuente: Asas.

El Asas ID de cada estrella es una composición de la Ascensión recta (RA) [4] y la declinación (DEC)[5] de la forma RA+DEC. Para solucionar este problema es necesario contar con un script que automatice la descarga o solicitar los datos directamente a alguno de los administradores del sito. Finalmente la estructura de cada archivo entregado por ASAS es similar a la descrita en la Tabla 4.

4.1.2. Preprocesamiento

Esta etapa es análoga a la fase de preparación de datos propuesta por la metodología **CRISP-DM**, ya que se busca hacer una limpieza y tratamiento de los datos en bruto de cada curva de luz obtenida previamente. Para logra esto, como primer paso es necesario quitar las mediciones que contengan un error fotométrico alto[6]. Como se pudo ver en la tabla 4 y en la figura 8 ASAS entrega mediante el parámetro GRADE una medida que indica la calidad de la medición, siendo los valores A y B los que corresponden a una calidad buena y promedio, mientras que un grade C significa que la estrella era demasiado tenue para registra un valor

[4]La ascensión recta es una de las coordenadas astronómicas que se utilizan para localizar los astros sobre la esfera celeste, equivalente a la longitud terrestre

[5]La declinación es el ángulo que forma un astro con el ecuador celeste. Es comparable a la latitud geográfica que se mide sobre el ecuador terrestre

[6]Error en la medición del brillo de la estrella debido a errores derivados de la calibración del instrumento o factores externos.

(se registra 29.99) y un grade D indica que la medición provenía de un frame de mala calidad, por lo que estas medidas deben ser descartadas. Como paso adicional una vez eliminados las mediciones con un grade C y D, es recomendable hacer algún tipo de tratamiento de outliers para cada una de las magnitudes. Una vez hecho esto se debe realizar la selección de la o las aperturas con las que se va a trabajar. Como se puede apreciar en la figura 8 ASAS entrega mediciones para 5 aperturas distintas, las cuales van desde los 2 píxeles (30'') hasta los 6 píxeles (90'') junto con sus correspondientes errores. A groso modo tenemos que la diferencia entre cada una de estas radica en la cantidad de luz que el instrumento logra captar, esto significa que cuando la apertura es pequeña existe la posibilidad de que no se logre captar todo el brillo de la estrella, generando un mayor error de poisson, mientras que cuando la apertura es grande es más probable que la medición tenga un mayor ruido de fondo debido al brillo de las estrellas vecinas, por lo que se debe seleccionar la apertura que contenga un menor error promedio.

4.1.3. Extracción de características

El objetivo de esta etapa, es estudiar y analizar el comportamiento de cada curva de luz, con el fin de encontrar un modelo que nos permita caracterizar la variabilidad presente en estas. Con esto se pretende obtener variables que ayuden a generar algún criterio de selección que pueda ser utilizado posteriormente en el proceso de clasificación. Esta etapa es análoga con la fase de modelado de la metodología **CRISP-DM** presentada en la sección 3.6.

Para el desarrollo de esta memoria, se hará uso del método propuesto en el trabajo realizado por Javiera Godoy [39] que fue utilizado previamente para estudiar la variabilidad de largo periodo en curvas de luz obtenidas desde ASAS. La metodología usada en este estudio consiste en verificar si existe una variabilidad a largo plazo mediante el análisis de las pendientes obtenidas a través de un ajuste lineal y un ajuste parabólico realizado a cada una de las curva de luz, considerando además la localidad de estas, ubicando a cada estrella dentro de una vecindad en la esfera celeste, a esta vecindad se le denominara **campo**. Con respecto a este punto podemos mencionar que la idea de utilizar el concepto de campo fue la solución usada en este trabajo, donde no se disponía de una gran cantidad de datos, por lo que no es la forma más generalizada de hacerlo. Una alternativa a esto es utilizar el método de los vecinos más cercanos presentado en la sección 3.1.1.

Los modelos para ambos ajustes están definidos a continuación

$$\begin{aligned} \text{Fit lineal:} \quad & m = (a_0 \pm e_{a0}) + (a_1 \pm e_{a1})t \\ \text{Fit parabólico:} \quad & m = (b_0 \pm e_{b0}) + (b_1 \pm e_{b1})t + (b_2 \pm e_{b2})t^2 \end{aligned}$$

Tenemos que m representa a la magnitud, mientras que a_0, a_1, b_0,b_1,b_2, son los coeficientes de cada uno de los ajustes, mientras que e_{a_0}, e_{a_1}, e_{b_0}, e_{b_1},e_{b_2} son los errores de cada coeficiente obtenido. Haciendo uso de las pendientes y sus respectivos errores, obtenemos los

indicadores Q'_1 y Q'_2 para cada ajuste

$$\text{Fit lineal:} \quad Q'_1 = \frac{a_1}{e_{a1}}$$
$$\text{Fit Parabólico:} \quad Q'_2 = \frac{b_2}{e_{b2}}$$

La idea detrás de esto es verificar que la pendiente supere su error, para así estar seguros de que efectivamente existe una variabilidad real. También como se menciono anteriormente cada curva de luz pertenece a un campo, por lo que es necesario normalizar estos indicadores por el promedio del campo al cual pertenece cada estrella, al hacer esto lo que se busca es identificar las estrellas que se escapen del comportamiento normal de su vecindad y posean una mayor variabilidad.

$$Q_1 = Q'_1 - \overline{Q'_1}$$
$$Q_2 = Q'_2 - \overline{Q'_2}$$

Finalmente este método incluye dos características adicionales que buscan categorizar el tipo de variabilidad presente en la curva de luz , para esto se analizan la dispersión de cada ajuste.

Desviación estándar de la magnitud

$$DEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Desviación estándar del fit lineal

$$DEFL = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(fl_i - y_i)^2}$$

Desviación estándar del fit parabólico

$$DEFP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(fp_i - y_i)^2}$$

Si la curva de luz presenta una variabilidad lineal la dispersión tendrá una tendencia a ser menor que la dispersión promedio, es decir $DEFL \leq DEP$, de igual forma si la curva de de luz presenta una variabilidad parabólica su dispersión será menor que la del fit lineal es decir $DEFP \leq DEFL$. Para medir esto se obtendrán los indicadores C_1 y C_2.

$$\text{Fit lineal:} \quad C_1 = 1 - \frac{DEFL}{DEP}$$
$$\text{Fit parabólico:} \quad C_2 = 1 - \frac{DEFP}{DEFL}$$

Como podemos ver este método nos entrega las características Q_1, C_1 asociadas al fit lineal y Q_2, C_2 asociadas al fit parabólico, que nos ayudaran a discriminar si las estrellas presentan algún tipo de variabilidad en el largo periodo.

El código necesario para realizar esta extracción de características, sin considerar la normalización por los campos, se puede encontrar en la sección 7.2 de los anexos.

Finalmente cabe mencionar que en la literatura existen métodos mucho más robustos usados ampliamente en astronomía como lo son la transformada de fourier, wavelets analysis o epoch folding [1] donde por lo general estas técnicas buscan encontrar periodos en la curva de luz. También se encuentran técnicas asociadas al estudio de las series temporales como las presentadas en [14, 17] que podrían ser utilizadas para extraer información, enfocándose en la variabilidad de estas, que posteriormente ayude a la clasificación.

4.1.4. Clasificación

El objetivo de esta etapa consiste en realizar el proceso de clasificación a cada estrella, en base a las características obtenidas a partir su curva de luz en el paso anterior, identificando solo las estrellas que cumplan con ser variables en el largo periodo. Dentro de los enfoques que podemos usar para realizar la clasificación se encuentran los siguientes

1. Clasificación informada o de criterio experto.
2. Clasificación supervisada.
3. Clasificación semi supervisada.

Dentro de estos enfoques lo ideal seria poder usar un método de clasificación automatizada, es decir que sea mediante el uso de algoritmos supervisados como los prestados en la sección 3.1 o el enfoque semi-supervisados definido en la sección 3.2, sin embargo al tratarse de un problema poco estudiado no se cuentan con ejemplos suficientes para poder realizar el entrenamiento de estos algoritmos. Es por esto que para el desarrollo de esta memoria se utilizara el enfoque de clasificación informada.

Los criterios de clasificación que se presentaran a continuación fueron expuestos en el trabajo de Javiera Godoy [39], estos criterios harán uso de los indicadores presentados en la sección 4.1.3. En este trabajo se comprobó de manera empírica que los mejores valores para establecer los criterios de clasificación de variabilidad lineal y parabólica son los siguientes

Criterio Lineal	Criterio Parabólico
$Q_1 \geq 4$	$Q_2 \geq 4$
$C_1 > 0{,}02$	$C_2 > 0{,}02$

Tabla 5: Criterios de clasificación
Fuente: Elaboración Propia.

Finalmente las estrellas que cumplan con estos criterios de clasificación serán las que se consideran para continuar con las siguientes etapas del pipeline. Por otro lado esta etapa puede ser análoga a las etapa de modelado y evaluación de la metodología **CRISP-DM**.

4.2. Parte 2: Clasificación de las estrellas variables de largo periodo en base a sus propiedades físicas

En el desarrollo de esta sección, se explicarán las diferentes etapas del proceso de clasificación no supervisado, que será aplicado al conjunto de estrellas variables de largo periodo y baja amplitud. Por otro lado, es importante recalcar, que el desarrollo de esta memoria está enfocado principalmente en esta etapa del pipeline, ya que la literatura sobre estrellas variables de largo periodo y baja amplitud es escasa. Debido a esto y como se menciono en la sección 3.3, la utilización de algoritmos de clustering, para realizar un análisis exploratorio de datos, que nos permita encontrar la estructura subyacente de estos sin ningún tipo de información previa, es una buena alternativa para determinar patrones que orienten el desarrollo de futuras investigaciones sobre este tipo de estrellas.

Finalmente, tenemos que las etapas involucradas en esta parte son las siguientes: Cross-matching, selección de características, reducción de dimensionalidad y clustering.

4.2.1. Cross-matching

El objetivo de esta etapa es conseguir una mayor cantidad de características que ayuden a construir una buena caracterización de cada estrella, lo que ayudara al posterior análisis. Esta esta al igual que la etapa de obtención de datos presentada en la sección 4.1.1 presenta similitudes con la etapa de comprensión de datos de la metodología **CRISP-DM**, ya que se buscan añadir nuevas features que entreguen una mayor información a cada estrella.

Como se mencionó en la sección 2.5 el trabajar con datos astronómicos muchas veces implica hacer uso de varias bases de datos para extraer información de diversa índole con respecto a un mismo objeto, para lo cual ya existen algoritmos que facilitan este procedimiento y observatorios virtuales que tienen disponibles servicios que permiten facilitar esta tarea.

Dentro de las múltiples alternativas para realizar el cruce de datos, tenemos una amplia ga-

ma de catálogos y herramientas que facilitan este proceso, donde podemos destacar los servicios provistos por Centre de Données astronomiques de Strasbourg (CDS): Vizier [27], Simbad [44] y X-Match. El primero de estos es uno de los servicio de catálogos astronómicos más populares, siendo el servidor de cerca de 7300 catálogos hasta la fecha. Por otro lado tenemos a Simbad, el cual no es un catálogo, si no que una meta-compilación hecha a partir de datos publicados en la literatura junto con identificaciones cruzadas. Gracias a esto, la base de datos de simbad entrega información como la naturaleza del objeto, sus coordenadas, magnitudes, movimientos propios, paralaje entre otros. Finalmente tenemos el servicio X-Match el cual tiene la facultad de hacer cross-matching entre dos catálogos distintos. Este servicio permite utilizar de forma complementaria los catálogos provistos por Vizier y la base de datos de Simbad. Otra alternativa para la obtención de datos, son las liberaciones de datos realizadas por la misión GAIA [36]. Esta misión es llevada a cabo por la agencia espacial europea, la cual pretende generar un mapa tridimensional de la vía láctea mediante una sonda espacial, además de generar datos precisos sobre las características de los distintos objetos celestes. Actualmente se han realizado dos liberaciones de datos.

Finalmente, en nuestro caso particular, para llevar a cabo el cruce de información, se utilizaron los servicios provistos por dos observatorios virtuales:

El primero de ellos es GAVO (German Astrophysical Virtual Observatory), donde se usó el catálogo HSOY (Hot Stuff for One Year) [3], el cual contiene datos con astrometria precisa para aproximadamente 583 millones de objetos. De este catálogo se pueden obtener parámetros de interés como lo son el movimiento propio y la magnitud. Este catálogo además incluye los respectivos errores para cada una de las características.

Como segunda herramienta tenemos VOSA (Virtual Observatory Sed Analyzer)[4], la cual permite hacer un análisis a cada estrella, permitiendo estimar diferentes parámetros como, la temperatura efectiva, luminosidad bolométrica, gravedad, masa y distancia.

4.2.2. Selección de características y reducción de dimensionalidad

Una vez realizado el cross-matching la dimensión de nuestros datos tendrá un aumento considerable por lo que, una alternativa será hacer una selección de características antes de pasar al proceso de clustering. En una primera instancia es recomendable hacer una preselección de los datos que sean significativos dentro de estas nuevas variables, eliminando elementos que sean redundantes o variables que estén demasiado correlacionadas. Una alternativa para esto, es hacer el filtro entre las variables que se usan para cuantificar la misma propiedad física, comparando sus errores y seleccionando la que tenga un mejor comportamiento. Con respecto a la relación que pueda existir entre variables una buena opción es hacer uso del índice de correlación de Pearson y construir una matriz de correlaciones para verificar la dependencia lineal que las variables puedan tener entre sí. El objetivo de hacer esto es verificar si existen variables que están fuertemente relacionadas, ya que si se da el caso se podría considerar solo una de estas.

Una vez hecha esta preselección si la dimensión continua siendo grande se puede recurrir a el uso de técnicas como las vistas en la sección 3.5 como PCA, para condensar la información entregada por estas variables en una cantidad menor.

Al hacer este análisis podremos mitigar el problema descrito en la sección 3.5.1 conocido como la maldición de la dimensionalidad el cual afecta de forma negativa el desempeño de los algoritmos de clustering. Esta etapa puede ser considerada similar a la etapa de preparación de los datos del método **CRISP-DM**.

4.2.3. Clustering

Esta es la etapa final del pipeline propuesto y consiste en utilizar el resultado de todas las demás etapas para aplicar algoritmos de clustering como los presentados en en la sección 3.3 y encontrar grupos en los datos que puedan ayudar a comprender por que existe un comportamiento variable de largo periodo en estas estrellas. Esta etapa puede ser análoga a las etapa de modelado y evaluación de la metodología **CRISP-DM**.

Método	Parámetros	Escalabilidad	Caso de uso	Geometría(métrica usada)
K-Means	Numero de *clusters*	Muy alta con n muestras, media con n *clusters*	• Uso general. • geometría plana.	Distancia entre puntos
Propagación por Afinidad	Amortiguamiento	No escalable con n muestras	• Muchos *clusters*. • tamaños de *clusters* desiguales. • geometría no plana.	Distancia entre grafos
Mean Shift	Ancho de banda	No escalable con n muestras	• Muchos *clusters*. • tamaños de *clusters* desiguales. • geometría no plana.	Distancia entre puntos
Clustering espectral	Numero de *clusters*	Media con n muestras, Pequeña con n *clusters*.	• Pocos *clusters*. • geometría no plana.	Distancia entre grafos
Clustereing jerárquico (Ward)	Numero de *clusters*	Alta con n muestras y n *clusters*.	• Muchos clusters • Posibles restricciones de conectividad.	Distancia entre puntos
Clustering Aglomerativo	Numero de *clusters*, tipo de enlace, distancia	Alta con n muestras y n *clusters*	• Muchos *clusters*. • posibles restricciones de conectividad. • Distancias no Euclidianas.	Cualquier métrica de distancia entre pares.
DBSCAN	Tamaño de la vecindad	Muy alta con n muestras, media con n *clusters*.	• Geometría no plana. • Tamaños de *clusters* desiguales. • Detección de datos atípicos.	Distancia entre puntos más cercanos.
Mixturas gaussianas	Número de componentes	No escalable	• Geometría plana • Bueno para estimar densidades.	Distancia Mahalanobis al centro
Birch	Factor de bifurcación, Umbral	Alta con n muestras y n *clusters*	• Grandes *datatsets*. • Eliminación de valores atípicos. • Reducción de datos.	Distancia Euclidiana entre puntos

Figura 9: Algoritmos de clustering disponibles en scikit learn
Fuente: Scikit Learn.

Algunos de los posibles algoritmos que pueden ser utilizados son presentados en la figura 9, donde es posible ver algunas de sus propiedades, como lo son la escalabilidad y bajo qué geometrías tienen un buen desempeño.

Con respecto a la selección del algoritmo lo ideal es escoger aquellos que tengan un buen comportamiento con una gran cantidad de datos. Esto quiere decir que el costo computacional sea adecuado y su tiempo de ejecución sea razonable. Algoritmos como DBSCAN, jerárquico aglomerativo, o K-means junto alguna de sus variaciones, pueden ser una buena opción debido que son bastante escalables al tamaño de la muestra y al número de grupos que se quiera utilizar, mientras que algoritmos como Gaussian mixtures, al ser poco escalables y computacionalmente costosos, su uso no es recomendado para grandes cantidades de datos. Otro factor a tener en consideración es seleccionar una medida de desempeño adecuada, para cuantificar qué tan buenas son los grupos encontrados por el algoritmo. En la sección 3.4 se presentaron los índices de Calinski-Harabaz, Silhouette, y Davies-Bouldin, los cuales están diseñados para evaluar el desempeño sin la necesidad de contar con ejemplos previamente etiquetados, por lo que es recomendable hacer uso de alguno de estos para medir el desempeño durante los experimentos.

Finalmente, para la elección del algoritmo a utilizar, una buena opción es considerar la relación entre los siguientes factores: La cantidad de grupos encontrados, cantidad de elementos que posee cada uno de estos y los valores obtenidos para la métrica de desempeño utilizada. Lo anterior es importante ya que, si se consideran por separado estos factores, puede darse el caso donde la métrica de desempeño sea alta, se disponga de una buena cantidad de grupos, pero la mayor cantidad de elementos esté concentrada en uno o dos. Por lo tanto es importante tener en cuenta los tres factores mencionados anteriormente para tomar una decisión que permita obtener un resultado coherente.

CAPÍTULO 5
RESULTADOS EXPERIMENTALES

5.1. Consideraciones previas

5.1.1. Ambiente de desarrollo

La solución propuesta en el capitulo 4 fue implementada usando el lenguaje de programación python en su versión 3.6 y las pruebas se realizaron haciendo uso de un jupyter notebook. Las librerías usadas para la implementación son las siguientes:

Propósito general	Manejo de datasets	Gráficos	Machine learning
Numpy	Pandas	Pyplot	Scikit learn
Scipy		Seaborn	Scikit learn extensions

Tabla 6: Librerías utilizadas.
Fuente: Elaboración Propia.

5.1.2. Datos Utilizados

Los datos utilizados en esta memoria fueron entregados por la astrónoma **Amelia Bayo** utilizando como fuente de datos ASAS. Este conjunto está conformado solo por estrellas visibles en el hemisferio sur y que se encuentren fuera del plano galáctico[7] para evitar obtener curvas de luz que contengan un ruido alto. Además, solo se consideraron curvas de luz que contengan al menos 400 mediciones y que la magnitud promedio sea de a lo más 12 magnitudes, para evitar errores fotométricos altos. También tenemos que los datos venían procesados según el trabajo de Javiera Godoy [39], implicando lo siguiente:

- Se selecciona una apertura con la cual trabajar. La elección de esta fue se llevo a cabo en base al error promedio, seleccionando la que tenga el error promedio menor. Por lo tanto, se usaran los datos que entrega esta apertura.
- Se eliminan las mediciones defectuosas (magnitudes iguales 29.999 y 99.999), y posteriormente se eliminan de cada curva de luz, las mediciones que sobrepasen la magnitud promedio más/menos tres desviaciones estándar.
- Se realiza el cálculo de los estadísticos Q1, Q2, C1, C2 presentados en la sección 4.1.3.

[7] Plano donde se encuentran la mayor cantidad de estrellas de una galaxia con forma aplanada. Este plano pasa por el centro de masa de la galaxia.

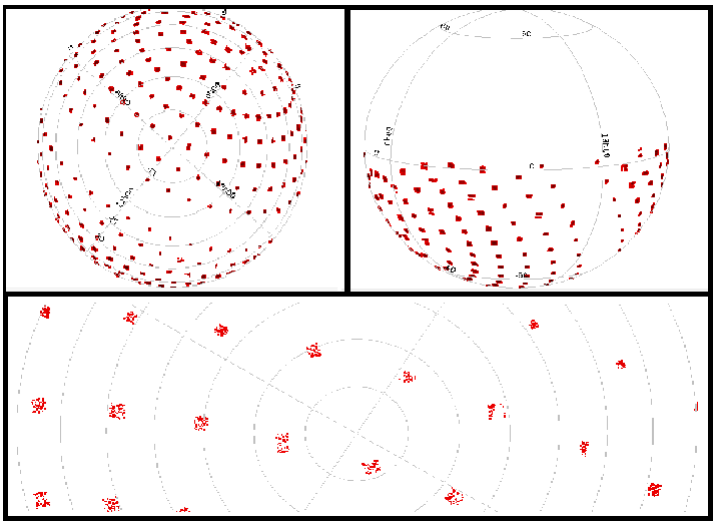

Figura 10: Diagrama de la esfera celeste con los campos utilizados.
Fuente: Elaboración propia

El dataset además incluye el cruce de datos, el cual fue realizado utilizando los servicios provistos por VOSA y el catalogo HSOY del observatorio virtual GAVO. Todo lo anterior implica que las etapas de **obtención de datos**, **preprocesamiento**, **extracción de características** y **cross-matching** se encuentran cubiertas.

Finalmente la cantidad de datos entregados corresponde a la información para **11317** estrellas presentes en 236 campos.

5.2. Clasificación

El proceso de clasificación se realizo utilizando el método propuesto en la sección 4.1.4. La cantidad de estrellas resultantes que se obtuvo posterior al aplicar este método corresponde a un $4{,}67\,\%$ de la muestra total, cifra que equivale a un total de **529 estrellas**.

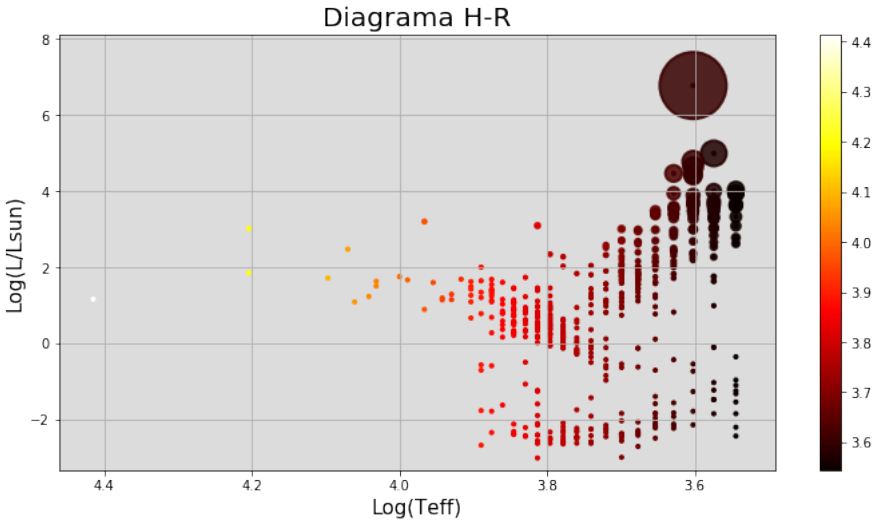

Figura 11: Diagrama H-R para las estrellas variables de largo periodo
El color de cada punto representa la temperatura efectiva, mientras que el tamaño representa el radio de la estrella.
Fuente: Elaboración propia

Si observamos la distribución de las estrellas seleccionadas en la figura 11, es posible identificar tres grupos claramente definidos. El primero de ellos (sector superior derecho) corresponde a estrellas de gran tamaño que se encuentran fuera de la secuencia principal, pudiendo caer en la categoría de gigantes y supergigantes, el segundo grupo (centro) corresponde a estrellas que están en la parte media/baja de la secuencia principal y el tercer grupo suponemos que puede calzar con algunas estrellas presentes en la parte más baja de la secuencia principal (enanas rojas) y otras pertenecientes a la categoría de enanas blancas. Finalmente podemos ver que, dentro de las estrellas clasificadas como variables de largo periodo, existe la ausencia de estrellas pertenecientes a la parte alta de la secuencia principal.

5.3. Selección de características y reducción de dimensionalidad

Como se menciono en la sección 5.1, para el desarrollo de este prototipo se dispone de una cantidad limitada de datos, los cuales incluyen una gran cantidad de características, que nos entregan información con la cual podemos trabajar. Sin embargo como se explico en la sección 4.2.2 es necesario discriminar entre las que nos sean útiles y las que no, para evitar problemas relacionados con la alta dimensionalidad.

5.3.1. Selección de características

Como primer paso es necesario quitar las variables categóricas que sirven para identificar cada estrella (ASAS ID, RA, DEC, latitud galáctica, longitud galáctica, etc), ya que no aportan ningún valor a la hora de estudiar el comportamiento de cada una de estas. El realizar este paso nos entrega las variables presentadas en la tabla 7. En las etapas iniciales del desarrollo de este prototipo, se realizaron numerosas pruebas considerando todas o la gran mayoría de las variables presentadas en la tabla 7, y se buscó realizar una reducción de dimensionalidad utilizando PCA, lo cual no trajo buenos resultados. Es por esto que en los experimentos posteriores se optó por hacer una selección previa de características, antes de aplicar PCA.

Característica	Descripción	Unidades
Ftot	Flujo total	$\frac{erg}{cm^2/s}$
Teff	Temperatura efectiva	Kelvin
Lbol	Luminosidad bolometrica	Unidades solares
Rad1	Radio de la estrella. Calculado usando $Md = \frac{R}{D}^2$	Unidades solares
Rad2	Radio de la estrella. Calculado usando $Lbol = 4\pi R^2 \sigma T_{eff}^4$.	Unidades solares
M1	Masa de la estrella. Calculada usando $g = \frac{GM}{R^2}$.	Unidades solares
M2	Masa de la estrella . Calculada usando $g = \frac{GM}{R^2}$, con R dado por el usuario.	Unidades solares
D(pc)	Distancia	Parsec
pm	Movimiento propio	$\frac{mas}{yr}$
Mean mag	Magnitud media	Unidades de magnitud.
AV	Extinción en rango visible.	Unidades de magnitud.
Meta	Metalicidad.	$\frac{Fe}{H}$
logg	Logaritmo de la gravedad superficial	$\frac{m}{s^2}$
Q1	Medida para la variabilidad lineal	Sin unidades.
Q2	Medida para la variabilidad parabólica	Sin unidades.
C1	Medida de dispersión para criterio lineal	Sin unidades.
C2	Medida de dispersión para el criterio parabólico	Sin unidades.

Tabla 7: Características disponibles
Fuente: Elaboración propia

La tabla 7 tiene un total de 17 variables donde varias incluyen sus errores (los cuales se omitieron en dicha tabla). De estas varias hacen referencia a lo mismo, por ejemplo tenemos

que Rad1 y Rad2 hacen referencia al radio, mientras que M1 y M2 hacen referencia a la masa de la estrella. Por otro lado tenemos varias variables que entregan una métrica para el brillo como lo son Ftot, Lbol y Mean mag. Con respecto a estas variables solo de debe escoger una para cada característico que representan para evitar trabajar con variables redundantes. Para escoger la masa y el radio que usaremos lo haremos en base a su error

Variable	Promedio	Min	25 %	50 %	75 %	Max
eRad1	1124.97	0	0.04	0.25	2.64	561251.25
eRad2	1142.78	0	0.10	0.34	3.68	569915.69
eM1	4.65e+04	7.93e-09	2.46e-03	1.67e-01	6.11	2.05e+07
eM2	4.83e+04	9.24e-09	2.48e-03	1.80e-01	6.16	2.11e+07

Tabla 8: Errores para las variables radio y masa
Fuente: Elaboración propia

Como se puede observar en la tabla 8 el comportamiento del error es bastante similar, sin embargo se logra observar un mejor comportamiento leve para las variables Rad1 y M1, por lo que estas serán las variables que se usaran para el radio y la masa respectivamente. El siguiente paso será escoger la variable que represente el brillo. La luminosidad bolométrica se encuentra en unidades solares, por lo que se escogerá esta debido a que se hace más intuitiva su interpretación al tener como punto de comparación el sol.

Finalmente se analizará la relación que tienen las variables preseleccionadas entre sí mediante el coeficiente de correlación de Pearson. Este índice se mueve en el rango de valores [-1,1], donde los valores cercanos a 1 indican una correlación directa, es decir si una variable aumenta la otra siempre lo hará, valores cercanos a 0 indican una correlación nula y valores cercanos a -1 indican una relación inversa, es decir si una variable aumenta la otra disminuirá.

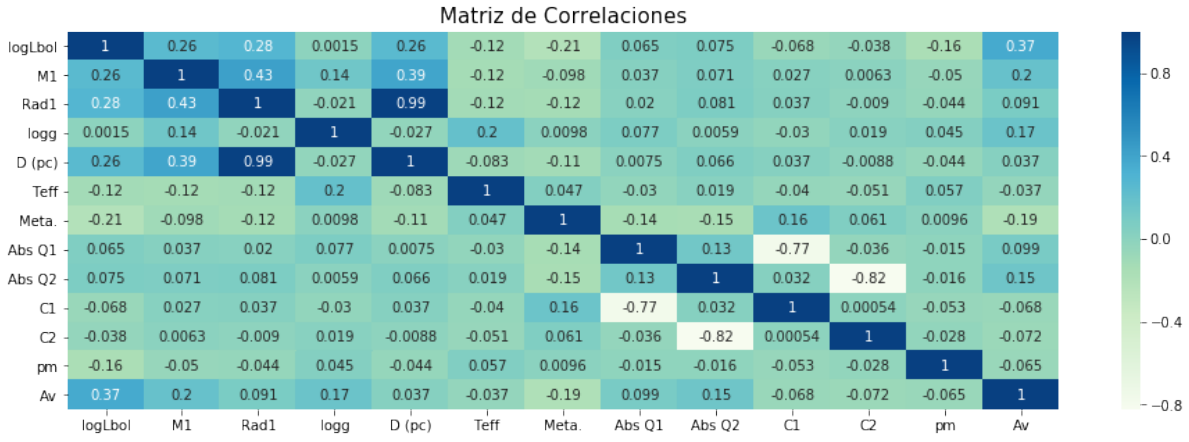

Figura 12: Matriz de correlaciones usando las variables preseleccionadas
Fuente: Elaboración propia

Como podemos apreciar las características Q1 y Q2 obtenidas por los criterios presentados en la sección 4.1.3 tienen una correlación casi nula con el resto de variables exceptuando sus valores asociados C1 y C2, los cuales tienen una correlación inversa bastante alta. Por otro lado podemos notar que el resto de variables mantienen una correlación directa débil o nula exceptuando el **radio** y la **distancia** que tienen una dependencia lineal directa casi perfecta. Con lo anterior podemos decir que las variables resultantes del proceso de extracción de características no afectan prácticamente en nada al resto de variables, por lo que su propósito para efectos de este prototipo, está acotado sólo a la clasificación y selección de estrellas, es por esto que para las etapas posteriores son prescindibles. Por otro parte tenemos que el radio y la distancia tienen una correlación lineal directa prácticamente perfecta, es decir, bastará usar solo una de estas variables. Durante la sección 2.3 se hizo énfasis en la importancia que el diagrama H-R tiene en la astronomía para entender el comportamiento de las estrellas según evolución estelar, por lo que con el fin de utilizar una estructura conocida para astrónomos y astrofísicos se dará prioridad en el uso de las variables que están directamente relacionadas con la estructura del diagrama.

Una vez explicado este punto tenemos que las variables que usaremos serán las siguientes:

- Logaritmo de la luminosidad bolométrica.
- Temperatura efectiva.
- Radio.
- Masa.
- Movimiento propio.

La idea es hacer el proceso de reducción de dimensionalidad con estas variables y posteriormente, como se verá en la siguiente sección, aplicar algoritmos de clustering para encontrar grupos que sean lo más definidos posibles. Es necesario mencionar que esta elección es particular para este prototipo, lo cual puede cambiar si la cantidad de datos o características disponibles es mayor.

5.3.2. Reducción de dimensionalidad

Para realizar la reducción de dimensionalidad usaremos el análisis de componentes principales (PCA), lo que nos permitirá encontrar p componentes principales que condensan las información de las n variables originales en un menor número, es decir $p \leq n$. Como paso previo a aplicar la transformación es necesario hacer una normalización de los datos, ya que pca es sensible a la escala de estos. Una vez hecho esto procedemos a aplicar la transformación para $p = n$ componentes principales, esto con el fin de ver cuántas componentes debemos usar. La forma de determinar lo anterior es usando el porcentaje de la varianza

explicada acumulada, donde se seleccionan la cantidad de componentes que expliquen entre el 85-90 por ciento de la varianza total, así lograremos que nuestra representación de los datos sea lo más fiel a los datos originales.

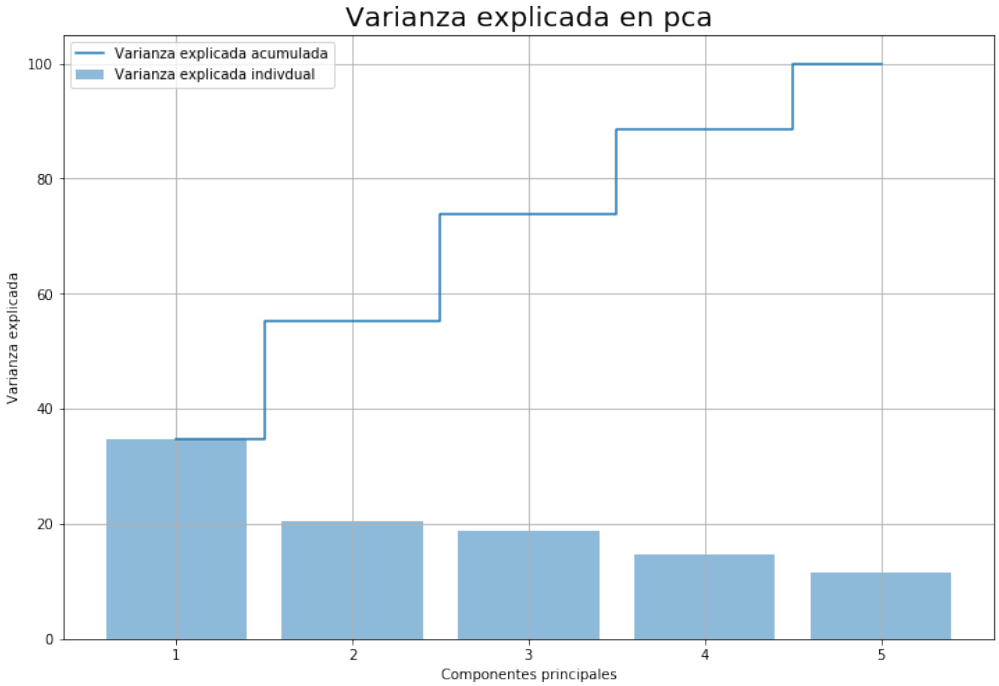

Figura 13: Varianza explicada acumulada e independiente por componente principal
Fuente: Elaboración propia

	PC1	PC2	PC3	PC4
Rad1	26.991622	15.870062	7.842266	12.580433
M1	26.650126	15.695432	6.404888	19.769623
pm	9.540331	47.249328	14.875501	21.396367
Teff	13.123177	10.037081	59.854121	0.690126
logLbol	23.694744	11.148097	11.023224	45.563451

Figura 14: Porcentaje de cada variable por cada componente principal
Fuente: Elaboración propia

Como podemos ver reduciendo una dimensión obtenemos una representación que explica aproximadamente el 85 % de la data original, por lo que usaremos Cuatro componentes principales para la siguiente etapa.

Para poder visualizar este resultado y a modo de comparación aplicaremos t-SNE sobre la data estandarizada y la data transformada mediante PCA

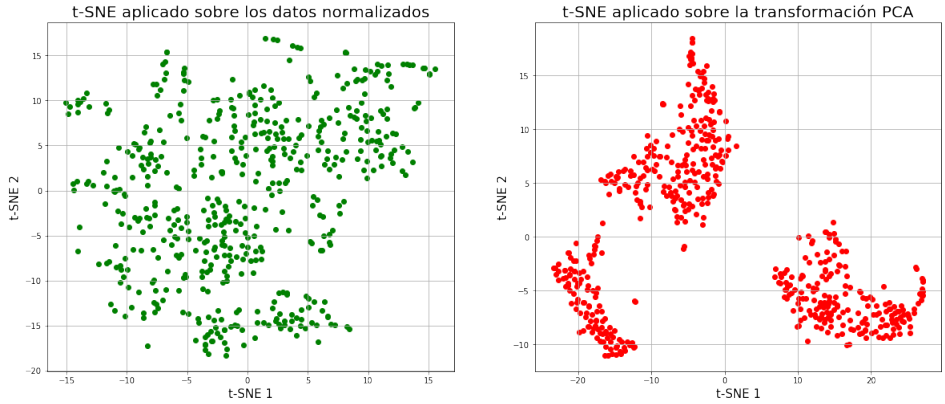

Figura 15: Visualización de los datos usando t-SNE
Fuente: Elaboración propia

Como podemos ver al estar transformados los datos usando PCA con 4 componentes principales, t-SNE logra entregar 3 grupos claros mientras que al no usar ningún tipo de transformación y considerar las cinco variables t-SNE entrega los datos con una gran dispersión. Esto nos puede indicar que la elección de transformar la data mediante PCA entregara un mejor resultado a la hora de aplicarlos respectivos algoritmo de clustering.

5.4. Clustering

En esta sección se estudiará el comportamiento para los algoritmos DBSCAN, k-means y jerárquico aglomerativo con el fin de escoger cual de estos conformará mejores grupos. Para poder evaluar el comportamiento de estos algoritmos usaremos el índice de silhouette, descrito en la sección 3.4. El principal motivo para usar este índice como medida de desempeño, es que este tiene bien definidas sus cotas lo que lo hace una buena opción a la hora de realizar comparaciones entre diferentes algoritmos/modelos, además al estar definido en el rango de puntajes entre -1 y 1 hace que su interpretación sea bastante intuitiva.

5.4.1. DBSCAN

DBSCAN es un algoritmo basado en densidad como se explicó previamente en el capítulo 3, este algoritmo recibe dos parámetros, donde el primero de ellos corresponde a la cantidad de elementos mínimos n que se requieren para formar el núcleo del cluster. El valor usado para este parámetro durante las pruebas corresponde a un valor porcentual de los datos, el que fijamos en un 2.5 %, es decir $n = 11$. El segundo parámetro corresponde al radio de vecindad que tiene cada punto. Este es un parámetro altamente sensible y difícil de sintonizar, por lo que se realizaron pruebas tomando un rango de valores entre $0{,}5 \leq \epsilon \leq 3{,}0$ con una separación de 0.075 entre cada valor, por lo que en definitiva las pruebas fueron realizadas para 34 valores de ϵ.

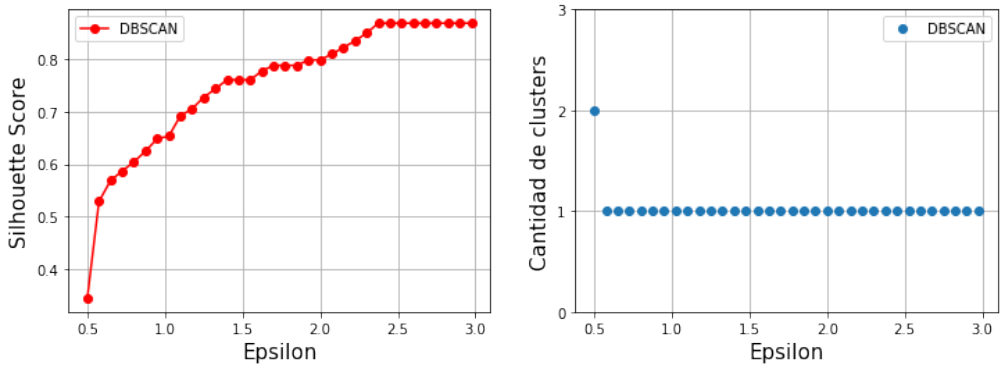

Figura 16: Gráficas comparativas para el valor del parámetro épsilon, tomando en cuenta el índice de silhouette y la cantidad de clusters encontrados por el algoritmo.
Fuente: Elaboración propia

Como podemos apreciar los resultados entregados por DBSCAN dejan mucho que desear debido a que los mejores puntajes obtenidos corresponden para 1 solo grupo y ruido, donde el puntaje va aumentando según cambia la cantidad de elementos que son detectados como ruido. Por otra parte el único resultado que nos dio más de un grupo fue a un radio bastante pequeño y un puntaje bastante bajo, solo encontrando dos grupos.

Además de esta prueba también se experimentó con el método propuesto en el trabajo " A New Approach to Determine Eps Parameter of DBSCAN Algorithm"[29], el cual está basado en las pendientes de la distancia de los k-vecinos más cercanos y que permite seleccionar de forma automática este parámetro.

Cantidad de puntos mínimos	11
Radio épsilon	2.31293888275
Clusters encontrados	1
Ruido	4

Tabla 9: Resultados obtenidos por DBSCAN

Como podemos ver DBSCAN en ambas pruebas no logro encontrar una estructura que nos permita hacer un análisis que nos ayude a entender el comportamiento de nuestros datos, debido a que la mayoría de los resultados obtenidos corresponden a 1 solo grupo y ruido, mientras que se logró encontrar un solo resultado con más de un grupo (2 grupos), pero la cantidad sigue siendo muy poca y su puntaje silhouette era demasiado bajo.

5.4.2. K-means

Este algoritmo a diferencia de DBSCAN, no tiene mayores complicaciones a la hora de fijar su parámetro k, ya que este puede ser seleccionado usando alguna medida de desempeño como las presentados en la sección 3.4. Dentro de las falencias que tiene este algoritmo, encontramos su alta sensibilidad a la presencia de outliers por lo que en caso de presentarse afectarían de forma negativa su desempeño.Para realizar las pruebas con este algoritmo, el parámetro k será determinado usando el índice de silhouette, donde se harán pruebas para un conjunto de valores y se escogerá el k que obtenga un mejor puntaje.

Aplicando k-means a toda la data obtenemos que el mejor resultado es para $k = 2$ con un puntaje cercano a 0.9, sin embargo para valores de k mayores este, el puntaje decae estrepitosamente, lo que nos hace sospechar de la presencia de outliers en nuestra data, los cuales distorsionan la distribución de los datos. Para solucionar este problema nos aprovecharemos del resultado obtenido por DBSCAN. Usando el método propuesto en [29] escogeremos de forma automática el valor de épsilon y aplicaremos DBSCAN para identificar los puntos de ruido y eliminarlos. Otra alternativa es utilizar la variante k-medians, la cual en vez de usar el promedio utiliza la mediana.

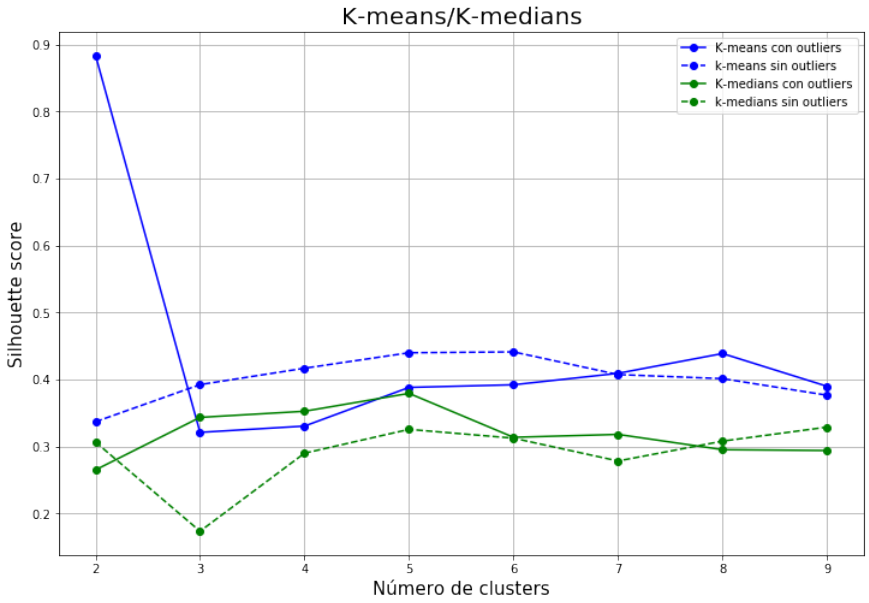

Figura 17: Gráfica comparativas para los algoritmos K-menas y K-medians.
Fuente: Elaboración propia

Como podemos ver la curva de rendimiento de k-means al quitar los outliers se estabilizó, donde el mejor valor obtenido fue para $k = 6$ con un índice de silhouette de 0.4413. Por otro lado tenemos que la variante k-medians con y sin outliers obtuvo puntajes bastante bajos por debajo de los 0.4.

5.4.3. Clustering jerárquico aglomerativo

Para las pruebas realizadas con este algoritmo se utilizaron estos criterios de linkage, average,ward y complete descritos en la sección 3.3.4 y en todos a excepción de ward (solo acepta la métrica euclidiana) se realizaron pruebas con las métricas de distancia euclidiana y manhattan.

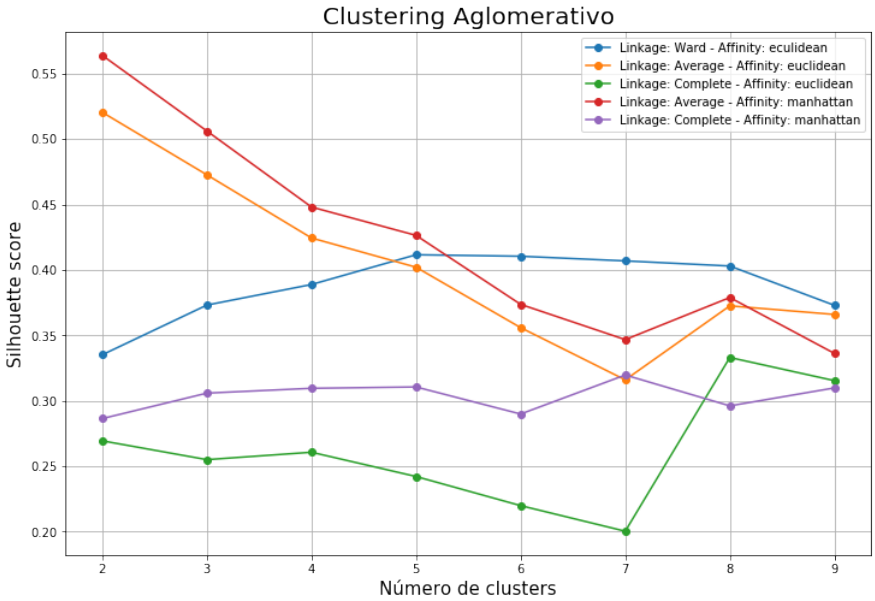

Figura 18: Clustering jerárquico aglomerativo con diferentes métricas de distancia y estrategias de vinculación

Fuente: Elaboración propia

Como se puede ver la estrategia ward se comporta similar a k-means pero se obtienen puntajes levemente más bajos. Por otro lado el enfoque average linkage presenta una curva descendente que se obtiene para las dos para las dos distancias utilizadas, sin embargo se logran obtener resultados sobresalientes hasta los 5 clusters. Finalmente el enfoque complete linkage es el peor de todos y para ambas métricas de distancia se obtienen puntajes por debajo de los 0.35.

5.4.4. Elección del algoritmo a utilizar

Si bien no se presentaron puntajes con un índice silhouette que indicara una estructura fuerte en los grupos, los mejores resultados son cercanos a un valor 0.5 lo cual es aceptable. Dentro de todos los algoritmos testeados tenemos que los mejores fueron k-means y el algoritmo jerárquico aglomerativo para sus enfoque avarage linkage con ambas métricas de distancia. Para escoger cual seleccionaremos veremos la combinación que nos entrega una

mejor relación entre el índice de silhouette la cantidad de clusters y la cantidad de elementos por cada uno de estos.

Algoritmo	estrategia	Distancia	Clusters	Silhouette score	Elementos por cluster
K-means	—-	Euclidiana	5	0.4401	/117/178/187/ /11/32/
K-means	—-	Euclidiana	6	0.4413	/180/178/115/ /28/11/13/
Aglomerativo	Ward	Euclidiana	5	0.4115	/61/183/112 /146/13/10
Aglomerativo	Ward	Euclidiana	6	0.4104	/156/61/112 /183/13/
Aglomerativo	Average linkage	Euclidiana	2	0.5204	/514/11/
Aglomerativo	Average linkage	Euclidiana	3	0.4726	/11/499/15/
Aglomerativo	Average linkage	Euclidiana	4	0.4244	/499/9/15/ /2/
Aglomerativo	Average linkage	Manhattan	2	0.5640	/515/10/
Aglomerativo	Average linkage	Manhattan	3	0.5061	/504/10/11/
Aglomerativo	Average linkage	Manhattan	4	0.4480	/11/10/20/ /484/

Tabla 10: Comparación entre algunos algoritmos de clustering

Como podemos ver los resultados entregados por el algoritmo jerárquico aglomerativo cuando se usa la estrategia de vinculación average tienden a entregar los puntajes más altos, sin embargo la distribución de elementos que hace para cada cluster es bastante dispareja, concentrando en todos los casos la gran mayoría de los datos en un solo grupo. Por otra parte K-means si bien entrega un puntaje menor los grupos encontrados se encuentran mejor distribuidos teniendo una mayor cantidad de grupos con ”densidades”similares. Finalmente usando el enfoque jerárquico con la estrategia de vinculación ward se logran obtener resultados muy similares a k-means con un índice de silhouette más bajo. Cómo se puede apreciar en las pruebas realizadas todos los algoritmos en general los puntajes no son muy buenos, donde el principal motivo puede ser la poca cantidad de datos disponibles o la cantidad de componentes principales que se están usando. Sin embargo el algoritmo que nos entrega una mejor relación entre el índice de silhouette y la distribución de los elementos por grupo es k-means por lo que usaremos los resultados entregados por este algoritmo. Con respecto al valor de k usado escogeremos $k = 6$.

5.5. Grupos encontrados

En esta sección presentaremos los grupos obtenidos al aplicar el algoritmo seleccionado en la sección 5.4.4. Como se describió el algoritmo seleccionado fue k-means aplicado para un valor $k = 6$. El objetivo de esta sección es estudiar el comportamiento de cada grupo según las variables mas importantes de la tabla 7. Como primer paso graficaremos el diagrama H-R destacando grupos encontrados para ver su distribución en este

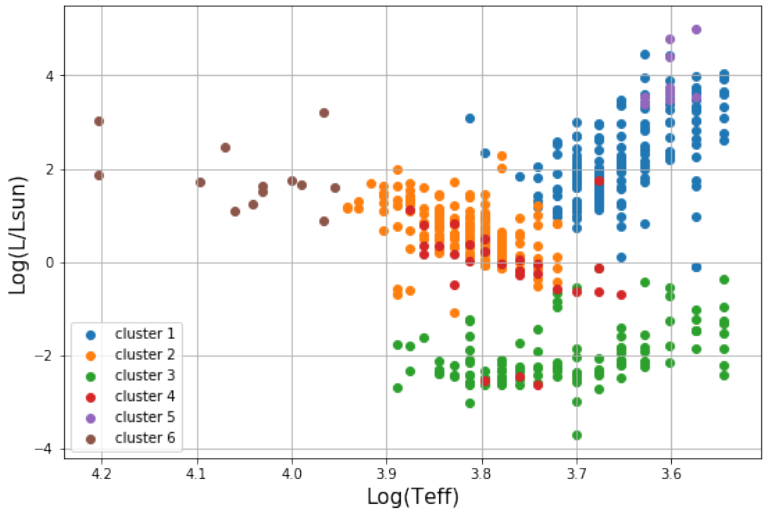

Figura 19: Distribución de los grupos encontrados
Fuente: Elaboración propia

Haciendo la comparación con el diagrama H-R de los datos sin agrupar visto en la figura 11, podemos notar que k-means logró agrupar los datos de forma esperable, encontrando una estructura similar a la que se lograba ver a simple vista. Podemos notar que el cluster 1 y 5 corresponden al grupo de estrellas gigantes/supergigantes, mientras que la secuencia principal serían los clusters 2,4 y 6 finalmente el cluster 3 correspondería a lo que se sospecha pueden ser enanas rojas/blancas.

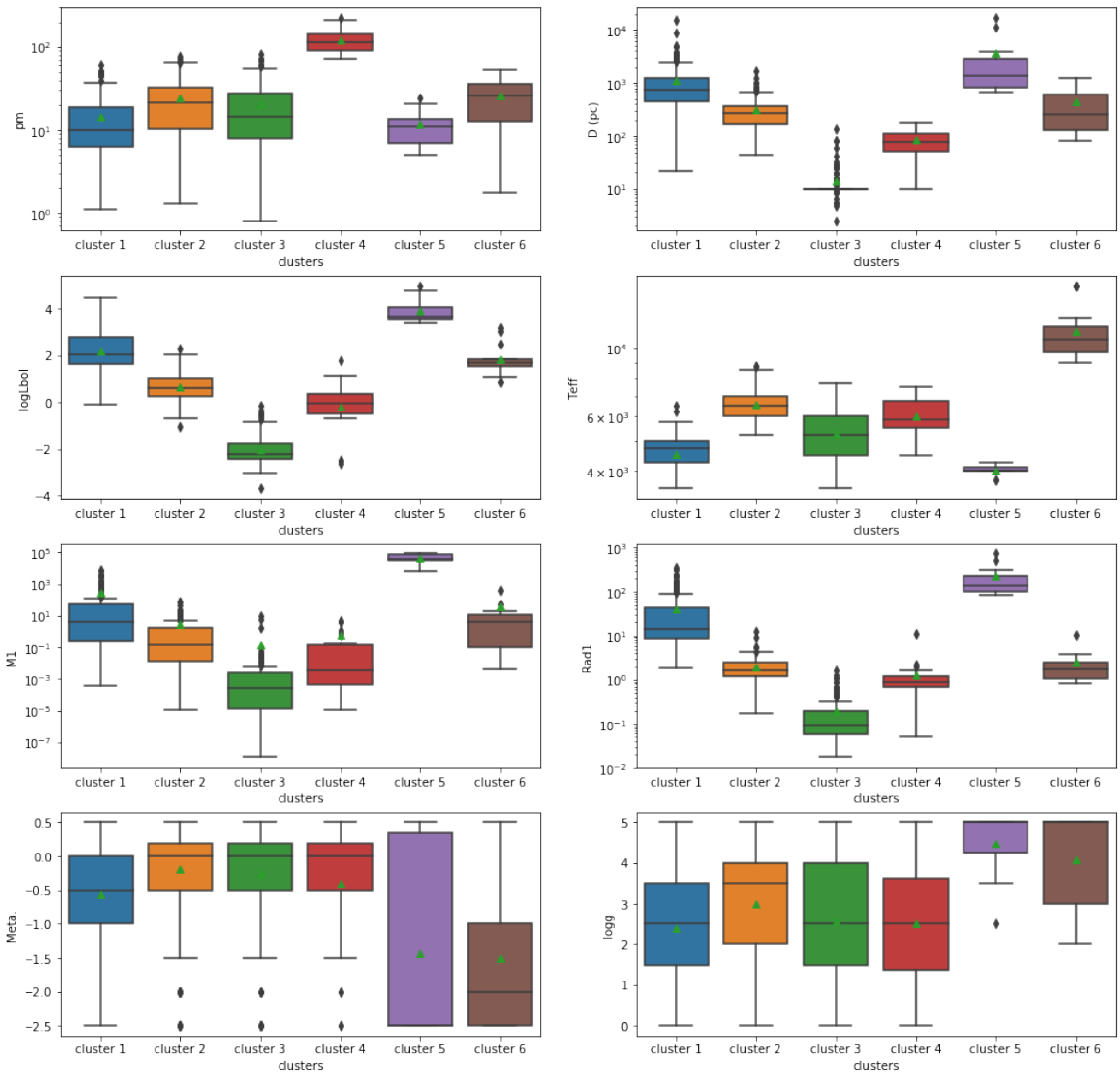

Figura 20: Boxplots para cada característica por cluster
Fuente: Elaboración propia

	Log(L/Lsun)	Masa	Radio	Log(g)	Distancia	Temperatura efectiva	Metalicidad	Movimiento propio	Extincion
Cluster1	2.188756	304.004150	40.815964	2.380556	1135.920532	4555.555664	-0.567222	14.120978	0.796111
Cluster2	0.644333	2.746593	1.970431	2.997191	315.763275	6603.932617	-0.189888	24.359865	0.415730
Cluster3	-2.026034	0.154441	0.203274	2.565217	14.519435	5267.391113	-0.268696	20.247919	0.565217
Cluster4	-0.183521	0.562654	1.283504	2.500000	84.639488	6017.856934	-0.400000	120.980141	0.532143
Cluster5	3.882822	42811.347656	227.148209	4.500000	3731.937256	4022.727295	-1.436364	11.624523	1.645455
Cluster6	1.817466	39.259342	2.520962	4.076923	442.244263	11346.154297	-1.500000	25.730175	1.292308

Figura 21: Valores promedio para cada característica por cluster.
Fuente: Elaboración propia

5.5.1. Cluster 1

Este es el grupo encontrado mas grande teniendo un total de 180 elementos. Dentro del diagrama H-R este grupo se encuentra por sobre la secuencia principal, donde habitan las estrellas de gran tamaño como lo son las subgigantes y gigantes rojas. Con respecto a sus características se puede apreciar el comportamiento esperable para estrellas pertenecientes a este segmento donde la relación entre una luminosidad radio y masa altos junto con una temperatura efectiva muy baja. Realizando la comparación con el resto de los grupos podemos notar que los valores promedios para las características recién nombradas son solo superados por el Cluster 5. Si vemos los boxplots para estas características podemos notar que hay una gran cantidad de outliers para el radio y la masa lo que nos puede indicar que hay varias estrellas perteneciente a este grupo que pueden estar mal clasificadas (Deberían pertenecer al Cluster 5).

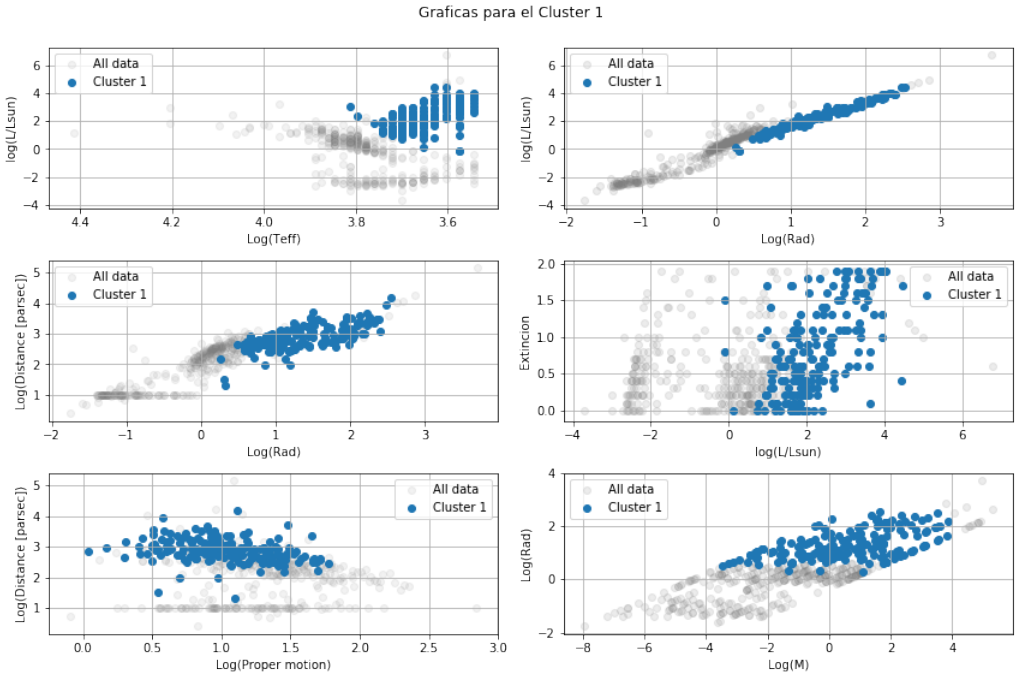

Figura 22: Gráficos para cluster 1
Fuente: Elaboración propia

Por otro lado si miramos el gráfico de **distancia vs radio** podemos corroborar la relación lineal casi perfecta mostrada en la figura donde a mayor radio mayor será la distancia, también se logra apreciar una relación lineal directa bastante fuerte entre la masa y el radio. Con respecto al movimiento propio, tenemos que este grupo presenta un valor promedio bastante bajo y si miramos el gráfico de **distancia vs movimiento propio** veremos que hay una tendencia lineal inversa entre estas características es decir a menor distancia el movimiento propio tiende a aumentar. Finalmente podemos mencionar que la extinción promedio es bastante alta y se puede ver una relación entre esta y valores altos en la luminosidad los

cuales corresponden a valores mayores a 2 .

5.5.2. Cluster 2

Este grupo al igual que el primero posee una gran cantidad de elementos llegando a un total de 178. Dentro del diagrama H-R este grupo está ubicado en la parte central de la secuencia principal por lo que en general los valores para algunas características debería estar cercanas a los valores que posee el sol.

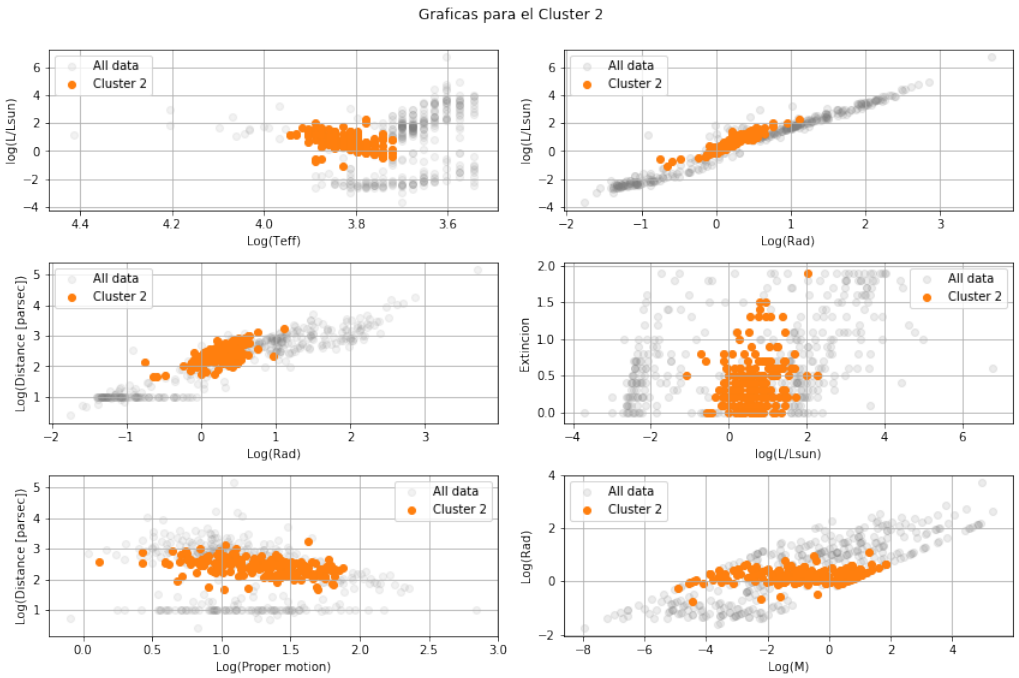

Figura 23: Gráficos para cluster 2
Fuente: Elaboración propia

Con respecto al comportamiento de sus características tenemos que los valores promedio para la luminosidad, radio, masa y temperatura efectiva son bajos y cercanos a los valores de nuestro sol como era esperable, además tenemos que el valor promedio de la gravedad es alto, siendo solo superado por el Cluster 5 y 6. Este comportamiento puede ser explicado al ver el gráfico de **radio vs masa** donde se puede apreciar que el valor del radio permanece casi invariable mientras que la masa se mueve en un rango bastante amplio. Por otro lado tenemos que el valor promedio para el movimiento propio no destaca pero si observamos el gráfico de **distancia vs movimiento propio** podemos ver la misma tendencia que en el Cluster 1, con una pendiente un poco menos pronunciada. Finalmente tenemos que la extinción tiene el valor promedio más bajo de entre todos los grupos y si vemos el gráfico de **luminosidad vs extinción** podemos ver que este grupo no supera las 2 unidades en luminosidad y que sus valores para la extinción están concentrados en valores menores a 0.5.

5.5.3. Cluster 3

Nuevamente encontramos un grupo bien definido con un total de 115 elementos, donde su ubicación en el diagrama H-R corresponde un grupo desconocido, que se presume está compuesto por estrellas pertenecientes a la parte más baja de la secuencia principal (enanas rojas) y estrellas pertenecientes al grupo de las enanas blancas. Dentro de las características de este grupo podemos encontrar que su luminosidad, masa, radio y distancia promedio son las más bajas de entre todos los grupos tomando valores por muy por debajo de los que tiene nuestro sol.

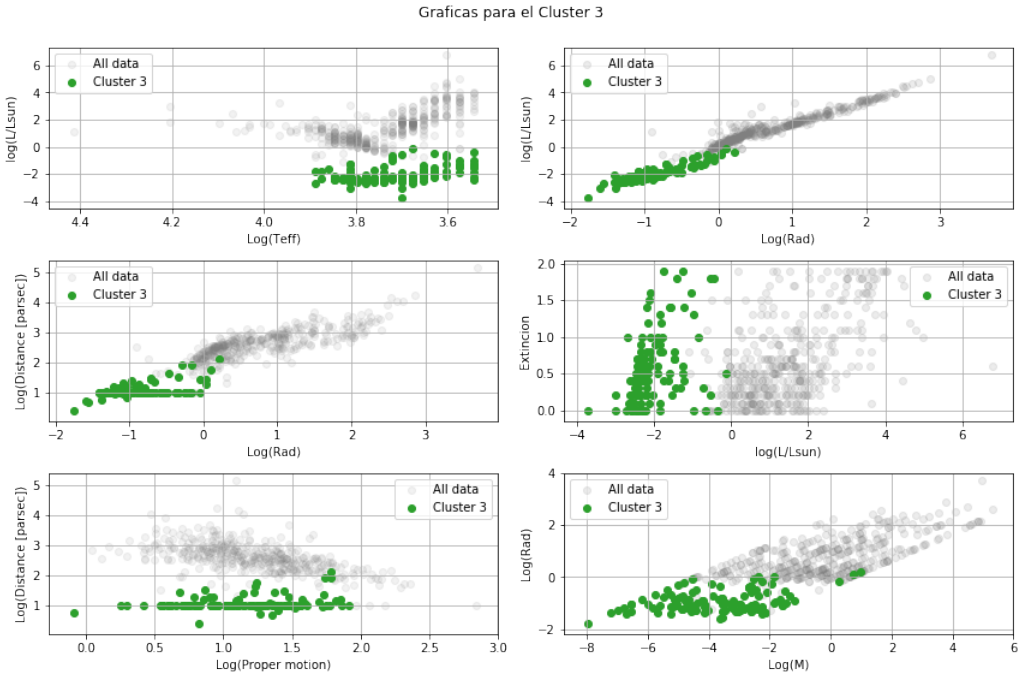

Figura 24: Gráficos para cluster 3
Fuente: Elaboración propia

Con respecto al movimiento propio podemos ver que a diferencia de los dos grupos anteriores donde se logra ver una relación entre esta característica y la distancia, aquí contrario a lo esperado vemos que el movimiento propio se mueve en un rango bastante amplio, mientras que la distancia permanece prácticamente invariable, por otro lado si vemos el gráfico de **distancia vs radio**, podemos encontrar el mismo comportamiento inusual en la distancia, lo que nos lleva a pensar que pueda existir un error en nuestros datos de distancia para este grupo. Finalmente si miramos los valores promedio para la extinción y gravedad podemos notar que para la primera de estas se tiene un valor promedio muy bajo y al ver la gráfica de **luminosidad vs extinción** tenemos que una gran cantidad de valores se agrupa en la parte baja del gráfico pero se tiene una gran dispersión. Con respecto a la gravedad notamos que si bien se tiene un valor bajo este supera la gravedad de estrellas más masivas como los son las que se encuentran en el cluster 1 y 4 lo cual resulta interesante.

5.5.4. Cluster 4

Este grupo es bastante pequeño teniendo un total de solo 28 elementos. La posición dentro del diagrama H-R es en la parte central-baja de la secuencia principal. Con respecto al comportamiento en sus características tenemos que la luminosidad es bastante baja siendo en su valor promedio menor que nuestro sol, mientras que el radio,masa, y temperatura tienen valores promedio bajos cercanos a los de nuestro sol.

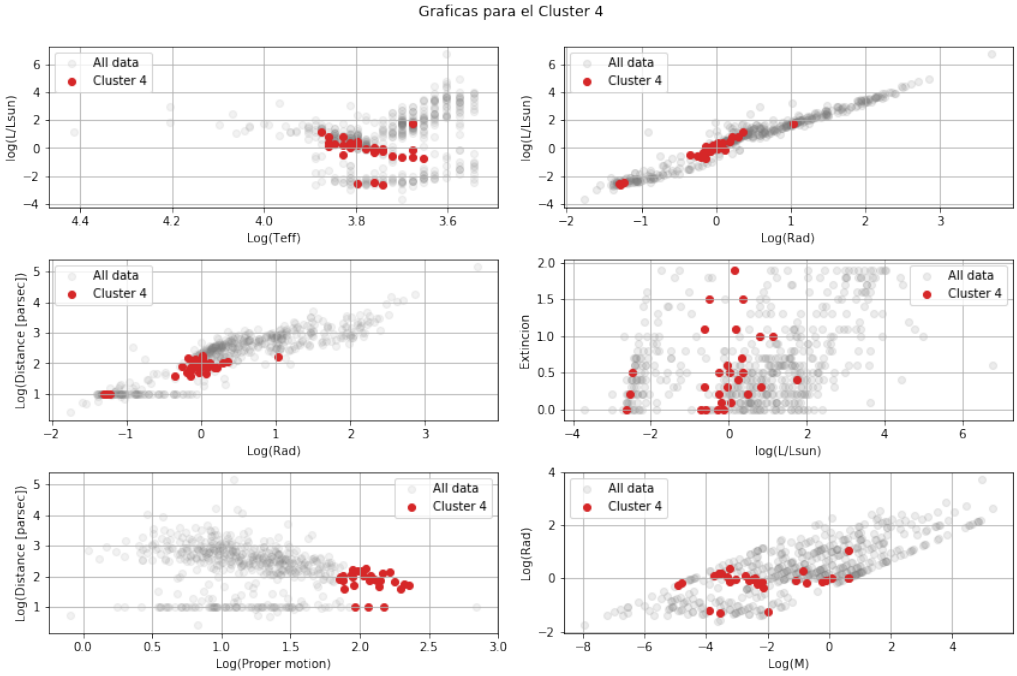

Figura 25: Gráficos para cluster 4
Fuente: Elaboración propia

Finalmente tenemos que el valor de la extinción es bajo y la característica más distintiva de este grupo es su movimiento propio muy alto, alejándose bastante de los valores promedio de los demás grupos

5.5.5. Cluster 5

Este grupo es el más pequeño de los encontrados y está compuesto solo por 11 elementos. Como podemos ver en la figura 20 este grupo en comparación con el resto tiene una menor dispersión en sus características. Dentro del comportamiento de sus propiedades, lo más destacables es que aquí se agrupan las estrellas que tienen los valores promedios más altos para el radio, masa, gravedad y distancia. Por otra parte tenemos que su temperatura efectiva promedio es la más baja de todos los grupos encontrados.

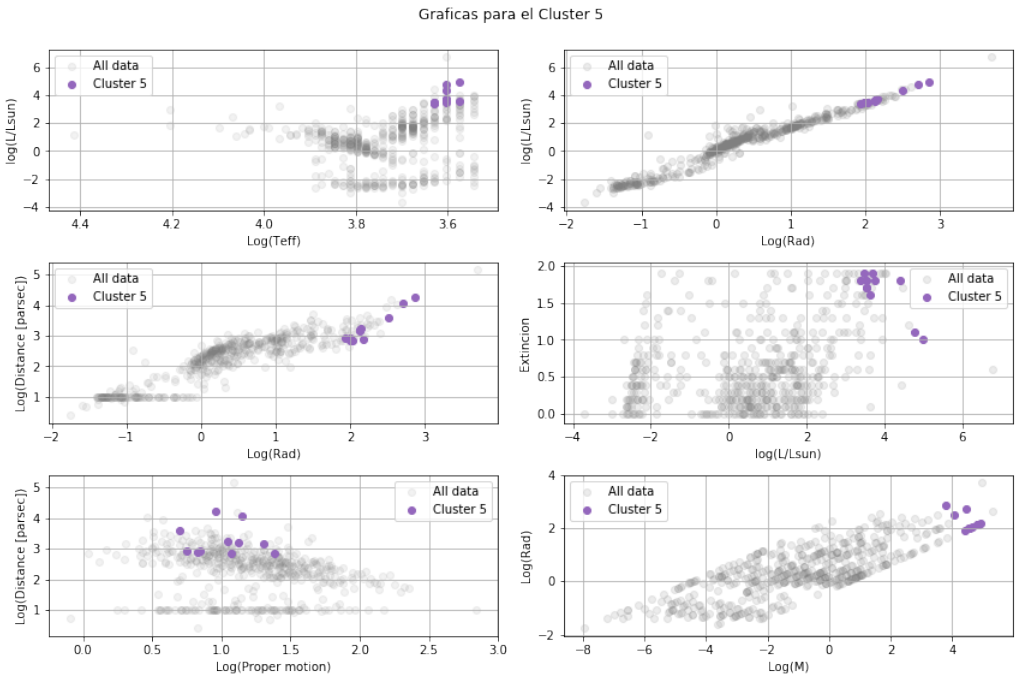

Figura 26: Gráficos para el cluster 5
Fuente: Elaboración propia

Lo anterior se puede corroborar con los gráficos comparativos de la figura 26, además el comportamiento en estas variables nos hace pensar que efectivamente nos encontramos ante un grupo de estrellas gigantes/supergigantes, ya que se logra ver patrones característicos de este tipo de estrellas como una alta luminosidad en conjunto con una temperatura efectiva baja y un radio alto. Otras de sus características sobresalientes son la extinción y el movimiento propio, donde la primera de ellas presenta el valor promedio más alto, el que se puede relaciona al igual que en los dos primeros grupos con la alta luminosidad que presentan estas estrellas, como se puede ver en la gráfica de **extinción vs luminosidad**, mientras que la segunda presenta el valor promedio más bajo, el que podemos relacionar con la distancia

5.5.6. Cluster 6

Similar al cluster 5 este grupo es bastante pequeño teniendo solamente 13 elementos. su ubicación dentro del diagrama H-R está en lo que denominamos secuencia principal, particularmente dentro de nuestros datos este grupo es el que se encuentra en la parte más alta de este segmento del diagrama.

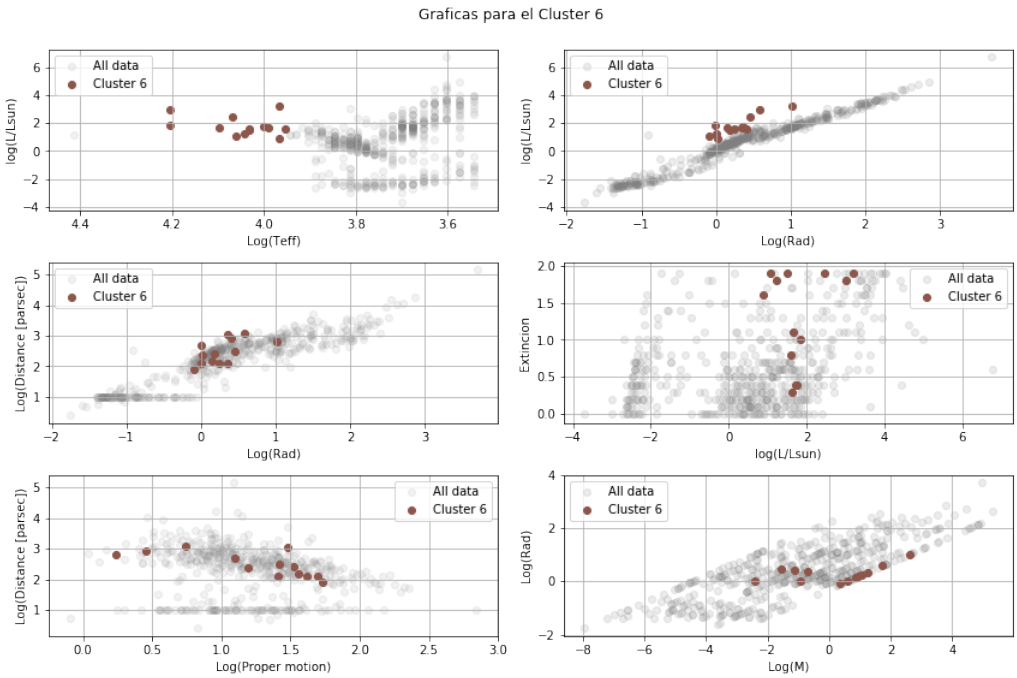

Figura 27: Gráficos para cluster 6
Fuente: Elaboración propia

Dentro de sus características podemos destacar que la temperatura efectiva promedio es la más alta, además encontramos valores promedio relativamente altos para para el radio, masa, distancia y luminosidad, siendo superada solo por el cluster 1 y cluster 5. Si vemos los gráficos de **luminosidad vs temperatura** y **luminosidad vs radio**, podemos ver que al ser estrellas con temperaturas altas y radios que a lo más superan 10 veces el radio solar, la luminosidad tiende a ser bastante alta como se mencionó anteriormente, por lo que podemos concluir que la temperatura hace que las estrellas irradien mucha más energía haciéndolas más luminosas. Por otro lado el valor promedio de la extinción está solo por detrás del valor encontrado por el cluster 5. Lo mencionado anteriormente puede que no sea tan claro como en el grupo anterior ya que en el gráfico de **luminosidad vs extinción** podemos ver que el grupo en general es bastante, disperso, donde los valores del Finalmente tenemos que su movimiento propio no destaca y se encuentra dentro de los valores promedio centrales.

En términos generales se pudieron observar algunos patrones en los diferentes grupos, donde algunos de estos son esperables, como por ejemplo la relación entre la luminosidad, temperatura efectiva y radio la cual es explicada por el diagrama H-R. El comportamiento conjunto de estas características tiende a explicar la luminosidad de las estrella, donde las estrellas más luminosas pueden tener una temperatura efectiva alta y un radio hasta 10 veces mayor que el sol o bien tener una temperatura efectiva baja y un radio muy grande llegando incluso hasta ser 1000 veces mayor que el de nuestro sol. Por otra parte a medida que decae la temperatura y el radio también lo hará la luminosidad. Otras tendencias que se puede observar es la luminosidad y la extinción, donde en general cuando el logaritmo de la luminosidad tiene valores mayor o igual a 2, la extinción tiende a ser alta. Para valores menores a estos

en general el comportamiento se torna bastante disperso. Por otro lado también logramos identificar una relación inversamente proporcional entre la distancia y el movimiento propio. Finalmente las magnitudes de la log gravedad y metalicidad presentan una gran variabilidad en todos los grupos por lo que encontrar relaciones entre estas y las demás características es algo complicado. Sin embargo podemos notar que las mayores magnitudes promedio en la log gravedad y Av se corresponden con las mayores magnitudes de la metalicidad.

CAPÍTULO 6
VALIDACIÓN DE LA SOLUCIÓN

En el presente capítulo se hará una revisión de los resultados obtenidos durante el proceso de clustering realizado en el capítulo anterior. Para una revisión de los resultados generales se solicitó la ayuda a las astrónomas Daniela Grandón y Amelia Bayo, las que con su conocimiento y experiencia ayudarán validar los resultados obtenidos mediante la aplicación del pipeline completo. También darán respuesta a las principales interrogantes que dejo este proceso, las cuales se plantearán a lo largo del capítulo. Cabe destacar que lo que se expondrá a continuación es una síntesis de la revisión que ellas hicieron a este trabajo.

6.1. Opinión experta sobre resultados obtenidos

Dentro de las principales observaciones que realizaron las expertas, tenemos varios puntos a considerar. El primero de ellos hace referencia a la luminosidad de la estrella (Log(L/L_{sun})), donde es esperable que los objetos que tengan valores bajo 0.1, en general sigan la secuencia principal como lo visto en la figura 2.3. Si consideramos las estrellas presentadas en la figura 11, se puede observar claramente que la mayoría de estas obedecen el comportamiento esperado (habiendo una baja densidad de objetos).

Como segundo punto, tenemos que existe un grupo de estrellas que se encuentran bajo la secuencia principal, y que presentan ciertas incongruencias en su comportamiento esperado. Por lo general en la zona inferior izquierda (bajo la secuencia principal) se encuentra el grupo de estrellas denominadas enanas blancas, las cuales se caracterizan por ser estrellas con temperaturas bastante altas y de poco tamaño, lo que hace que su luminosidad sea baja a pesar de su alta temperatura. Por otro lado, en la figura 11 podemos ver un grupo de estrellas que tienen valores de temperatura y luminosidad que no se corresponden con estrellas de la secuencia principal ni enanas blancas. Esto hace sospechar de la validez en los valores que poseen la mayoría estos objetos.

Otro punto importante a destacar, son los grupos que el proceso logro encontrar. Aquí se puede ver claramente como el cluster 1 es consistente con el grupo de estrellas denominadas gigantes rojas, mientras los cluster 2, 4 y 6, se conforman por estrellas pertenecientes a la secuencia principal. Esto último está bastante bien, ya que a lo largo de la secuencia principal existen diversas poblaciones estelares por lo que considerar a esta como un gran grupo, asignado un mismo valor promedio para las distintas características (metalicidad, log(g), radio, etc.) no sería correcto.

Por otro lado, tenemos que grupo de estrellas visto en la figura 11 que posee un comportamiento anómalo, fue agrupado en el cluster 3. Si analizamos las características de este grupo haciendo uso del boxplot de distancias mostrado en la figura 20, y el gráfico de Log(D) vs Log(Rad) mostrado en la figura 24, podemos concluir que ese cluster efectivamente no

está correcto, ya que muchas de las estrellas quedan fijas a una distancia de 10 parsecs. Esto se debe a que no se conocen sus distancias reales y al utilizar los servicios provistos por VOSA, se procede a utilizar el valor por defecto, el cual efectivamente corresponde a 10 parsecs. Una recomendación importante para evitar en el futuro este tipo de inconvenientes, es utilizar los datos más actualizados entregados por la misión GAIA, ya que este catálogo posee distancias calculadas de forma bastante precisa para un gran número de estrellas, lo que permite además tener mejores estimaciones de la luminosidad, radio e incluso la masa de la estrella. Respecto a este mismo cluster que se postula en la sección 5.5.3 como enanas rojas, se puede apreciar que tienen una temperatura efectiva promedio de 5300 K. Las enanas rojas poseen en promedio temperaturas mucho más bajas. Lamentablemente el promedio se ve afectado por las estrellas a la izquierda (más calientes) que tienen un error en los valores de algunas de sus características.

Por lo tanto, en base a todo lo anteriormente expuesto, el cluster 3 debe ser descartado, y no se deben obtener conclusiones a partir de él.

6.2. Preguntas hacia las expertas

En base al comportamiento de las características presentadas en cada uno de los grupos encontrados, surgen algunas interrogantes sobre qué tan importantes, normales o esperables son los comportamientos que estos describen. A continuación, se plantean las principales dudas que generó el proceso de clustering y las respuestas entregadas por las expertas.

1. **¿Es correcto que el movimiento propio sea más alto en estrellas más cercanas?**

   Examinando los boxplots de distancia y movimiento propio mostrados en la figura 20, se puede observar que efectivamente existe una relación inversa entre estas características, por lo que plantea esta pregunta en términos generales es correcto. Sin embargo, puede darse el caso que esto no sea siempre así, ya que la distancia no es el único factor involucrado en el movimiento propio de una estrella.

2. **¿ Es normal que en el cluster 4 su movimiento propio sea tan alto siendo que no son las estrellas más cercanas?**

   Esto es relativo, debido a que pueden existir objetos con un movimiento propio muy pequeño moviéndose radialmente, esto quiere decir que se encuentran alejándose o acercándose a nosotros, incluso se podría tener dos estrellas con el mismo movimiento propio, pero a distancias diferentes.

3. **¿Es normal que en estrellas con masas altas como las vistas en el cluster 1, la magnitud en su gravedad sea más baja que en estrellas menos masivas?**

   Tenemos que Log(g) depende de la masa, pero es aún más dependiente del radio ($g = \frac{Gm}{r}^2$). Las estrellas gigantes al tener un radio mucho mayor, provocan una disminución mucho más pronunciada en la gravedad, si la compráramos por ejemplo con una pérdida de masa.

4. **¿Es normal que el valor promedio de la extinción sea alto en las estrellas más luminosas?**

   La extinción es un fenómeno provocado por el polvo existente entre el planeta tierra y la estrella a lo lago de la línea de visión. Este polvo absorbe fotones con longitudes de onda más cortas, lo que implica una mayor cantidad de energía, por ende, afecta más a estrellas con una luminosidad alta. El polvo se calienta y re-emite estos fotones en longitudes de ondas más largas, afectando la clasificación espectral de la estrella, luminosidad y su posición en el diagrama HR.

5. **¿Es correcto que exista una relación entre valores altos de gravedad, extinción y metalicidad?**

   La gravedad (log(g)) se extrae de la distribución espectral de energía de la estrella, por lo que puede verse afectado por extinción (Av). Por otro lado, la metalicidad se obtiene del espectro de la estrella (observando sus líneas de emisión y absorción), sin embargo, la extinción solo afecta el continuo (distribución espectral de energía, no las líneas específicamente). La Metalicidad habla más de la población estelar a la que la estrella pertenece, mientras que la extinción afecta a la estrella a través de un proceso extrínseco.

6.3. Conclusiones sobre las observaciones

Con lo expuesto en la sección anterior, podemos decir que independiente de que para la mayoría de los grupos encontrados, el comportamiento cae dentro de lo que se espera para la etapa evolutiva en la que se encuentran (exceptuando el cluster 3), el proceso planteado en el capítulo 4 fue capaz de encontrar estructuras complejas sólo a partir de los datos, sin el uso de etiquetas previas, incluso logrando agrupar en un mismo cluster las estrellas que tenían problemas con su distancia.

Con respecto a los datos, tenemos que el cruce fue realizado utilizando VOSA y el catálogo HSOY, el cual está basado en el Gaia data release 1, lo que significa que el problema con las distancias encontradas en el cluster 3 es debido a que no se pudo hacer el match en ninguna de nuestras fuentes de datos, por lo que para trabajos futuros es necesario contar con más alternativas como las propuestas en la sección 4.2.1. Por otro lado también se hace necesario contar con más características que nos entreguen más información acerca del movimiento de la estrella, como lo puede ser la velocidad radial y otras propiedades que nos ayuden a develar el motivo por que ocurre este fenómeno.

Finalmente podemos mencionar que utilizar el diagrama H-R como base para formular la interpretación de cada grupo facilitó bastante la apreciación de las expertas, por lo que fue un acierto utilizar esta referencia. Por otro lado podemos decir que queda demostrado que el uso de algoritmos de clustering son bastante útiles a la hora de realizar clasificaciones sin conocer la etiqueta o la estructura de los datos previamente.

CAPÍTULO 7
CONCLUSIONES

Como se pudo ver durante el desarrollo de esta memoria el problema de las estrellas variables de largo periodo es un tema bastante complejo, donde para facilitar su desarrollo lo dividimos en una serie de etapas las cuales agrupamos en dos partes. La primera parte se enfocó en realizar la clasificación sobre qué estrellas presentan el comportamiento variable en el largo periodo, mientras que la segunda parte se enfocó en realizar la búsqueda de grupos en base a sus propiedades físicas utilizando el resultado del a primera parte, es decir la búsqueda de grupos se hizo en base a las estrellas ya clasificadas.

Dentro de las etapas pertenecientes a la primera parte, podemos identificar dos etapas críticas que afectan de forma importante el desarrollo y los resultados de esta. Dentro de estas etapas podemos mencionar, el proceso de obtención de datos, ya que las curvas de luz que fueron usadas para la realización de este trabajo necesariamente deben estar compuestas por mediciones realizadas durante varios años, lo que acota bastante nuestras opciones para escoger una fuente de datos confiable. Otro punto importante a destacar sobre esta etapa es que si bien existen al menos dos sky surveys que nos pueden proporcionar los datos, estos no tienen una estructura que soporte la extracción masiva y automatizada de estos, como lo es en el caso de ASAS. Otra de las etapas críticas es el proceso de clasificación, donde al ser un tema poco estudiado no se cuentan con los ejemplos suficientes para poder aplicar técnicas supervisadas o semi supervisadas que automaticen esta etapa, por lo que es necesario hacer uso de técnicas estadísticas que nos ayuden a formular algún criterio para realizar la clasificación entre las estrellas que efectivamente tengan un comportamiento variable en el largo periodo. La elección de estos criterios puede afectar los resultados de la clasificación y por ende el desarrollo de las etapas pertenecientes a la segunda parte del problema. Finalmente podemos recalcar que dada la complejidad de esta parte el desarrollo de esta puede dar pie para la realización de otro trabajo de investigación.

Con respecto a la segunda parte, podemos mencionar que la etapa mas critica es la realización del cross-matching, ya que el desarrollo de las etapas siguientes dependen de esta. Si bien ya existen variadas soluciones para este problema que consiste en hacer el cruce de datos astronómicos entre múltiples bases de datos y obtener distintas características para un mismo objeto, para poder realizarlo de forma correcta son necesarios ciertos conocimientos astronómicos previos, de lo contrario se pueden obtener datos incorrectos lo que puede hacer que se produzca un análisis erróneo sobre el comportamiento de los datos.

Finalmente dentro del proceso de clustering si bien los puntajes del índice de silhouette que obtuvimos al aplicar k-means no fueron “buenos”, los grupos que se lograron encontrar siguen la estructura del diagrama H-R, lo cual es bastante bueno ya que el uso de esta estructura facilita enormemente la interpretación de los datos para los astrónomos. Con respecto a los resultados, dentro de los grupos no se logró ver un comportamiento inusual donde la mayoría sigue un comportamiento acorde a las distintas etapas de la evolución estelar que

representa cada grupo encontrado. La excepción a esto se vio en el cluster 3, el cual tenía problemas con los valores de su distancia los cuales era incorrectos. Si bien este grupo estaba mal construido debido a un error de los datos k-means logró detectar esta anomalía, y agrupar todos estos elementos dentro de un mismo cluster.

Con respecto a los objetivos que se plantearon en un inicio, podemos decir que se lograron cumplir a cabalidad ya que el proceso propuesto logró automatizar el proceso de detección y clasificación de estrellas variables en el largo periodo. Si bien como se mencionó anteriormente el proceso de detección el cual corresponde a la primera parte mencionada anteriormente se realiza mediante una clasificación informada, aun así logra cumplir su objetivo ya que al estar programado el algoritmo que obtiene las características y aplica los criterios, este proceso se lleva a cabo de forma automática pudiendo ser aplicado a una gran cantidad de datos si así se deseara. Con respecto a la parte de clasificación, esta corresponde a la segunda parte y se logró mediante al realizar una clasificación no supervisada mediante algoritmos de clustering en base a las propiedades físicas de los datos.

7.1. Consideraciones para el trabajo futuro

Como se mencionó en el capítulo de validación los resultados encontrados obedecen a un comportamiento normal acorde a las distintas etapas evolutivas que describe cada grupo. Dentro de los posibles factores que pueden haber afectado la obtención de resultados tenemos en primer lugar la cantidad de datos que se usaron. Si bien se contaba con la información para un poco más de 11000 estrellas al aplicar el proceso de detección este número se reduce a solo 529 estrellas, que representan alrededor de un $4{,}5\,\%$ de la muestra total. Por otro lado ASAS cuenta con una base de datos de 17 millones de estrellas con mediciones en un periodo de 9 años, pero debido a la dificultad para poder conseguir los datos, no se logró obtener una muestra mayor. Entonces como primer paso para poder mejorar este trabajo es necesario contar con una cantidad de datos que permita que el proceso de clustering permite encontrar grupos representativos. Otro punto que debe ser revisado son la etapa de extracción de características y posterior clasificación ya que el método que se usó para el desarrollo de esta memoria es bastante simple. Lo ideal sería encontrar un método estadístico más robusto que entregue más información sobre la curva de luz de cada estrella. Algunas alternativas podrían ser métodos basados en series temporales, o métodos populares usados en astronomía.Sin embargo hay que tener cuidado con la elección de un nuevo método ya que varios de estos se basan en la búsqueda de periodos, y particularmente para este problema tenemos que muchas curvas de luz pueden ni siquiera completar un solo periodo. Por ultimo, también se hace necesario considerar las estrellas que ya han sido clasificadas como variables en el corto periodo y ver si estas además presentan algún tipo de variabilidad en el largo periodo.

7.2. Conclusiones generales

Finalmente y a modo de conclusión general tenemos que el estudio de las estrellas variables resulta importante para disciplinas como la astronomía y astrofísica debido a la gran cantidad de información relevante que se puede obtener de estas, información que ayuda a la comprensión del comportamiento de estos cuerpos celestes y los sistemas estelares que las rodean. Por otro lado el estudio de este fenómeno en el largo periodo necesita del apoyo de alguna metodología como la presentada en este trabajo que ayude a analizar la gran cantidad de información que hay disponible para el estudio, ya que en su mayoría, las estrellas que pueden presentar este comportamiento están clasificadas como constantes. Por otro lado el contar con algoritmos y técnicas de aprendizaje automático no solo se aplica para la resolución de este problema, si no que marcan la pauta para el desarrollo en la astronomía, esto principalmente debido a que con la construcción de observatorios como el LssT y los diferentes Sky Surveys se generarán petabytes de datos en periodos relativamente cortos de tiempo, por lo que será una necesidad tener contempladas técnicas de machine learning y data mining para analizar semejante cantidad de información.

ANEXOS

7.1. Algoritmos de clasificación supervisada

7.1.1. Arboles de decisión

EL algoritmo presentado a continuación corresponde al algoritmo **ID3** para crear arboles de decisión. Este algoritmo recibe como entrada un conjunto de ejemplos descritos mediante una serie de pares atributo-valor y la salida será un árbol de decisión que separa a los ejemplos de acuerdo a las clases a las que pertenecen.

Algorithm 5 Árbol de decisión [43]

Require: T, F, C
1: **if** $T_{i,C}$ es el mismo para $i \in 1, 2, 3, .., |T|$ **then**
2: **return** nodo hoja etiquetado como $T_{1,C}$
3: **else**
4: **if** $|F| = 0$ **then**
5: **return** nodo hoja etiquetado como $argmax_c \sum_{i=1}^{|T|} I(T_{i,C} = c)$
6: **else**
7: f = ChooseBestFeature(T, F)
8: establece que el árbol sera un nodo no terminal con el test f
9: **for** $v \in f$ **do**
10: $T_{f=v} = \phi$
11: **end for**
12: **for** $T_i \in T$ **do**
13: $v = T_{i,f}$
14: Add examplevT_i to set $T_{f=v}$
15: **end for**
16: **for** $v \in f$ **do**
17: subtree = ArbolDeDecision($T_{f=v}, F - f, C$) ←Paso recursivo
18: Añade una rama al arbol etiquetado f con los sub arboles
19: **end for**
20: **end if**
21: **end if**
22: **return** tree

En el algoritmo la función ChooseBestFeature(T, F) esta definida en base a la ganancia máxima de información que entrega una característica para seleccionar el atributo más útil en cada paso. Esto ultimo esta definido por.

$$Gain(T, f) = Entropy(T) - \sum_{v \in values(f)} \frac{|T_{v=f}|}{|T|} Entropy(T_{v=f}) \tag{13}$$

y donde la entropía esta dada por

$$Entropy(T) = \sum_{i \in values(C)} - p_i \log_2(p_i) \tag{14}$$

Aquí tenemos que $values(f)$ es el conjunto de posibles valores de características f, si f es una característica categórica. Si f es una característica ordenada se consideran diferentes puntos binarios de división. T es el conjunto de entrenamiento, $T_{v=f}$ es un subconjunto del conjunto de entrenamiento que tiene el valor v para una característica f, C es la clase de característica y p_i es la fracción del conjunto de entrenamiento tiene la clase C_i. La ganancia de información es la entropía menos la suma de entropías de todos los subconjuntos de entrenamiento. que resultan de separar el conjunto de entrenamiento según la característica f. Mientras mas homogenio sea el conjunto de entrenamiento en términos de los diferentes valores de las clases menor será la entropía. Por lo tanto, la ganancia de información mide cuanto una característica mejora la separación del conjunto de entrenamiento en subconjuntos de clases homogéneas.

7.1.2. Support vector machines

Las SVM comienzan realizando una etapa de aprendizaje donde se busca un hiper-plano que mejor separe el conjunto de datos **X** $\in$ **R**^{d} según a la clase **Y** $\in \{-1, 1\}$ que pertenezcan. A esta distancia se le denominara margen, y a los puntos presentes en dicho margen serán los vectores de soporte.

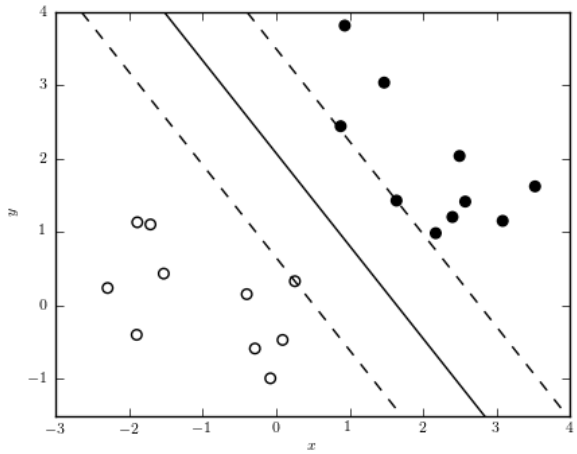

Figura 28: Diagrama que muestra la aplicación del algoritmo SVM.
El espacio entre las lineas punteadas es el margen, mientras que los puntos que se encuentran sobre estas son los vectores de soporte. [19]

El separador lineal o hiper-plano que maximiza el margen es el que entrega la mejor capacidad de distinguir características comunes de los datos de cada clase, permitiendo así la

clasificación de datos que no pertenezcan al conjunto de entrenamiento. Para obtener el hiper-plano es necesario considerar u vector normal w y una distancia b hacia el origen, todo esto considerando la linea central del hiper-plano.

Para el caso en que el conjunto de datos sea perfectamente separable, aprender un vector de soporte implica usar los datos de entrenamiento para encontrar el mejor valor de w, el cual se obtiene resolviendo el siguiente problema de optimización:

$$\begin{aligned} \text{min} \quad & ||w|| \\ & \text{sujeto a} \\ x_i \cdot w + b \geq 1 \quad & \text{para} \;\; y_i = +1 \\ x_i \cdot w + b \leq -1 \quad & \text{para} \;\; y_i = -1 \end{aligned}$$

Cuando la data no es perfectamente separable, este método puede ser adaptado para compensar los datos que quedan mal clasificados con la introducción de variables de holgura ξ que mide el error producido por las instancias mal clasificadas de los datos de entrenamiento. Estas variables de holgura miden la distancia a la que se encuentra la variable mal clasificada respecto del hiper-plano. Entonces SVM encuentra el mejor hiper-plano que separa a los grupos y que minimiza la suma de los errores producidos por las variables de holgura ξ_i. El modelo de optimización a resolver se transforma en:

$$\begin{aligned} \text{min} \quad & ||w|| + B \sum_{i=1}^{|T|} \xi_i \\ & \text{sujeto a} \\ x_i \cdot w + b \geq 1 - \xi_i \quad & \text{para} \;\; y_i = +1 \\ x_i \cdot w + b \leq -1 + \xi_i \quad & \text{para} \;\; y_i = -1 \end{aligned}$$

donde B es un peso que se le da a las variables de holgura ξ_i. Si B es grande, las penalizaciones a los errores será mayor, por lo que se considerara un clasificador mas complejo, por otro lado si B es pequeño, el clasificador será más simple pero es posible que el error sea más grande en el conjunto de entrenamiento.

7.2. Código y jupyter notebooks

Todo el código utilizado para el desarrollo de este prototipo así como los jupyter notebooks utilizados para las diferentes pruebas realizadas, se encuentran disponibles en el siguiente repositorio de github:

https://github.com/PaHorth/var-stars

REFERENCIAS BIBLIOGRÁFICAS

[1] Timing analysis. `https://imagine.gsfc.nasa.gov/science/toolbox/timing2.htm`. Accessed: 2018-11-5.

[2] Aggarwal, C. C., and Reddy, C. K. Data clustering: algorithms and applications. CRC press, 2013.

[3] Altmann, M., Roeser, S., Demleitner, M., Bastian, U., and Schilbach, E. Hot stuff for one year (hsoy)-a 583 million star proper motion catalogue derived from gaia dr1 and ppmxl. Astronomy & Astrophysics 600 (2017), L4.

[4] Bayo, A., Rodrigo, C., y Navascués, D. B., Solano, E., Gutiérrez, R., Morales-Calderón, M., and Allard, F. Vosa: virtual observatory sed analyzer-an application to the collinder 69 open cluster. Astronomy & Astrophysics 492, 1 (2008), 277–287.

[5] Bellman, R. E. Adaptive control processes: a guided tour, vol. 2045. Princeton university press, 2015.

[6] Bishop, C. M. Pattern recognition and machine learning. springer, 2006.

[7] Borne, K. D. Astroinformatics: a 21st century approach to astronomy. arXiv preprint arXiv:0909.3892 (2009).

[8] BSJ. variables: Types of variables. url https://www.aavso.org/types-variables. Accedido 10-06-2017.

[9] Davies, D. L., and Bouldin, D. W. A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence, 2 (1979), 224–227.

[10] eduardoc's. la maldición de la dimensionalidad en machine learning. urlhttp://ingenierobeta.com/maldicion-de-la-dimensionalidad. Accedido 15-12-2018.

[11] ESO. Hertzsprung-russell diagram. url https://www.eso.org/public/images/eso0728c/. Accedido 25-10-2018.

[12] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (1996), vol. 96, pp. 226–231.

[13] Facility, A. T. N. Variable stars. url http://www.atnf.csiro.au/outreach/education/senior/astrophysics/varia types.html. Accedido 10-06-2017.

[14] González Casimiro, M. P. Análisis de series temporales: Modelos arima.

[15] Gorski, K. M., Wandelt, B. D., Hansen, F. K., Hivon, E., and Banday, A. J. The healpix primer. arXiv preprint astro-ph/9905275 (1999).

[16] Gray, J., Nieto-Santisteban, M. A., and Szalay, A. S. The zones algorithm for finding points-near-a-point or cross-matching spatial datasets. arXiv preprint cs/0701171 (2007).

[17] Hamilton, J. D. Time series analysis, vol. 2. Princeton university press Princeton, NJ, 1994.

[18] Han, J., Pei, J., and Kamber, M. Data mining: concepts and techniques. Elsevier, 2011.

[19] Ivezić, Ž., Connolly, A. J., VanderPlas, J. T., and Gray, A. Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data. Princeton University Press, 2014.

[20] Koposov, S., and Bartunov, O. Q3c, quad tree cube–the new sky-indexing concept for huge astronomical catalogues and its realization for main astronomical queries (cone search and xmatch) in open source database postgresql. In Astronomical Data Analysis Software and Systems XV (2006), vol. 351, p. 735.

[21] Kozak, M. "a dendrite method for cluster analysis" by caliński and harabasz: A classical work that is far too often incorrectly cited. Communications in Statistics-Theory and Methods 41, 12 (2012), 2279–2280.

[22] Kunszt, P. Z., Szalay, A. S., and Thakar, A. R. The hierarchical triangular mesh. In Mining the sky. Springer, 2001, pp. 631–637.

[23] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. Journal of machine learning research 9, Nov (2008), 2579–2605.

[24] Mitchell, T. M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill 45, 37 (1997), 870–877.

[25] M.Saladyga. variables: what are they and why observe them? url https://www.aavso.org/variables-what-are-they-why-observe-them. Accedido 10-06-2018.

[26] Nieto-Santisteban, M. A., Thakar, A. R., and Szalay, A. S. Cross-matching very large datasets. In National Science and Technology Council (NSTC) NASA Conference (2007).

[27] Ochsenbein, F., Bauer, P., and Marcout, J. The vizier database of astronomical catalogues. Astronomy and Astrophysics Supplement Series 143, 1 (2000), 23–32.

[28] Oswalt, T. D., and Gilmore, G. Planets, stars and stellar systems. Springer, 2013.

[29] Ozkok, F. O., and Celik, M. A new approach to determine eps parameter of dbscan algorithm. International Journal of Intelligent Systems and Applications in Engineering 5, 4 (2017), 247–251.

[30] Palma, D. C. The hertzsprung-russell diagram. urlhttps://www.e-education.psu.edu/astro801/content/l4-p6.html. Accedido 31-09-2018.

[31] Park, H., and Ozeki, T. Singularity and slow convergence of the em algorithm for gaussian mixtures. Neural processing letters 29, 1 (2009), 45–59.

[32] Peña, D. Análisis de datos multivariantes. McGraw-Hill España, 2013.

[33] Pichara, K., Protopapas, P., and León, D. Meta-classification for variable stars. The Astrophysical Journal 819, 1 (2016), 18.

[34] Pojmanski, G. The all sky automated survey. arXiv preprint astro-ph/9712146 (1997).

[35] Pojmanski, G., and Maciejewski, G. The all sky automated survey. catalog of variable stars. iv. 18^ h-24^ h quarter of the southern hemisphere. Acta Astronomica 55 (2005), 97–122.

[36] Prusti, T., De Bruijne, J., Brown, A. G., Vallenari, A., Babusiaux, C., Bailer-Jones, C., Bastian, U., Biermann, M., Evans, D. W., Eyer, L., et al. The gaia mission. Astronomy & Astrophysics 595 (2016), A1.

[37] Richards, J. W., Starr, D. L., Miller, A. A., Bloom, J. S., Butler, N. R., Brink, H., and Crellin-Quick, A. Construction of a calibrated probabilistic classification catalog: application to 50k variable sources in the all-sky automated survey. The Astrophysical Journal Supplement Series 203, 2 (2012), 32.

[38] Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20 (1987), 53–65.

[39] s, J. G. Estudio estadístico de estrellas variables a largo plazo con baja amplitud a partir de base de datos asas.

[40] Triguero, I., García, S., and Herrera, F. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowledge and Information systems 42, 2 (2015), 245–284.

[41] Vapnik, V. The nature of statistical learning theory. Springer science & business media, 2013.

[42] Vogt, N., Kroll, P., and Splittgerber, E. A photometric pilot study on sonneberg archival patrol plates-how many “constant” stars are in fact long-term variables? Astronomy & Astrophysics 428, 3 (2004), 925–934.

[43] Way, M. J., Scargle, J. D., Ali, K. M., and Srivastava, A. N. Advances in machine learning and data mining for astronomy. CRC Press, 2012.

[44] Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., Genova, F., Jasniewicz, G., Laloë, S., Lesteven, S., et al. The simbad astronomical database-the cds reference database for astronomical objects. Astronomy and Astrophysics Supplement Series 143, 1 (2000), 9–22.

[45] Wirth, R., and Hipp, J. Crisp-dm: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (2000), Citeseer, pp. 29–39.