Agrupamiento de exoplanetas similares a la Tierra usando *sklearn* de Python

C. Iván Pineda S.[1], E. Daniel Ortiz C.[1]

[1] *Universidad Nacional Autónoma de México*
08 de enero de 2018

Resumen

Se utilizó una base de datos [1] en la que se agregó la información de la Tierra, Marte y Júpiter con el fin de clasificar exoplanetas de acuerdo a su parentesco con la Tierra. Después de normalizar y limpiar los datos, con cinco variables (Masa, radio, semieje mayor del planeta, masa y radio de la estrella) se hizo un *PCA* [2] y se utilizaron dos componentes principales con los cuales se hizo un *clustering k-means*[3] para agrupar los planetas y obtener los más parecidos a la Tierra. Posteriormente se realizó el mismo procedimiento pero tomando en cuenta la temperatura esperada en el exoplaneta, la masa y el radio. Finalmente se hizo una comparación con otra base de datos [4] con una clasificación entre exoplanetas habitables y no habitables obteniendo cinco planetas parecidos a la Tierra.

Introducción

En el año de 1992 se descubrieron los primeros 2 exoplanetas[5] , estos cuerpos fueron detectados debido a que el púlsar que orbitan tiene anomalías en sus pulsaciones [*Wolszczan et al.*, 1994]. A partir de este momento comenzó un fase de constante descubrimiento de exoplanetas a tal grado de que para el año 2017 ya se habían confirmado 3726[1].

Con el uso de telescopios espaciales como el Kepler y con el nuevo lanzamiento en la próxima década del telescopio espacial James Webb, se espera que el número de detecciones nuevas de

[1] http://exoplanet.eu/catalog/all_fields/ revisado el miércoles 3 de enero del 2018

[2] Análisis de componentes principales (Principal component analysis).

[3] Agrupamiento de k-medias.

[4] phl_hec_all_confirmed.csv obtenida de http://phl.upr.edu/projects/habitable-exoplanets-catalog/data/database el domingo 7 de enero del 2018

[5] Planetas orbitando una estrella que no sea el Sol

exoplanetas aumente significativamente. Esto quiere decir que se deben emplear nuevos recursos para analizar una base de datos que crece día con día a ritmos acelerados.

Un problema central en el análisis de datos multivariantes es la reducción de la dimensionalidad, en estadística estos problemas son solucionados con técnicas que contrastan la correlación y la varianza entre variables de una base de datos, en la actualidad se utilizan estas técnicas en distintas áreas de la ciencia para describir el comportamiento de distintas variables en un estudio.

El análisis multivariante (AM) es la parte de la estadística y del análisis de datos que estudia, analiza, representa e interpreta los datos que resultan de observar más de una variable estadística sobre una muestra de individuos. Las variables observables son homogéneas y correlacionadas, sin que alguna predomine sobre las demás. La información estadística en AM es de carácter multidimensional, por lo tanto, la geometría, el cálculo matricial y las distribuciones multivariantes juegan un papel fundamental. La información multivariante es una matriz de datos, pero a menudo, en AM la información de entrada consiste en matrices de distancias o similitud, que miden el grado de discrepancia entre los individuos [*Cuadras*, 2007].

Análisis Clúster es el nombre genérico de una amplia variedad de procedimientos que pueden ser usados para crear una clasificación. Mas concretamente, un método clúster es un procedimiento estadístico multivariante que comienza con un conjunto de datos conteniendo información sobre una muestra de entidades e intenta reorganizarlas en grupos relativamente homogéneos a los que llamaremos clúster. En Análisis Clúster poca o ninguna información es conocida sobre la estructura de las categorías, lo cual lo diferencia de los métodos multivariantes de asignación y discriminación. De todo lo que se dispone es de una colección de observaciones, siendo el objetivo operacional en este caso, descubrir la estructura de las categorías en la que se encajan las observaciones.

El objetivo es ordenar las observaciones en grupos tales que el grado de asociación natural es alto entre los miembros del mismo grupo y bajo entre miembros de grupos diferentes. Aunque poco o nada se conoce sobre la estructura de las categorías a priori, se tiene con frecuencia algunas nociones sobre características deseables e inaceptables a la hora de establecer un determinado esquema de clasificación. En términos operacionales, el analista es informado suficientemente sobre el problema, de tal forma que puede distinguir entre buenas y malas estructuras de categorías cuando se encuentra con ellas.

La idea central del *PCA* es reducir la dimensionalidad de un conjunto de datos consistente en un número elevado de variables interrelacionadas. Se trata de mantener de la mejor manera posible

la variación contenida en los datos. Esto se logra transformando el conjunto original en un nuevo conjunto de datos, los componentes principales, que son no correlacionados y que están ordenados de tal manera que los primeros pocos retengan la mayor parte de la variación presente en todas las variables originales.

Metodología

Todo el desarrollo computacional se hizo con Python. Se utilizó la libreria de *pandas* para el procesamiento del archivo csv donde venían los datos, *sklearn* y *numpy* para el desarrollo estadístico y para graficar, *matplotlib.pyplot*.

Procesamiento de de los datos

Primero se normalizaron [1] los datos para evitar que algunas variables no fueran tomadas en cuenta a la hora de hacer el *clustering*.

```
#Normalizamos los datos

datosnorm=preprocessing.normalize(datoslimpios.get(['mass [Mj]','radius [Rj]','semi_major_axis [UA]'
                                                    ,'star_mass [Ms]','star_radius [Rs]']))
print(pd.DataFrame(datosnorm))
```

Figura 1. Normalización usando *preprocessing* de *sklearn*.

Después se asignó más peso a las variables de interés [2 & 3].

```
#Le daremos más importancia a los parámetros dentro de una zona habitable, esto es darle mayor peso a la distancia
#a la estrella, su masa y su radio.
datosnorm2=datosnorm
for i in range(len(datosnorm2)):
    datosnorm2[i][0]=datosnorm[i][0]*9 #Masa del planeta
    datosnorm2[i][1]=datosnorm[i][1]*9 #Radio del planeta
    datosnorm2[i][2]=datosnorm[i][2]*10 #Semieje mayor
    datosnorm2[i][3]=datosnorm[i][3]*12 #Masa de la estrella
    datosnorm2[i][4]=datosnorm[i][4]*12 #Radio de la estrella
print(pd.DataFrame(datosnorm2))
```

Figura 2. Multiplicación de las variables normalizadas, mayor número a las que se les da más peso. Este es el caso para el primer *clustering*, el que no toma en cuenta la temperatura calculada.

Finalmente se hizo el *PCA* para obtener dos componentes principales [4], estas componentes tienen la característica de que una es normal a la otra y ambas contienen la mayor variabilidad de las variables originales.

```
#Le damos más peso al radio, en la normalización de la temperatura la mayoría de los datos queda bastante grande compara-
#ndo con la masa y el radio, aun así es la variable a la que más peso se le deb dar y si nos fijamos en la lista
#de abajo, podemos ver que la varaible 2 correspondiente a la temperatura, es la más pesada.
datosnorm1_12=datosnorm1_1
for i in range(len(datosnorm1_12)):
    datosnorm1_12[i][0]=datosnorm1_1[i][0]*10 #Masa del planeta
    datosnorm1_12[i][1]=datosnorm1_1[i][1]*50 #Radio del planeta
    datosnorm1_12[i][2]=datosnorm1_1[i][2]*20 #Temperatura calculada
print(pd.DataFrame(datosnorm1_12))
```

Figura 3. Multiplicación de las variables normalizadas, mayor número a las que se les da más peso. Este es el caso para el segundo *clustering*, el que toma en cuenta la temperatura calculada.

```
# Hacemos el PCA para obtener 2 componentes principales.
X3 = PCA(n_components=2).fit_transform(datosnorm2)
print(X3)
```

Figura 4. *PCA* usando *sklearn*. Este es el caso para el primer *clustering*, el que no toma en cuenta la temperatura calculada.

Clustering

Se realizó el método del codo [5] para saber que cantidad de clústers realizar y no hacer más de los necesarios, este algoritmo se utiliza para ver a partir de que número de clústers la convergencia es tal que ya no se nota una diferencia significativa.

```
#“Within cluster sum of squares by cluster”
wcss3 = []
#i es el número de veces que queremos hacer el clustering.
#The elbow method se usa para ver si el problema converge y saber el número de clusterings necesarios.
for i in range(1, 11):
    kmeans3 = KMeans(n_clusters = i, init = 'k-means++', random_state = 42) #Características
    kmeans3.fit(X3) #Le pasamos los datos
    wcss3.append(kmeans3.inertia_)
fig=plt.figure(figsize=(15,4))
plt.plot(range(1, 11), wcss3)
plt.title('Método del codo, zona habitable')
plt.xlabel('Número de clusters')
plt.ylabel('WCSS')
plt.savefig('elbow.png',dpi=400)
plt.show()
```

Figura 5. Método del codo usando *sklearn*. Este es el caso para el primer *clustering*, el que no toma en cuenta la temperatura calculada. Para este caso se observó una convergencia a partir de los seis clústers.

Se hizo un *clustering* [6] con los dos componentes principales obtenidos del *PCA* para ambos casos.

Histogramas

Se realizaron histogramas [7] y se calcularon las medidas de tendencia central (media, varianza y desviación estándar) para observar la dispersión de los datos obtenidos en el clúster donde estaba

```
#Realizamos el Clustering para 6 clústers.
kmeans3 = KMeans(n_clusters = 6, init = 'k-means++', random_state = 42)
y_kmeans3 = kmeans3.fit_predict(X3)

# Graficamos los clústers
fig=plt.figure(figsize=(20,6))
plt.scatter(X3[0,0], X3[0,1], s = 200, c = 'black', label = 'Tierra')
plt.scatter(X3[1,0], X3[1,1], s = 200, c = 'gray', label = 'Júpiter')
plt.scatter(X3[2,0], X3[2,1], s = 200, c = 'pink', label = 'Marte')
plt.scatter(X3[y_kmeans3 == 0, 0], X3[y_kmeans3 == 0, 1], s = 50, c = 'red', label = 'Cluster 0')
plt.scatter(X3[y_kmeans3 == 1, 0], X3[y_kmeans3 == 1, 1], s = 50, c = 'blue', label = 'Cluster 1')
plt.scatter(X3[y_kmeans3 == 2, 0], X3[y_kmeans3 == 2, 1], s = 50, c = 'green', label = 'Cluster 2')
plt.scatter(X3[y_kmeans3 == 3, 0], X3[y_kmeans3 == 3, 1], s = 50, c = 'cyan', label = 'Cluster 3')
plt.scatter(X3[y_kmeans3 == 4, 0], X3[y_kmeans3 == 4, 1], s = 50, c = 'magenta', label = 'Cluster 4')
plt.scatter(X3[y_kmeans3 == 5, 0], X3[y_kmeans3 == 5, 1], s = 50, c = 'brown', label = 'Cluster 5')
plt.scatter(kmeans3.cluster_centers_[:, 0], kmeans3.cluster_centers_[:, 1], s = 100, c = 'yellow', label = 'Centroides')
plt.title('Clúster PCA de Exoplanetas peso a zona habitable')
plt.xlabel('PC 1')
plt.ylabel('PC 2')
plt.legend()
plt.savefig('clusterzh.png',dpi=400)
plt.show()
```

Figura 6. *Clustering* usando *sklearn*. Este es el caso para el primer *clustering*, el que no toma en cuenta la temperatura calculada. Se realizan seis clústers debido a que fue lo que se observó con el método del codo [5].

la Tierra.

```
#Calculamos el histograma y medidas de tendencia central de la masa de la lista obtenida arriba.
masa1=[]
for i in planetasTierra3:
    masa1.append(i[1]*317.7710843)
print(masa1[0])
print(np.mean(masa1))
print(np.var(masa1))
print(np.std(masa1))

fig=plt.figure(figsize=(15,4))
plt.hist(masa1, bins='auto')  # arguments are passed to np.histogram
plt.scatter(masa1[0],55,color='black',label='Tierra',s=200)
plt.title('Histograma de masa con peso en zona habitable')
plt.xlabel('Masa [Mt]')
plt.ylabel('Frecuencias')
plt.legend()
plt.savefig('histogramamasa.png',dpi=400)
plt.show()
```

Figura 7. Histogramas y medidas de tendencia central usando *matplotlib.pyplot* y *numpy*. Este es el caso para la masa del primer *clustering*, el que no toma en cuenta la temperatura calculada.

Comparación

Se crearon dos funciones [8 & 9] para comparar los datos obtenidos por los dos agrupamientos que se hicieron.

```python
#Función que compara y nos dice el porcentaje de la variable x2 que está contenido en x1
def comparar(x1,x2):
    contadorcompara=0
    for i in x1:
        for j in x2:
            if i==j:
                contadorcompara+=1
    return 100*contadorcompara/len(x2)
```

Figura 8. Función que compara las veces que un dato en x1 es igual a un dato en x2, para este caso se usó para comparar los nombres de los exoplanetas de las distintas tablas obtenidas, la función regresa el porcentaje contenido de x2 en x1.

```python
def analogolimpios(x,limpio):
    analogo=[]
    for i in x:
        for j in limpio:
            if i==j:
                analogo.append(i)
    return analogo
```

Figura 9. Función que regresa una lista con los datos que comparten otras dos listas. Se usó para obtener un arreglo que contenga los exoplanetas de la segunda base de datos (de comparación) que también estaban en los datos limpios (sin NaN) para los dos casos (con y sin temperatura calculada).

Resultados

Caso 1: Masa, radio, semieje mayor del planeta, masa y radio de la estrella

Para este caso, con el método del codo decidimos hacer seis clústers [10].

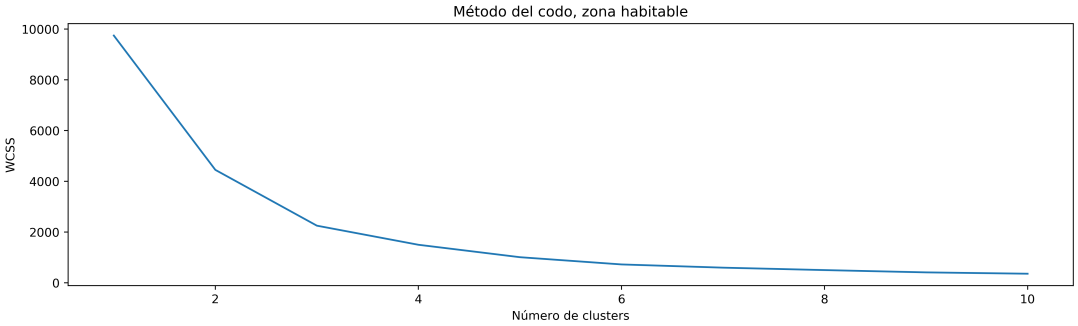

Figura 10. Método del codo para el primer caso, converge en seis. *WCSS* significa *"Within cluster sum of squares by cluster"*, es un algoritmo que utiliza la distancia euclidiana para encontrar la lejanía de los datos con sus centroídes o centros del clúster. A partir de seis clústers, la distancia de los centroides a los datos de sus respectivos grupos ya no varía significativamente.

Realizamos el agrupamiento con [6] para seis clústers [11].

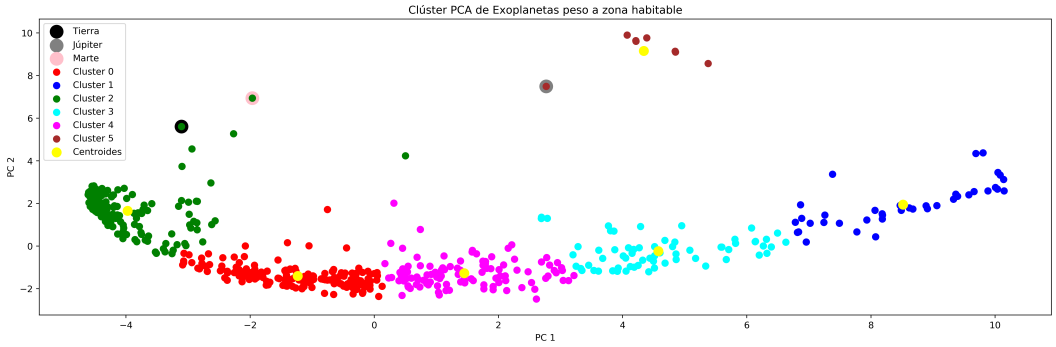

Figura 11. Clustering para el primer caso, seis clústers. La Tierra está en el cluster 2 por lo que ese es el que se va a analizar.

En el clúster 2 que es donde está la Tierra, se obtuvieron 552 exoplanetas de los 152 que había antes de realizar el agrupamiento, son demasiados datos como para analizarlos uno a uno. Por esta razón se realizaron histogramas con el fin de observar el nivel de clasificación [12,14, 15 & 16].

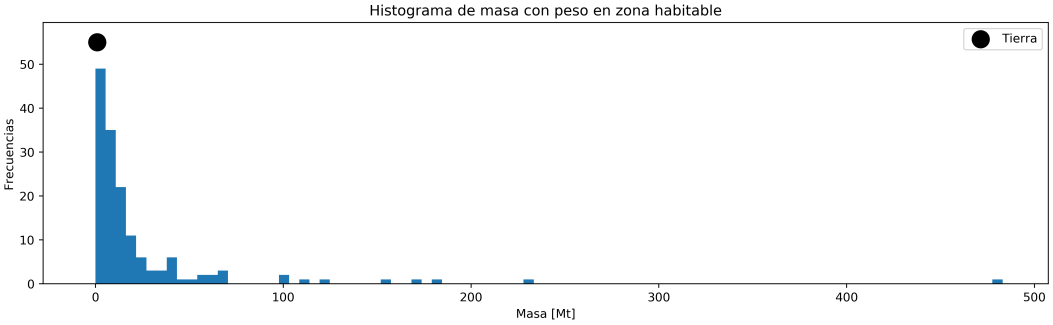

Figura 12. Histograma del clúster de la Tierra para la masa en el caso 1, la mayoría de los datos tienen la misma masa que la Tierra o al menos están cercanos.

La media de las masas es de 24.03 Mt (Masas terrestres) y la desviación estándar es de 50.79 Mt, por lo que aproximando en una distribución normal la Tierra con 1 Mt entra dentro de una desviación estándar, junto a las super Tierras y planetas subterranos. Aunque como se puede ver en el histograma, los extremos están extendidos, de hecho la varianza es de 2579.70 Mt^2.

La media de los radios es de 3.41 Rt (Radios terrestres) y la desviación estándar es de 7.85 Rt, con lo que también la Tierra entra dentro de una desviación estándar, en este caso la varianza no es muy grande 2.80 Rt^2

Para el caso del semieje mayor [15] existe un sesgo, los datos que se trabajaron, al limpiarlos perdieron todos aquellos planetas que no tuviesen el dato del radio.

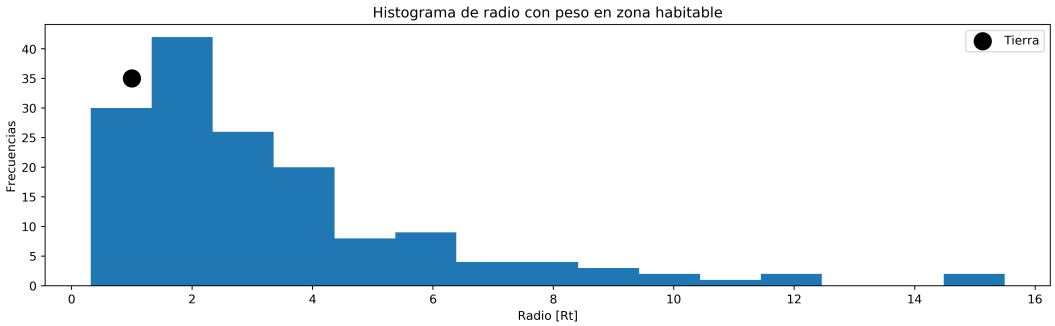

Figura 13. Histograma del clúster de la Tierra para radio en el caso 1, la mayoría de los datos tienen el misma radio que la Tierra o al menos están cercanos.

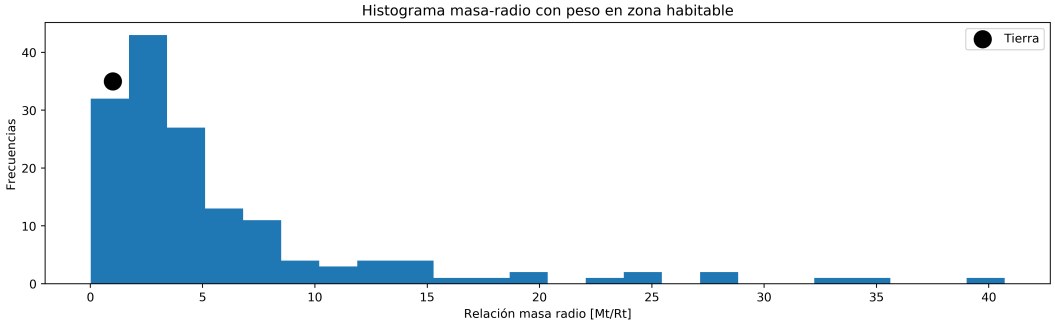

Figura 14. Histograma para el cluster de la Tierra, para la relación masa-radio en el caso 1, la mayoría de los datos están a menos de una desviación estándar de la Tierra.

Los radios se miden por tránsito, los planetas de radios pequeños comparados con su estrella (como la Tierra) deben de estár muy cerca de esta para que pueda ser medida una atenuación en la radiación que reciben los telescopios. A partir de esta atenuación es como se deduce el radio. Es por esto que la mayoría de los planetas del clúster de la Tierra tienen un semieje mayor de menos de 0.5 UA.

Para el semieje mayor la media es de 0.17 UA, la desviación estándar es de 0.29 y la varianza es de 0.08, por lo que la Tierra está mas allá de una desviación estándar, incluso más allá de dos. A la hora de realizar este clúster me percaté de que Marte tendía a irse al clúster de Júpiter, también la Tierra pero menos drásticamente.

Esto es debido a que las variables del Sol relacionaban más a estos planetas que las variables de masa y radio con los demás exoplanetas, por lo mismo de que en la base de datos no hay planetas con masa y radio similares a la de la Tierra y con un semieje mayor de más de 0.5 UA.

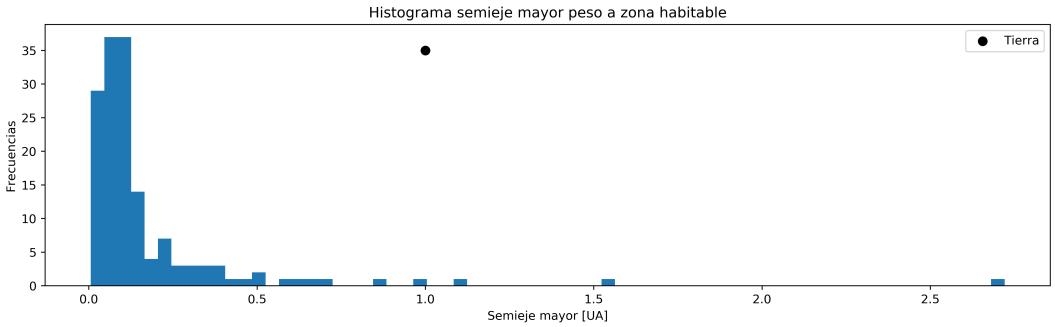

Figura 15. Histograma del clúster de la Tierra para el semieje mayor en el caso 1, hay un sesgo observacional.

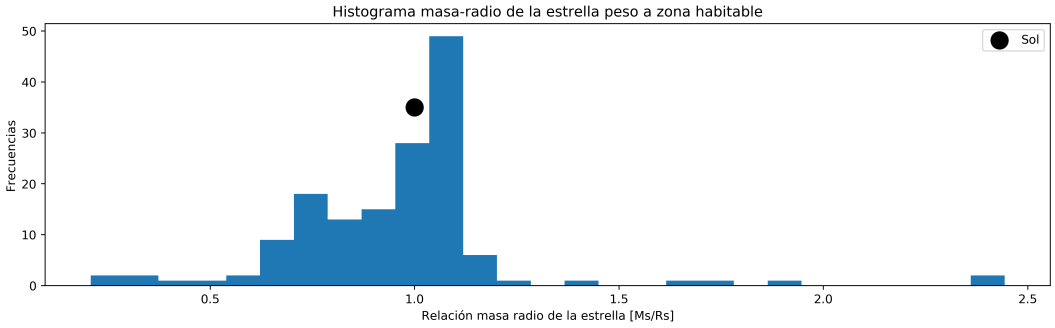

Figura 16. Histograma del clúster de la Tierra para la relación masa-radio de la estrella en el caso 1, la mayoría de los datos están a menos de una desviación estándar de la Tierra.

Para la relación masa-radio de la estrella, el Sol está casi en la media de 0.96 Ms/Rs (Masas solares/radios solares), la desviación estándar es de 0.28 Ms/Rs y la varianza es de 0.078 $(\text{Ms/Rs})^2$.

Caso 2: Masa, radio y la temperatura calculada

Para este caso, al igual que para el caso 1 se utilizó la masa y el radio, se le dió mas peso al radio [3] que a la masa, pero a lo que más peso se le asignó fue a la tempertura calulada, esto es debido a que esta variable define si un planeta es habitable o no.

Primero se realizó el *PCA*, después el método del codo [17] para determinar el número de clústers que fueron 6 y se realizó el clustering [18].

Después de hacer el agrupamiento, se obtuvo una tabla [19] con los planetas asignados al clúster de la Tierra y se aplicó un filtro para sólo quedarse con los que tuvieran temperaturas desde los 173 K hasta los 373 K, que son extremos muy optimistas para encontrar vida.

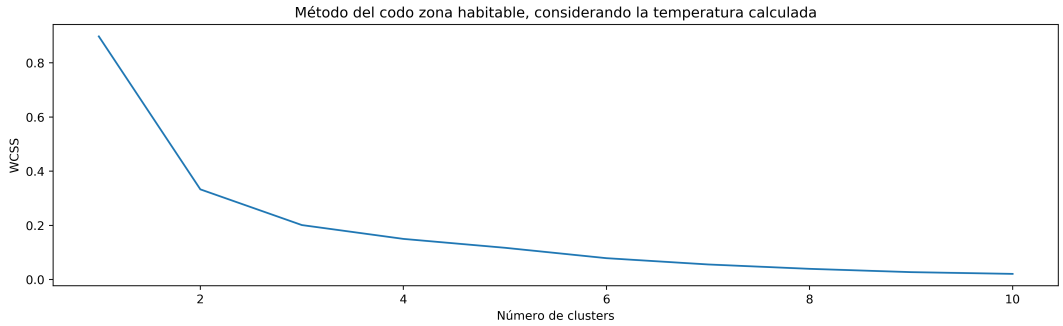

Figura 17. Método del codo para el segundo caso, converge en seis.

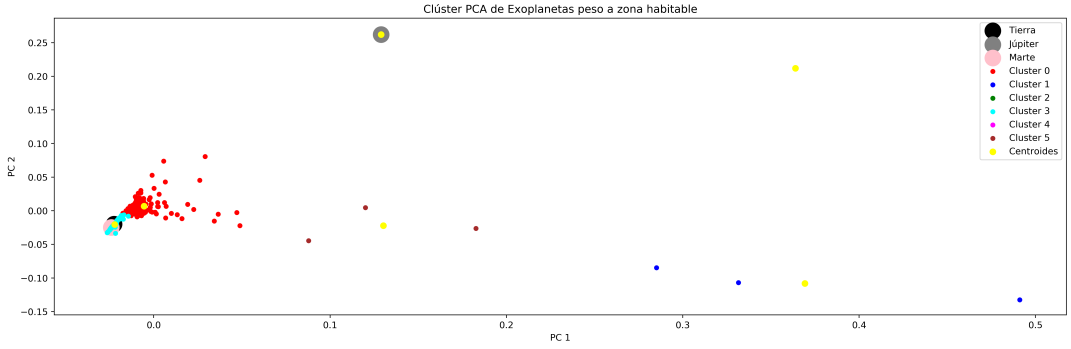

Figura 18. Clustering para el segundo caso, seis clústers. La Tierra está en el cluster 3 por lo que ese es el que se va a analizar.

El sistema planetario Trappist-1 está compuesto de siete planetas terrestres templados, de los cuales cinco (b, c, e, f y g) son similares en tamaño a la Tierra, y dos (d y h) son de tamaño intermedio entre Marte y la Tierra. Tres de los planetas (e, f y g) orbitan dentro de la zona habitable [*Gillon et al.*, 2017].

	# name	mass [Mj]	radius [Rj]	temp_calculated [K]	cluster
0	Earth	0.003147	0.089328	287.0	3
1	Mars	0.000338	0.047577	227.0	3
2	K2-3 c	0.006600	0.165000	344.0	3
3	K2-3 d	0.034900	0.135000	282.0	3
4	LHS 1140 b	0.020900	0.128000	230.0	3
5	TRAPPIST-1 c	0.004340	0.094210	341.9	3
6	TRAPPIST-1 d	0.001300	0.068900	288.0	3
7	TRAPPIST-1 e	0.002000	0.081900	251.3	3
8	TRAPPIST-1 f	0.002100	0.093230	219.0	3
9	TRAPPIST-1 g	0.004220	0.100500	198.6	3

Figura 19. Planetas del caso 2 al aplicar el filtro de temperatura.

LHS 1140b es un planeta extrasolar del tipo rocoso que orbita alrededor de la estrella LHS-1140, una estrella enana roja de tipo espectral M4.5V, en la constelación de Cetus (la ballena). Este planeta orbita alrededor de su sol a una distancia relativamente cercana, de tal manera que aunque la estrella emite una luminosidad muy baja, el planeta está ubicado en la llamada zona de habitabilidad de la estrella, donde si se dan

las condiciones adecuadas, podría albergar mares de agua líquida en su superficie [*Dittmann et al.*, 2017].

K2-3 d podría ser un objeto similar a la Tierra o un supervenus, dada su ubicación en el límite interno de la zona habitable de su estrella. Como consecuencia, sus altas temperaturas estimadas lo sitúan entre los mesoplanetas y los termoplanetas [*Mendez*, 2011].

Para el caso 2 también se realizaron histogramas, pero como para esta tabla no había tantos datos, se decidió solo aplicar el filtro de temperatura y describir brevemente los planetas obtenidos [19].

Comparación

Se aplicó un análogo de un filtro a los datos del archivo phl_hec_all_confirmed.csv [3], esto se hizo porque se observó que al limpiar la primer base de datos, perdiamos más del 80 % de exoplanetas, entonces a la hora de comparar, ese sesgo nos impedía sacar conclusiones.

El filtro consistia en comparar la base de datos de exoplanitas clasificados en habitables y no habitables [3] con la base de datos utilizada para el clustering después de limpiarla de NaN. Se hizo un nuevo archivo de comparación que incluía solo los datos de la primer base de datos (sin NaN)que también estaban en la segunda.

Al comparar los datos del caso 1 y 2 con el análogo de cada uno obtuvimos las siguientes tablas que contienen los planetas confirmados como habitables [20 & 21]:

Son casi los mismos planetas que se obtuvieron al aplicar el filtro de la temperatura, de hecho, después de aplicar el filtro análogo a los datos de la tabla de comparación, el 100 % de dicho análogo está contenido en nuestos clústers de la Tierra para el caso 1 y 2.

	Exoplanetas 1
0	K2-18 b
1	Kepler-22 b
2	Kepler-62 e
3	Kepler-62 f
4	LHS 1140 b
5	TRAPPIST-1 d
6	TRAPPIST-1 e
7	TRAPPIST-1 f
8	TRAPPIST-1 g

Figura 20. Planetas del caso 1 confirmados.

	Exoplanetas 2
0	LHS 1140 b
1	TRAPPIST-1 d
2	TRAPPIST-1 e
3	TRAPPIST-1 f
4	TRAPPIST-1 g

Figura 21. Planetas del caso 2 confirmados.

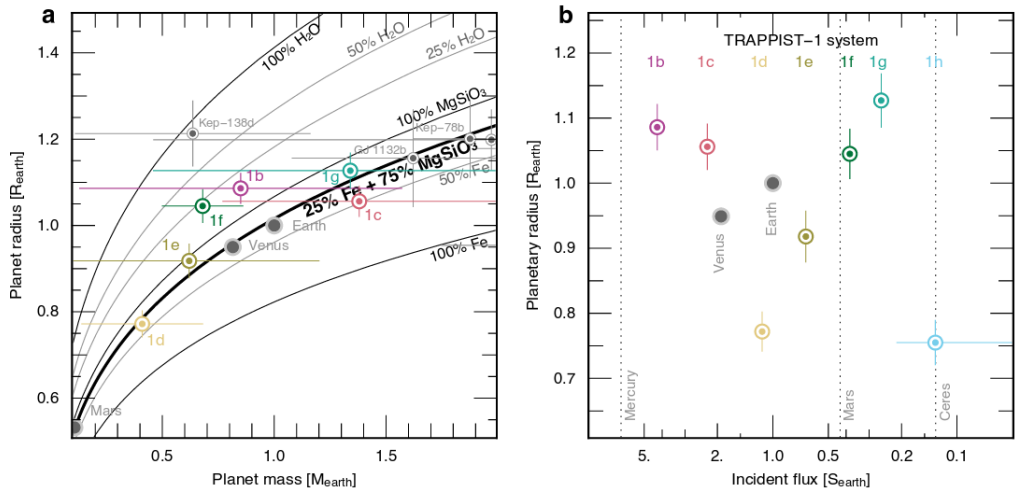

Figura 22. Izquierda: Composición esperada de acuerdo a la masa y radio de los planetas de Trappist-1 (b, c, d, e, f y g), Tierra, Venus, Marte, GJ 1132B, Kep-138d y Kep-78b. Derecha: Flujo incidente contra radio para los planetas de Trappist-1 (b, c, d, e, f, g y h), Tierra y Venus [*Gillon et al.*, 2017].

Podemos observar en la figura [22], que Trappist-1 d,e y g tienen una composición similar a la Tierra, principalmente silicatos y hierro, mientras que Trappist-1 f está compuesto por una mayor cantidad de agua, 25 %. En cuanto al flujo incidente, d y e reciben prácticamente el mismo flujo que la Tierra, mientras que f y g reciben un poco menos que Marte.

Conclusiones

Cada caso tuvo sus ventajas y desventajas, por un lado al usar planetas con los datos del radio, nos encontramos con un sesgo que nos llevaba a una variabilidad alta para el clustering 1 en la variable del semieje mayor, mientras que al usar la temperatura calculada, era justo esta la variable problemática y tuvimos que usar un filtro.

Aun así, en ambos casos llegamos a planetas similares, y en todos los casos se repitieron 5 de ellos, lo que nos lleva a la conclusión de que fueron bien hechas las suposiciones ya que a pesar de tener distintos inconvenientes todas las agrupaciones y filtros nos llevaron a casi los mismos resultados. Además al comparar con la literatura científica comprobamos que los 5 planetas son habitables y parecidos a la Tierra.

Referencias

Cuadras, C. M., *Nuevos métodos de análisis multivariante*, CMC Editions, 2007.

Dittmann, J. A., et al., A temperate rocky super-earth transiting a nearby cool star, *Nature*, *544*(7650), 333–336, 2017.

Gillon, M., et al., Seven temperate terrestrial planets around the nearby ultracool dwarf star trappist-1, *Nature*, *542*(7642), 456–460, 2017.

Mendez, A., A thermal planetary habitability classification for exoplanets, http://phl.upr.edu/library/notes/athermalplanetaryhabitabilityclassificationforexoplanets, accedido 07-01-2018, 2011.

Wolszczan, A., et al., Confirmation of earth-mass planets orbiting the millisecond pulsar psr b1257+12, *Science-AAAS-Weekly Paper Edition-including Guide to Scientific Information*, *264*(5158), 538–542, 1994.