MISIÓN “CATÁLOGO EXOPLANETARIO” – PRÁCTICA K-MEANS Y CLASIFICACIÓN CON ÁRBOLES Y SVM

José Nicolás García Castillo

Digitech FP Málaga

Sistemas de Aprendizaje Automático

26/11/2025

ÍNDICE

1. Objetivos y contexto

El objetivo principal de la misión de la Patrulla de Análisis de Exoplanetas (PAE) consiste en analizar, buscar y procesar un conjunto de datos recibidos de sondas automáticas, con el propósito de encontrar patrones ocultos en los exoplanetas que se encuentran en una tabla de datos.

Para la realización de la misión, se han seguido y realizado los siguientes pasos:

1. Utilización de algoritmos de agrupamiento (clustering), para proponer una clasificación de los planetas en función de sus propiedades físicas, cuyas características se mencionarán más adelante.

2. Realización de un entrenamiento de modelos (árboles de decisión y SVM), cuya capacidad consiste en replicar la clasificación de los planetas de forma automática para futuros descubrimientos.

Con estos pasos, lo que queremos conseguir es automatizar la información de los exoplanetas, de la manera más precisa y eficiente posible, para así obtener datos fiables para el estudio y cumplir el objetivo de la misión.

2. Selección de variables

A raíz de la información que disponemos sobre los exoplanetas en un catálogo de la NASA, tenemos más de 70 columnas de datos disponibles. Para la selección de nuestras variables, se ha realizado una limpieza de valores nulos, y se han escogido las siguientes 3 variables físicas, que han sido de gran ayuda para el análisis:

- **Masa del planeta (pl_bmassj):** medidas en masas de Júpiter (masa x Júpiter), que han sido importantes para poder distinguir entre planetas rocosos (ligeros) y gigantes gaseosos (pesados).

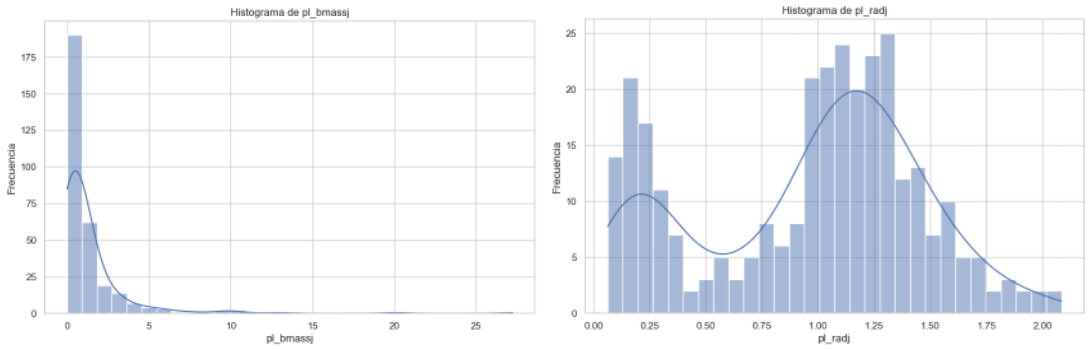
- **Radio del planeta (pl_radj):** medidas en radios de Júpiter (radio x Júpiter), que han servido para la definición del tamaño físico y el volumen del planeta.

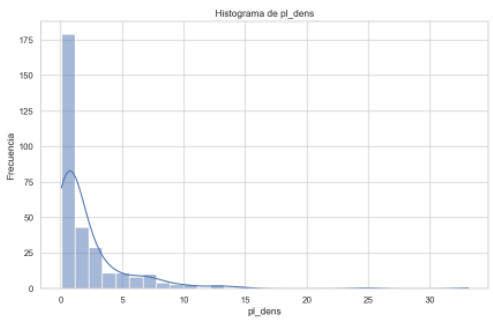
- **Densidad del planeta (pl_dens):** medidas en gramos por centímetro cúbico, que nos permiten entender la densidad de un planeta a través del radio y de la masa que se ha obtenido con los datos que disponemos.

El motivo principal detrás de la elección de las variables mencionadas es que, estas variables tienen una gran correlación física. En este caso, existe una relación inversa entre el radio de un planeta y su densidad. Para ello, nos hemos guiado de la ley de densidad, que establece la densidad de un planeta está relacionada con su masa y volumen, por lo que la densidad es directamente proporcional a la masa e inversamente proporcional al radio al cubo, ya que se necesita el radio del planeta para calcular su volumen, para después calcular la densidad.

3. K-Means

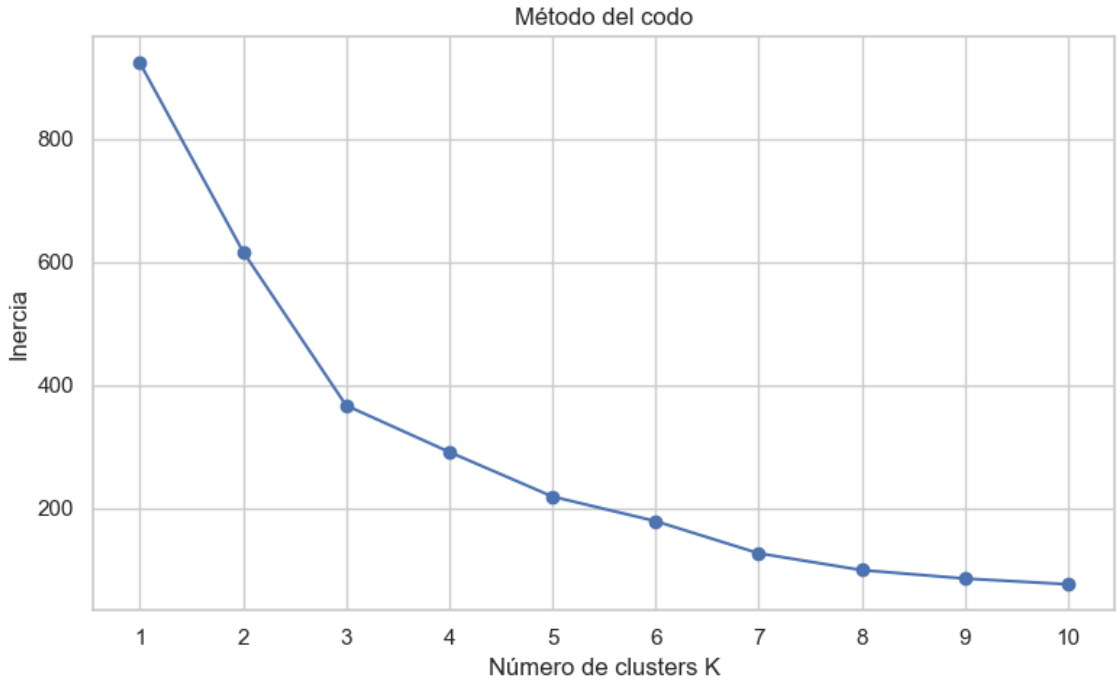
Para determinar estas medidas, lo primero que se realiza es evaluar el comportamiento de los datos de las variables que hemos escogido, y detectar la presencia de outliers, que pueden alterar el proceso de la formación del clustering. De las variables estudiadas, obtenemos los siguientes histogramas:





De izquierda a derecha, la primera imagen corresponde a la masa de los planetas, y se puede observar que la mayoría de exoplanetas se concentran en valores bajos o medios, pero aparecen algunos con masa muy elevada, que se alejan del resto de planetas. La siguiente imagen representa el radio de los planetas, y se puede interpretar que existe un núcleo de planetas con radios cercanos al de Júpiter, mientras que unos pocos presentan radios extremadamente grandes o muy pequeños en comparación a otros. Por último, en la imagen de abajo, que corresponde a la densidad, se ve una gran dispersión, con muchos valores moderados y, en algunos casos, densidades o muy altas o bajas. Por lo tanto, los posibles outliers que se nos pueden presentar son los planetas con masas y radios muy grandes, y los planetas con densidades muy grandes, que representan puntos aislados del gráfico, y no tienen un comportamiento normal al resto de exoplanetas.

Una vez que se han examinado los datos, se realiza un escalado de los datos, ya que los sensores no miden todas las magnitudes de la escala. El escalado es importante realizarlo porque este algoritmo se basa directamente en el cálculo de distancias entre los puntos y los centroides. Si las variables tienen rangos muy diferentes, uno se sobrepondrá a otro, por lo que los clústeres no serán fiables. Este escalado se realizará mediante el método del codo, y obtenemos la siguiente gráfica:

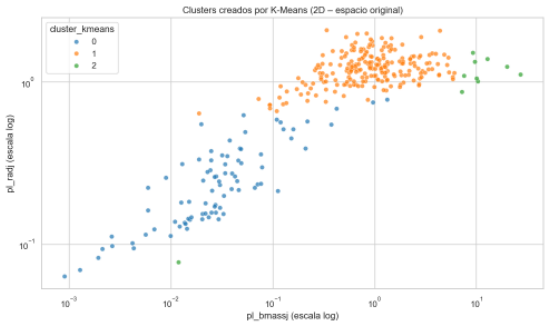


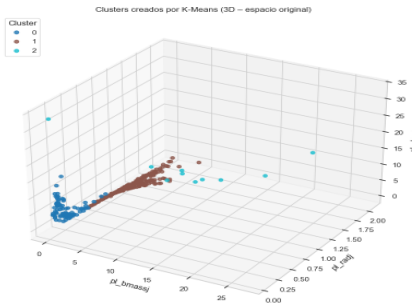
Para obtener K-Means, se elige el valor K situado en el "codo" de la curva, donde la mejora deja de ser significativa, cuando se inicia un cambio de pendiente. Si observamos la gráfica, podemos interpretar que el cambio de pendiente se produce en K = 3.

Ahora averiguaremos las frecuencias de cada clúster, y visualizaremos los clusters en 2D y en 3D, y se obtiene lo siguiente:

```python
k_elegido = 3
print("Usaremos K =", k_elegido)

kmeans_final = KMeans(n_clusters=k_elegido, random_state=42, n_init=10)
clusters = kmeans_final.fit_predict(X_km_scaled)

# Guardamos el número de cluster en el DataFrame
df_km["cluster_kmeans"] = clusters

print("\nFrecuencia de cada cluster:")
freq_clusters = df_km["cluster_kmeans"].value_counts().sort_index()
print(freq_clusters)
```

```
Usaremos K = 3

Frecuencia de cada cluster:
cluster_kmeans
0     83
1    215
2     10
Name: count, dtype: int64
```

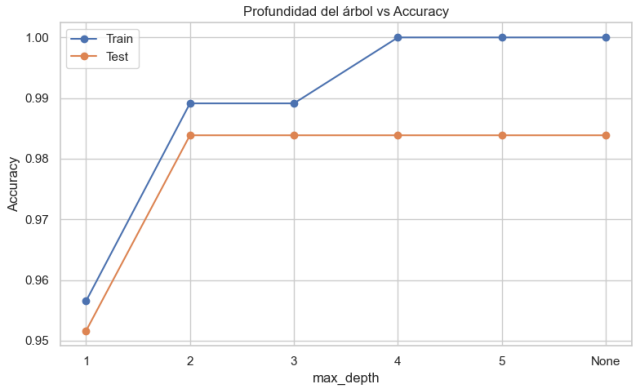




Observando los datos obtenidos, podemos concluir que los clústeres más poblados son los planetas que agrupan valores moderados, mientras que el menos poblado suelen estar formados por exoplanetas con masas, radios o densidades muy grandes. Además, también podemos observar que los clústeres no están perfectamente separados, por lo que hay solapamiento, lo que nos indica que algunas características de los exoplanetas son similares, aunque los que representan medidas más extremas tienen una separación más clara. Respecto a las diferencias entre las vistas en 2D y 3D, se observa que la vista 3D mejora la diferenciación de los clústeres que en 2D, lo que permite identificar con mejor claridad los grupos principales.

4. Árbol de decisión

Para automatizar la clasificación realizada por K-Means, se ha realizado un algoritmo de entrenamiento por árbol de decisión, en la que se han probado diferentes profundidades para optimizar el modelo, y así evitar el sobreajuste. Mediante la precisión de los datos a base de la profundidad que se han probado, se ha realizado una gráfica en el que se compara la profundidad del árbol de decisión con la precisión de los datos, y así obtenemos de manera definitiva la profundidad óptima del árbol, como se obtiene en las siguientes imágenes:

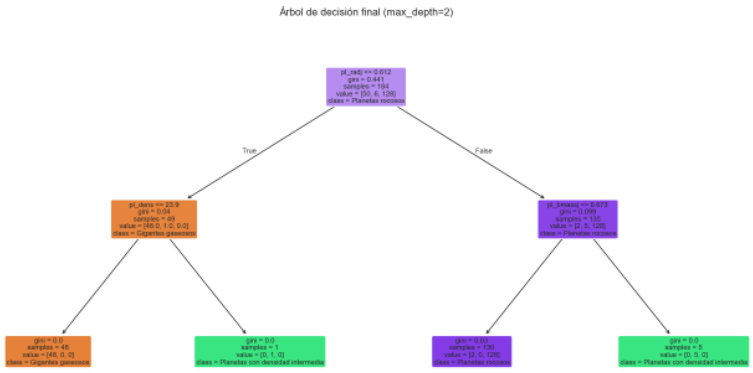


max_depth	acc_train	acc_test
1	0.957	0.952
2	0.989	0.984
3	0.989	0.984
4	1.000	0.984
5	1.000	0.984
None	1.000	0.984

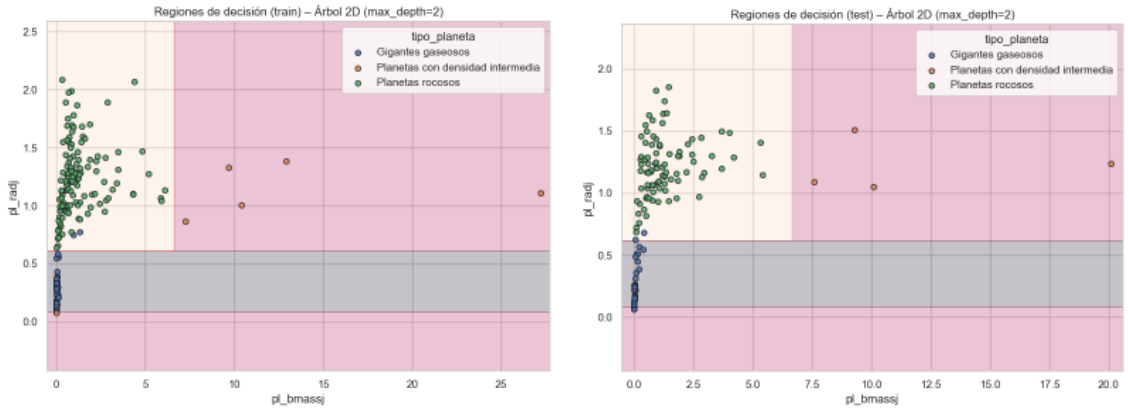
Como podemos observar en la primera imagen, la profundidad óptima del árbol de decisión es de 2, lo que nos da unas precisiones muy altas, como muestra la siguiente imagen:

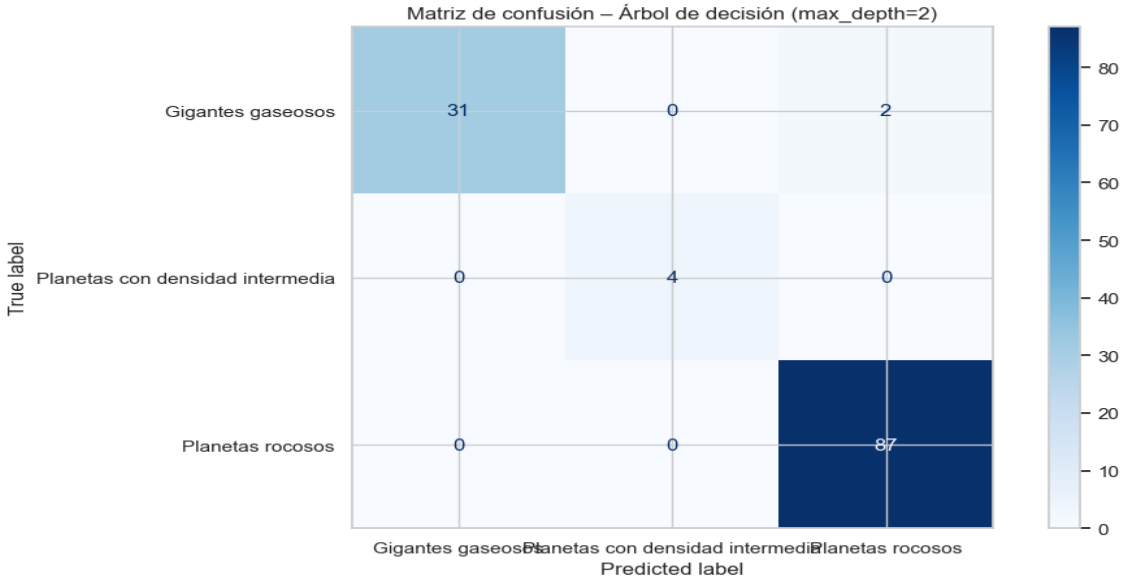
```
Árbol final con max_depth = 2
Accuracy (train): 0.989
Accuracy (test) : 0.984
```

Por lo tanto, el árbol de decisión final que se obtiene con todos estos datos es el siguiente:



Una vez que se ha obtenido el árbol de decisión, realizamos las regiones de decisión en 2D y la matriz de confusión:





Con estos datos, llegamos a la siguiente conclusión: la gran mayoría de planetas están calificados de forma correcta, pero hay unos pocos que cumplen unas características diferentes y que se alejan del resto de planetas que siguen las mismas características, lo que da lugar a que la precisión no sea igual a 1.

5. SVM

Para estudiar las fronteras de decisión entre los tipos de planetas, se ha realizado un modelo de entrenamiento de modelos SVM. Para ello, se ha hecho uso de tres kernels diferentes:

- **Linear:** genera fronteras rectas, y es rápido y fácil de interpretar.

- **Polinómico:** genera fronteras de decisión de forma polinómica, permite modelar relaciones más complejas, y su dificultad depende del grado del polinomio

- **RBF:** genera fronteras de decisión muy flexibles y no lineales, y suele ser el kernel más usado.

Teniendo en cuenta los datos del árbol de decisión, se ha realizado los modelos SVM con C = 100, grado 3 y gamma escalado. Teniendo en cuenta la precisión, hemos obtenido que tanto el SVM linear como el RBF obtienen la misma precisión, con un valor de 0,984. En cambio, si reducimos la C, el modelo linear tiende a tener más precisión que el RBF. Esto ocurre porque, cuanto mayor sea el valor de C, mayor será el número de fronteras y, por lo tanto, mayor será la precisión para todos los modelos. Así que, para menor valor de C, el modelo linear es el mejor, ya que es más preciso, y para mayor valor de C, tanto el modelo linear como el de RBF serán modelos mejores para analizar.

```
======================================================================
SVM con kernel = linear
======================================================================
Accuracy en test: 0.984

Informe de clasificación:
                              precision    recall  f1-score   support

              Gigantes gaseosos      1.000     0.970     0.985        33
Planetas con densidad intermedia      1.000     0.750     0.857         4
               Planetas rocosos      0.978     1.000     0.989        87

                       accuracy                          0.984       124
                      macro avg      0.993     0.907     0.943       124
                   weighted avg      0.984     0.984     0.983       124
```

```
======================================================================
SVM con kernel = rbf
======================================================================
Accuracy en test: 0.984

Informe de clasificación:
                              precision    recall  f1-score   support

              Gigantes gaseosos      1.000     0.939     0.969        33
Planetas con densidad intermedia      1.000     1.000     1.000         4
               Planetas rocosos      0.978     1.000     0.989        87

                       accuracy                          0.984       124
                      macro avg      0.993     0.980     0.986       124
                   weighted avg      0.984     0.984     0.984       124
```

```
================================================================
SVM con kernel = poly
================================================================
Accuracy en test: 0.960

Informe de clasificación:
                               precision    recall  f1-score   support

              Gigantes gaseosos      0.967     0.879     0.921        33
Planetas con densidad intermedia      0.600     0.750     0.667         4
               Planetas rocosos      0.978     1.000     0.989        87

                       accuracy                          0.960       124
                      macro avg      0.848     0.876     0.859       124
                   weighted avg      0.962     0.960     0.960       124
```

```
Resumen de accuracy en test por kernel:
- linear: 0.984
- rbf    : 0.984
- poly   : 0.960
```

```
Resumen de accuracy en test por kernel:
- linear: 0.984
- rbf    : 0.992
- poly   : 0.960
```

```
Resumen de accuracy en test por kernel:
- linear: 0.976
- rbf    : 0.968
- poly   : 0.944
```

A la izquierda, C = 100; a la derecha, C = 10, y en el centro C = 1.

6. Conclusión

En resumen, el aprendizaje automatizado se ha utilizado para realizar un análisis completo de las clases de exoplanetas y compuesto por qué se dividen en diferentes clases. Se puede concluir que, en primer lugar, los exoplanetas constituyen categorías no totalmente discretas, pero tienen una gran relación con la masa, el radio y la densidad. Sin embargo, hay algunos tipos principales, como gigantes gaseosos a la distancia de Júpiter, planetas más densos y compactos y, aparte, mundos extremos que se comportan como valores atípicos. Por lo tanto, algunos planetas siempre rompen la lógica general de las clasificaciones y, por tanto, pueden considerarse formas diferentes. Por otro lado, SVM ha permitido clasificar los exoplanetas en función de etiquetas conocidas, mostrando cómo el modelo aprende fronteras de decisión basadas en ejemplos previos. Se ha observado que los kernels no lineales (especialmente RBF) ofrecen una mejor capacidad para distinguir clases complejas, mientras que el modelo lineal resulta más limitado cuando las relaciones entre variables no son simples.