Análisis Exploratorio y Modelado Orbital de Asteroides

Mayte Torres Hernández

Agosto 2025

Objetivo general

Aplicar herramientas de Python para realizar un análisis exploratorio, visualización, clasificación o predicción sobre asteroides o cometas a partir de sus datos orbitales y físicos, utilizando datos del Minor Planet Center. Este proyecto busca fomentar la interpretación de datos reales, la limpieza de datos con estructuras complejas y la aplicación de un modelo de aprendizaje automático en un contexto astronómico.

The Minor Planet Center

El Centro de Planetas Menores (MPC) es el único centro mundial para la recepción y distribución de mediciones de posición de planetas menores, cometas y satélites naturales irregulares exteriores de los planetas mayores. El MPC es responsable de la identificación, designación y cálculo de la órbita de todos estos objetos. Esto implica mantener los archivos maestros de observaciones y órbitas, rastrear al descubridor de cada objeto y anunciar los descubrimientos al resto del mundo mediante circulares electrónicas y un extenso sitio web. El MPC opera en el Observatorio Astrofísico Smithsoniano, bajo los auspicios de la División F de la Unión Astronómica Internacional (UAI).

Carga y procesamiento de datos

Los datos se descargaron de la página de The Minor Planet Center en formato de bloque de notas. Para leer el conjunto de datos se utilizó la librería `pandas`:

- **Primero:** Se realizó un acomodo para las columnas de las variables a estudiar (Figura 1).
- **Segundo:** Se creó una lista de los nombres de cada columna (Figura 2).

```
colspecs = [
    (0, 7),     # Número o designación provisional
    (8, 13),    # Magnitud absoluta H
    (14, 19),   # Parámetro de pendiente G
    (20, 25),   # Época (en formato empaquetado)
    (26, 35),   # Anomalía media
    (37, 46),   # Argumento del perihelio
    (48, 57),   # Longitud nodo ascendente
    (59, 68),   # Inclinación
    (70, 79),   # Excentricidad
    (80, 91),   # Movimiento medio diario
    (92, 103),  # Semieje mayor
    (105, 106), # Parámetro de incertidumbre
    (107, 116), # Referencia
    (117, 122), # Número de observaciones
    (123, 126), # Número de oposiciones
    (127, 131), # Año inicial o longitud del arco
    (132, 133), # Separador '-' o espacio
    (133, 137), # Año final o 'days'
    (137, 141), # Error cuadrático medio (rms)
    (142, 145), # Indicador grueso de perturbadores
    (146, 149), # Indicador preciso de perturbadores
    (150, 160), # Nombre de computadora
    (166, 194), # Designación legible
    (194, 202), # Fecha última observación (YYYYMMDD)
]
```

Figura 1: Acomodo de variables.

```
nombres_columnas = [
    "ID", "H", "G", "Epoch", "MeanAnomaly", "ArgPerihelion",
    "LongAscNode", "Inclination", "Eccentricity", "MeanMotion",
    "SemiMajorAxis", "Uncertainty", "Reference", "NObs",
    "NOppositions", "ObsStart", "Dash", "ObsEnd",
    "RMS", "CoarsePerturbers", "PrecisePerturbers",
    "Computer", "ReadableDesignation", "LastObsDate"
]
```

Figura 2: Lista de las variables.

- **Tercero:** Se leyó el conjunto de datos utilizando:

  `pandas.read_fwf()`

  con formato de columnas de ancho fijo.

LastObsDate.1	Orden_descubrimiento	ID	H	G	Epoch	MeanAnomaly	ArgPerihelion	LongAscNode	Inclination	Eccentricity	...	NObs	NOppositions
1979-12-15	811554	J79X00B	18.60	0.15	K2555	250.37720	77.08223	84.35590	24.55263	0.709683	...	16.0	1
1994-04-10	811634	J94G00K	24.20	0.15	K2555	22.36781	112.98846	14.24321	5.76115	0.611780	...	10.0	1
1996-01-29	812027	J96B00T	22.79	0.15	K2555	46.40556	328.21250	296.80064	12.03482	0.831897	...	21.0	1
1997-11-23	812228	J97V06G	19.60	0.15	K2555	201.84662	250.92727	51.63275	18.49664	0.564574	...	50.0	1
1998-07-31	812304	J98O04P	24.00	0.15	K2555	322.47236	167.73315	126.61814	13.40025	0.536992	...	21.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2025-07-23	226254	M6255	16.60	0.15	K2555	278.04723	293.53641	91.71610	5.95651	0.149878	...	1004.0	14
2025-07-23	95375	95376	15.46	0.15	K2555	79.96135	139.77020	76.22783	3.96844	0.030934	...	1188.0	19
2025-07-23	223648	M3649	16.25	0.15	K2555	321.29452	275.26625	64.26064	5.22174	0.169441	...	832.0	20
2025-07-23	8233	8234	13.13	0.15	K2555	94.91238	6.01227	184.18058	1.71606	0.010286	...	5135.0	28
2025-07-23	109442	A9443	15.14	0.15	K2555	250.34966	72.35058	352.80356	12.27315	0.162259	...	1442.0	17

72177 rows × 24 columns

- **Cuarto:** Se convirtió la columna `LastObsDate` a un formato legible, ya que venía en la forma `20241101`, utilizando `pd.to_datetime()` y el tipo `date` del módulo `datetime`.
- **Quinto:** Se limpiaron las columnas con datos nulos usando:

  `tabla1.isna().sum()`

  para identificar columnas con valores faltantes y eliminarlos.

Datos utilizados

Las variables utilizadas incluyen 14 numéricas y 10 tipo objeto.

```
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Orden_descubrimiento  72177 non-null  int64
 1   ID                    72177 non-null  object
 2   H                     72177 non-null  float64
 3   G                     72177 non-null  float64
 4   Epoch                 72177 non-null  object
 5   MeanAnomaly           72177 non-null  float64
 6   ArgPerihelion         72177 non-null  float64
 7   LongAscNode           72177 non-null  float64
 8   Inclination           72177 non-null  float64
 9   Eccentricity          72177 non-null  float64
 10  MeanMotion            72177 non-null  float64
 11  SemiMajorAxis         72177 non-null  float64
 12  Uncertainty           72177 non-null  float64
 13  Reference             72177 non-null  object
 14  NObs                  72177 non-null  float64
 15  NOppositions          72177 non-null  int64
 16  ObsStart              72177 non-null  int64
 17  Dash                  72177 non-null  object
 18  ObsEnd                72177 non-null  object
 19  RMS                   72177 non-null  float64
 20  CoarsePerturbers      72177 non-null  object
 21  PrecisePerturbers     72177 non-null  object
 22  Computer              72177 non-null  object
 23  ReadableDesignation   72177 non-null  object
dtypes: float64(12), int64(3), object(9)
```

Entre las más relevantes se encuentran: H (magnitud absoluta), G (parámetro de pendiente), `MeanAnomaly`, `Eccentricity`, `Inclination`, `MeanMotion`, `SemiMajorAxis`, `NObs`, `NOppositions`, entre otras.

Análisis univariado

Se generaron histogramas para las variables numéricas, verificando su distribución. Ninguna de las variables sigue una distribución normal según la prueba de Shapiro–Wilk.

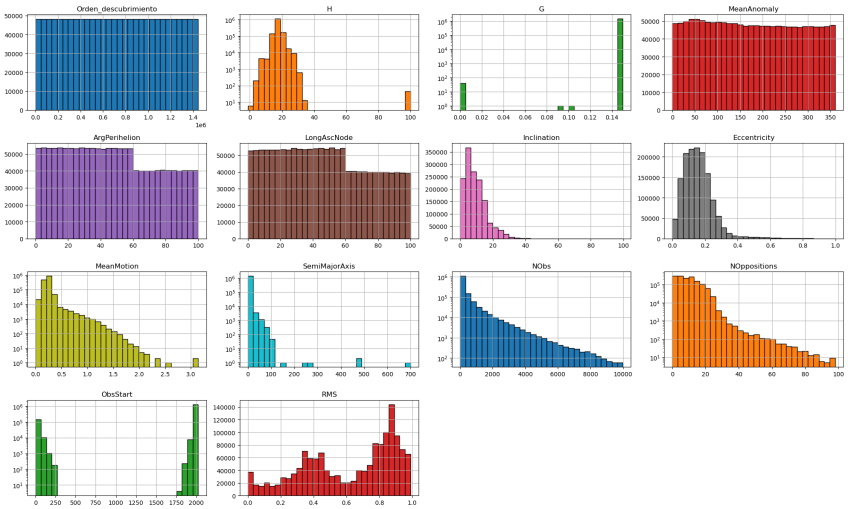
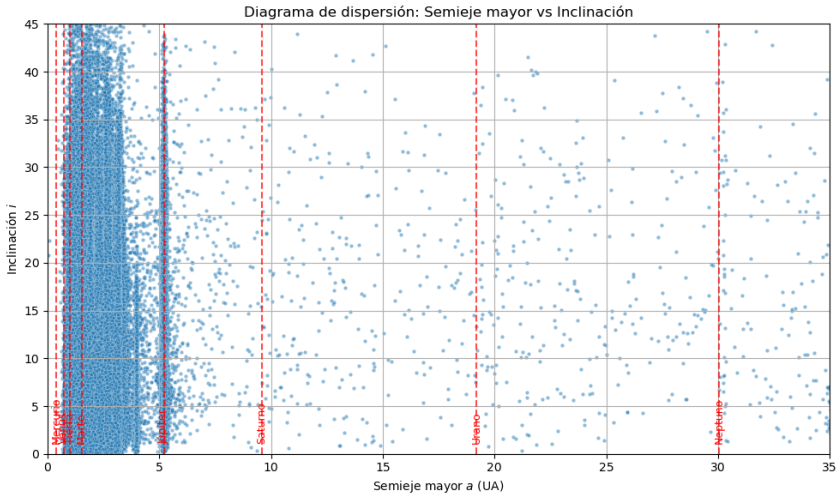


Figura 3: Distribuciones de variables numéricas.

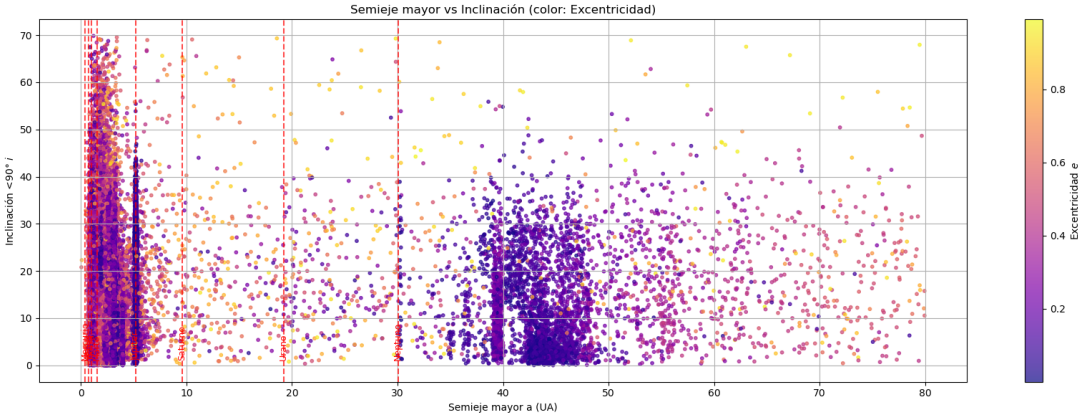
Análisis bivariado

Se realizaron diagramas de dispersión para pares de variables para identificar tendencias.



¿Qué se observa en este gráfico?

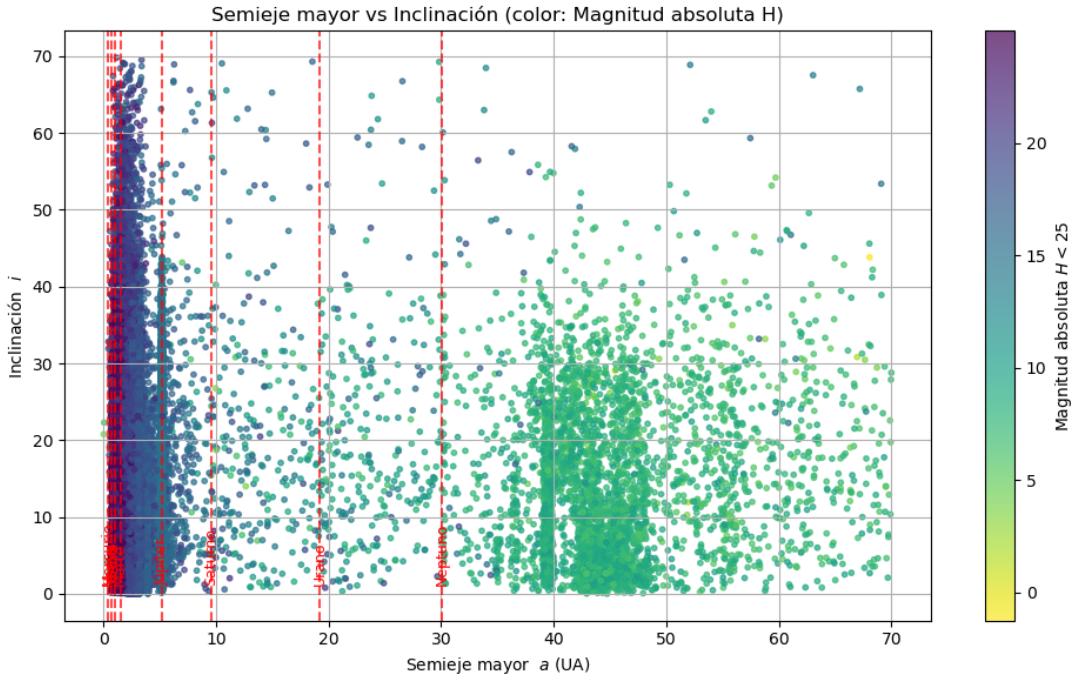
Concentración alrededor de asteroides de $2 - 3{,}5\ UA$ e inclinaciones bajas.
La descripción corresponde al cinturón principal de asteroides, una región del sistema solar ubicada entre las órbitas de Marte y Júpiter, donde se concentra una gran cantidad de asteroides con inclinaciones orbitales bajas. Esta zona se encuentra aproximadamente entre 2 y 3.5 unidades astronómicas (UA) del Sol.



¿Qué se observa en este gráfico?

Zonas con excentricidad baja (color más oscuro) entonces se tiene órbitas casi circulares, como muchos asteroides del cinturón principal.

Los puntos más claros muestran órbitas más elípticas, como cometas, objetos dispersos o centauros.

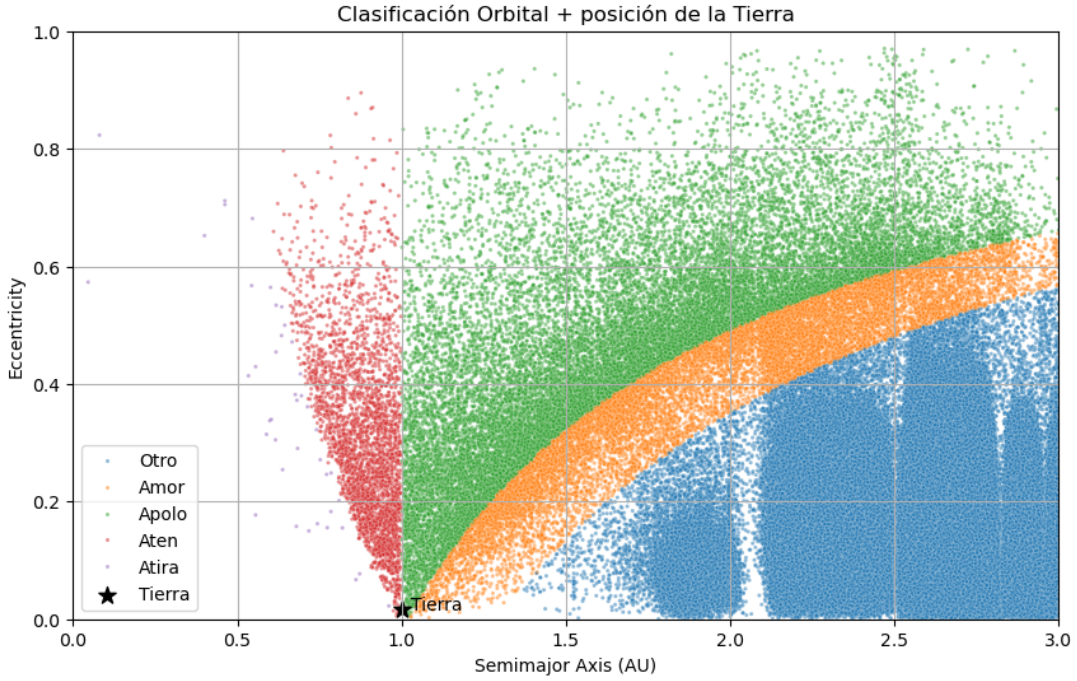


¿Qué se observa?

Concentración de objetos grandes (H bajo) cerca de $2-4$ UA: el cinturón de asteroides.

Objetos más pequeños (H alto) suelen dominar en regiones exteriores.

Se puede notar si ciertas regiones del sistema solar contienen objetos más grandes o si predominan los pequeños.

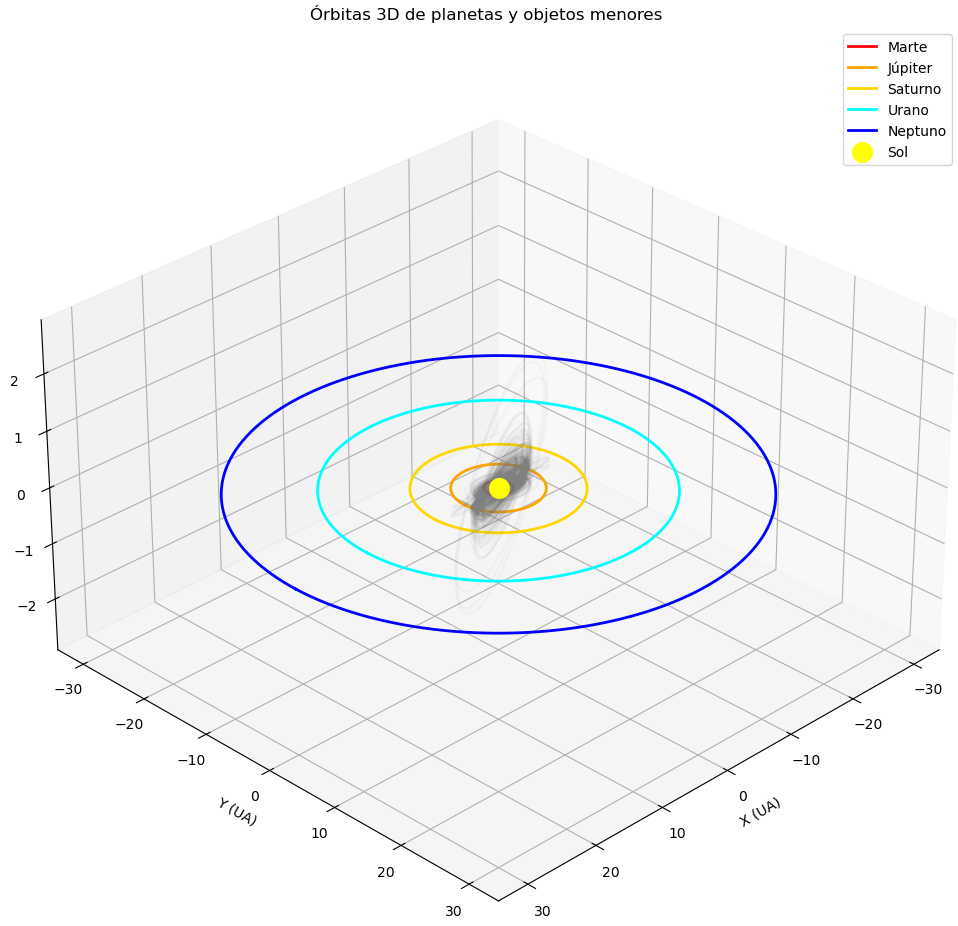


La gráfica muestra la clasificación de asteroides según su órbita. Los asteroides cercanos a la Tierra se clasifican dinámicamente tomando con base en dos parámetros orbitales:

$q = a(1 - e)$: perihelio = es el punto donde el objeto está más cerca del Sol

$Q = a(1 + e)$: afelio = es el punto donde está más lejos del Sol. tomando a : semieje mayor (en UA).

- **Atira:** $a < 1{,}0$ UA y $Q < 0{,}983$ UA.Asteroides completamente dentro de la órbita de la Tierra.
- **Aten:** $a < 1{,}0$ UA y $Q > 0{,}983$ UA. Asteroides con órbitas interiores, pero que cruzan la órbita terrestre.
- **Apolo:** $a > 1{,}0$ UA y $q < 1{,}017$ UA. Asteroides que cruzan la órbita de la Tierra desde fuera. Son los NEOs más numerosos.
- **Amor:** $a > 1{,}0$ UA y $1{,}017 < q < 1{,}3$ UA. Asteroides que se acercan a la órbita de la Tierra, pero no la cruzan.



Se usa Skyfield para obtener el semieje mayor a actual de cada planeta.

Dibuja órbitas circulares aproximadas en el plano eclíptico para los planetas.

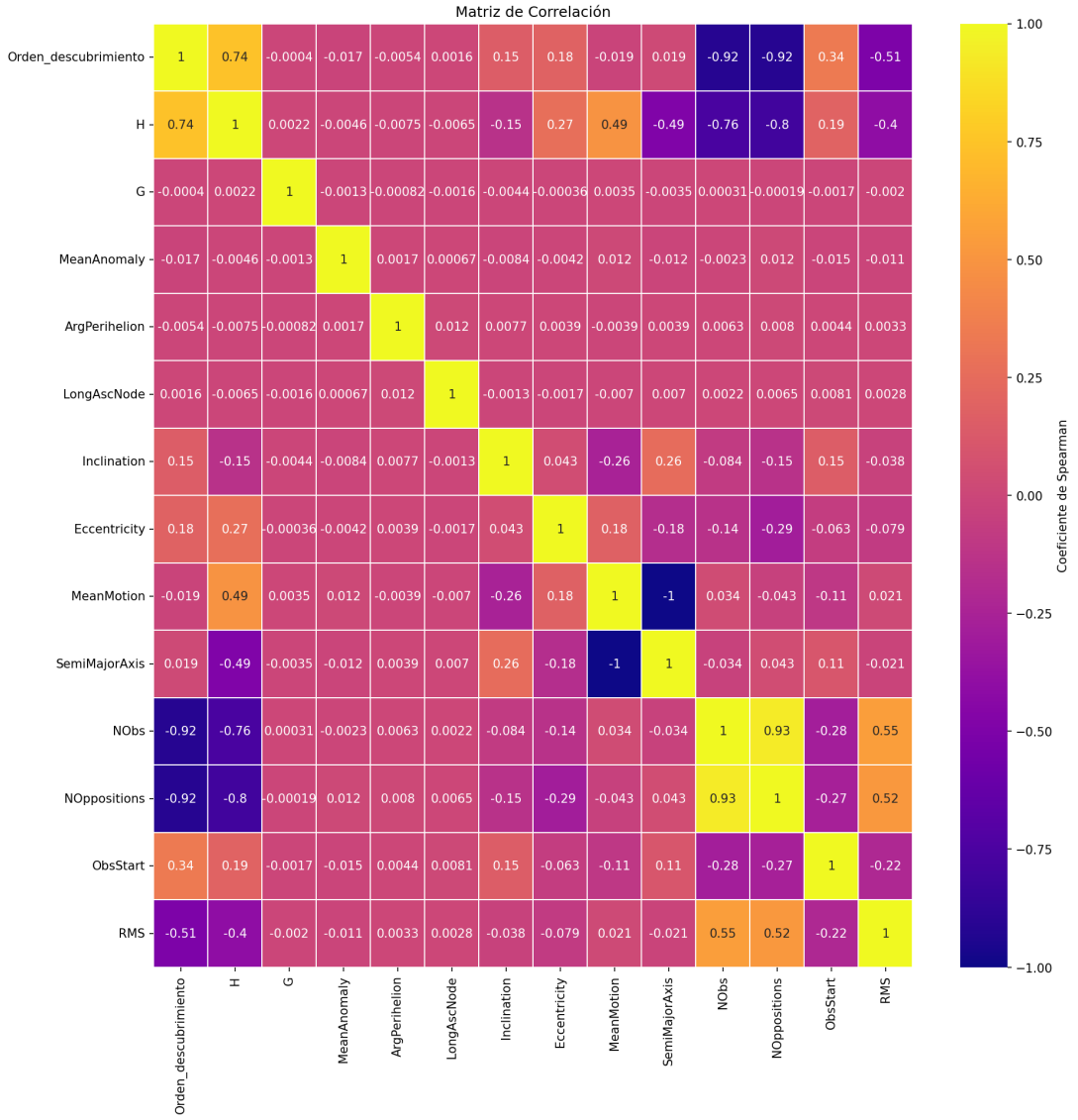
Usa los elementos orbitales (a, e, i) para aproximar las órbitas de objetos menores en 3D.

Inclina las órbitas de objetos menores con respecto al plano eclíptico usando i.

Correlaciones

Se calculó la matriz de correlación de Spearman para las 14 variables numéricas, se uso Spearman debido a la prueba de Shapiro que mostró un valor $p = 0$ para todas las

variables númericas. Se obervó una fuerte correlación positiva de 0,74 entre **H** y **Orden de descubrimiento**, también una fuerte correlación negativa de 0,76 entre **H** y **NObs** y **H** y **NOppositions** de 0,8.Debido a que H muestra una mayor correlación con las otras variables se realizara regresión para predecir usando Random Forest.



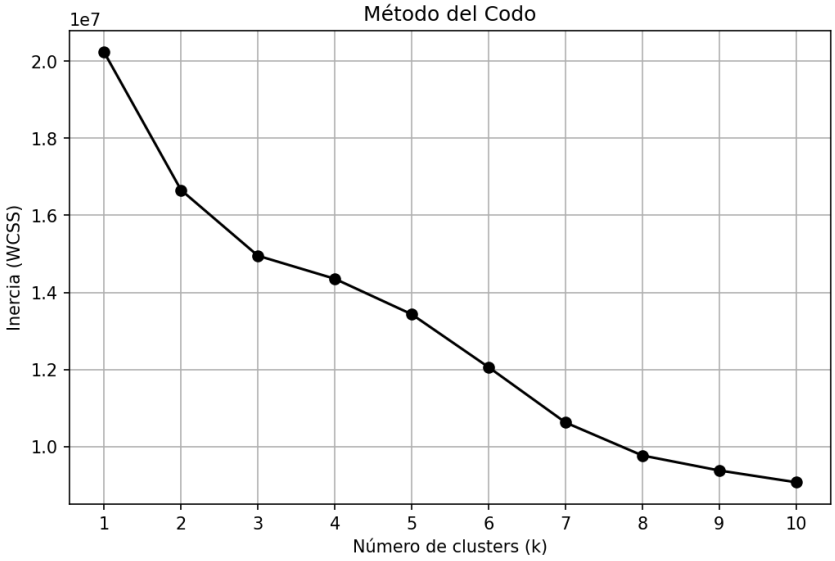
Agrupamiento con K-Means

K-means es un algoritmo de aprendizaje no supervisado utilizado para resolver problemas de clustering (agrupamiento). Su objetivo es dividir un conjunto de datos en k grupos o clusters de tal manera que los elementos dentro de un mismo grupo sean lo más similares posible entre sí, y lo más diferentes posible de los elementos de otros grupos.

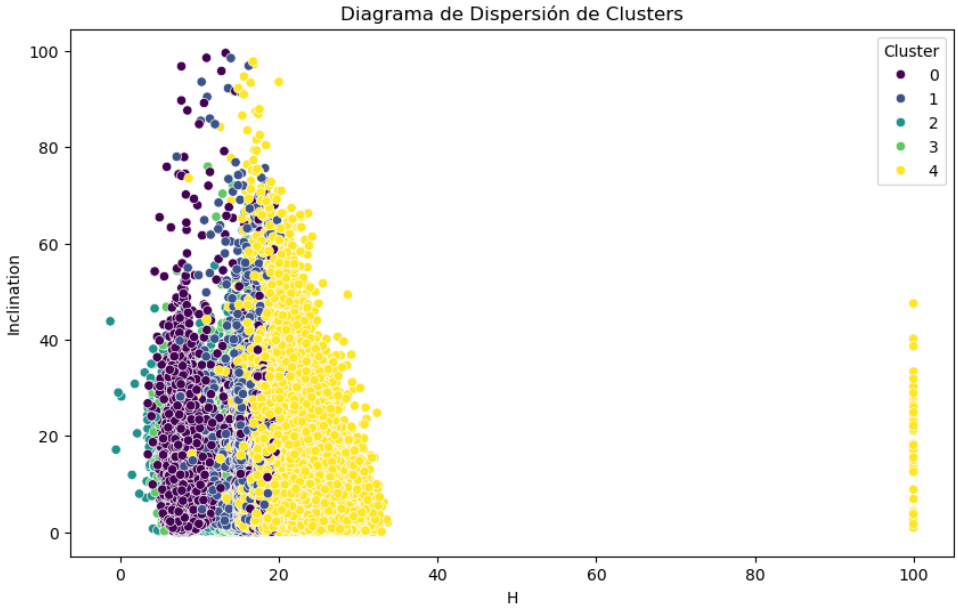
Para este proyecto se aplicó K-Means para identificar familias de objetos celestes, seleccionando el número óptimo de clústeres usando el método del codo, el cual nos ayuda a obtener el mejor valor de k, se obtuvo como valor de agrupamiento $k = 5$. Para realizar el gráfico se ulizaron las paqueterias de

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

```
from sklearn.metrics import silhouette_score
```



La siguiente gráfica muestra los grupos que presentaron similitudes en inclinación y magnitu absoluta.



En el grafico se presenta el diagrama de dispersión resultante del algoritmo **K-Means**, tomando como variables de referencia la magnitud absoluta (H) en el eje horizontal y la inclinación orbital (*Inclination*) en el eje vertical. Cada color representa uno de los cinco clústeres detectados por el modelo. Se observa que la partición generada por K-Means tiende a diferenciar principalmente a los objetos en función de su magnitud absoluta, agrupando a los objetos más brillantes (valores bajos de H) en clústeres distintos de los que contienen objetos más débiles (valores altos de H). La inclinación orbital presenta una distribución más dispersa dentro de cada clúster, aunque se aprecia que la mayoría de los objetos tienen inclinaciones menores a 20°. El clúster identificado con color amarillo agrupa principalmente objetos con valores altos de H, mientras que los clústeres púrpura y azul concentran los objetos más brillantes.

Tercera Ley de Kepler

La tercera ley de Kepler, establece que el cuadrado del período orbital de un planeta es directamente proporcional al cubo de la distancia media de ese planeta al Sol.

$$P^2 = a^3$$

Tercera ley (forma newtoniana):

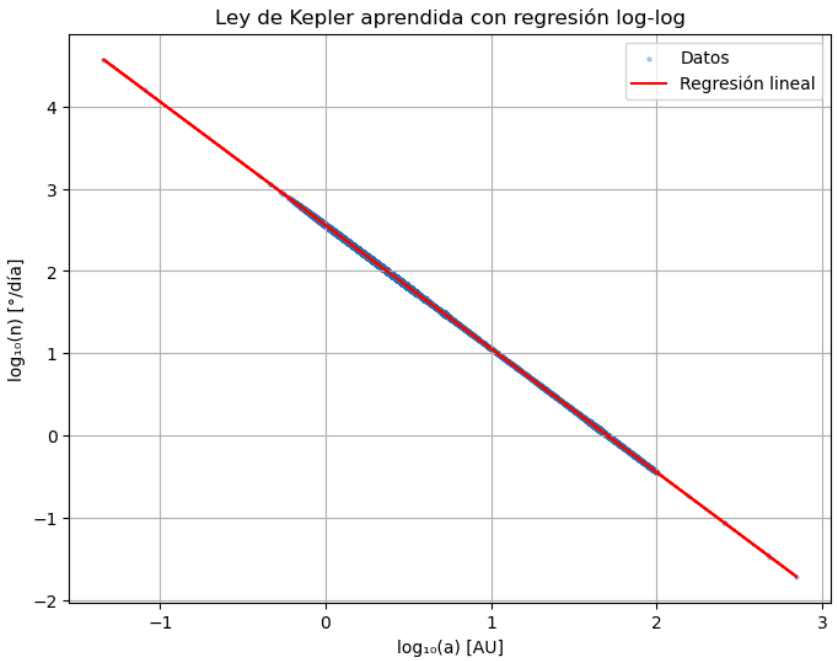
$$n^2 a^3 = \mu, \qquad n = \frac{2\pi}{T}, \quad \mu = G(M+m).$$

Movimiento medio:

$$n = \frac{2\pi}{T}.$$

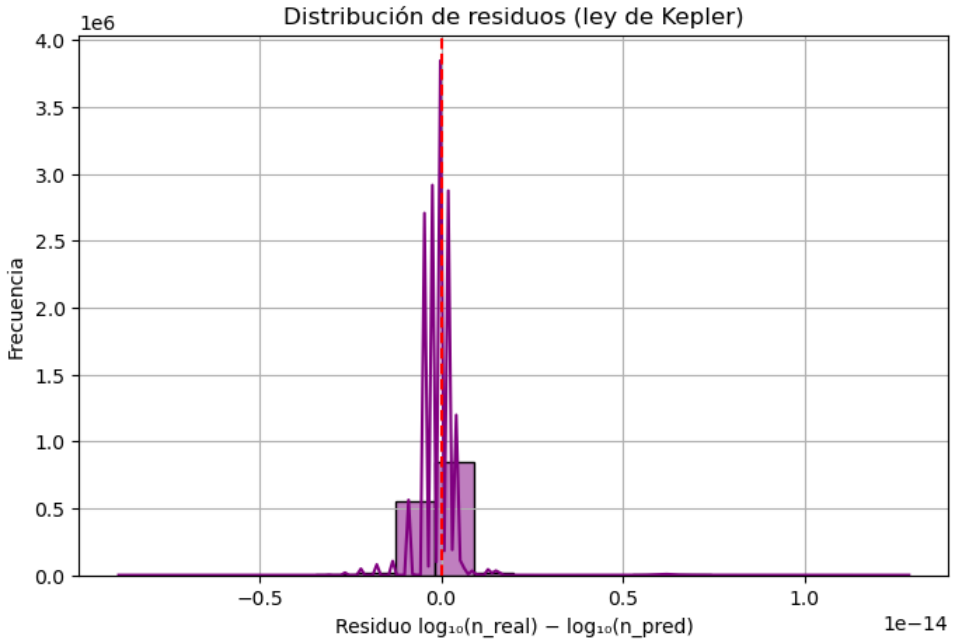
Se evaluó la relación $n \propto a^{-3/2}$ mediante regresión lineal en escala log–log:

$$\log_{10}(n) = m \log_{10}(a) + b$$



El ajuste se dió perfectamente. La gráfica de distribución de residuos muestra que si el residuo es 0, significa que $n_{real} = n_{pred}$.

Si es positivo, el valor real es mayor que el predicho; si es negativo, es menor.



Modelo Random Forest

Para tener una buena predicción de la magnitud absoluta H se tuvo que:

- Calcular la energía de cada asteroide, y guardarla en una columna nueva.

```
mu = 1.32712440018e11  #parámetro gravitacional del Sol km³/s²
Datos['energia_orbital'] = - mu / (2 * Datos['a_km']) #energía orbital km²/s²
```

Lo anterior es para calcular la energía orbital total (energía por unidad de masa) de cada objeto donde tomamos la masa de los objetos igual a 1 para una buena aproximación, usando la fórmula de orbital total:

$E = -\frac{G\mu}{2a}$

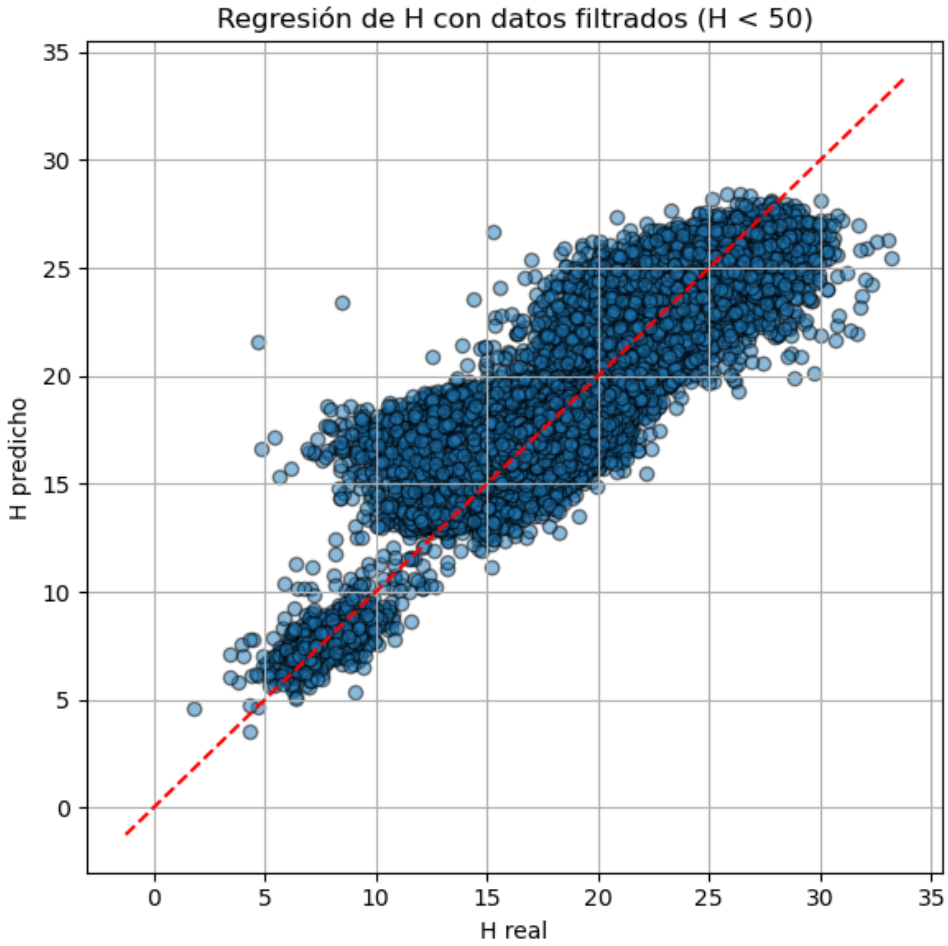
donde:

E = energía orbital total

μ = parámetro gravitacional del Sol

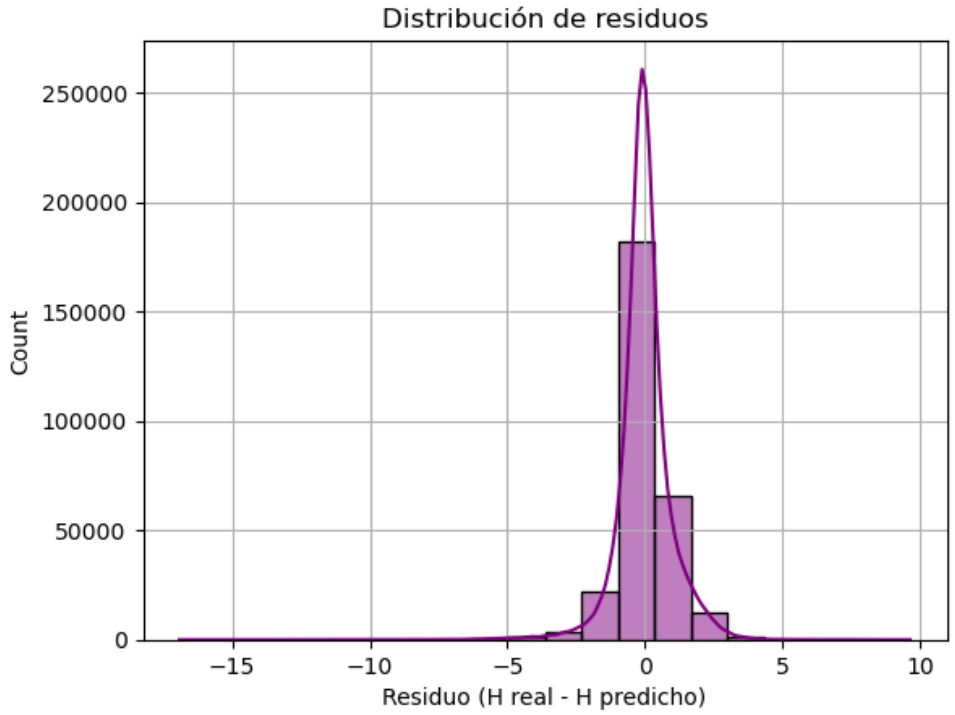
a = semieje mayor de la órbita

Se entrenó un `RandomForestRegressor` para predecir H usando parámetros físicos y orbitales.



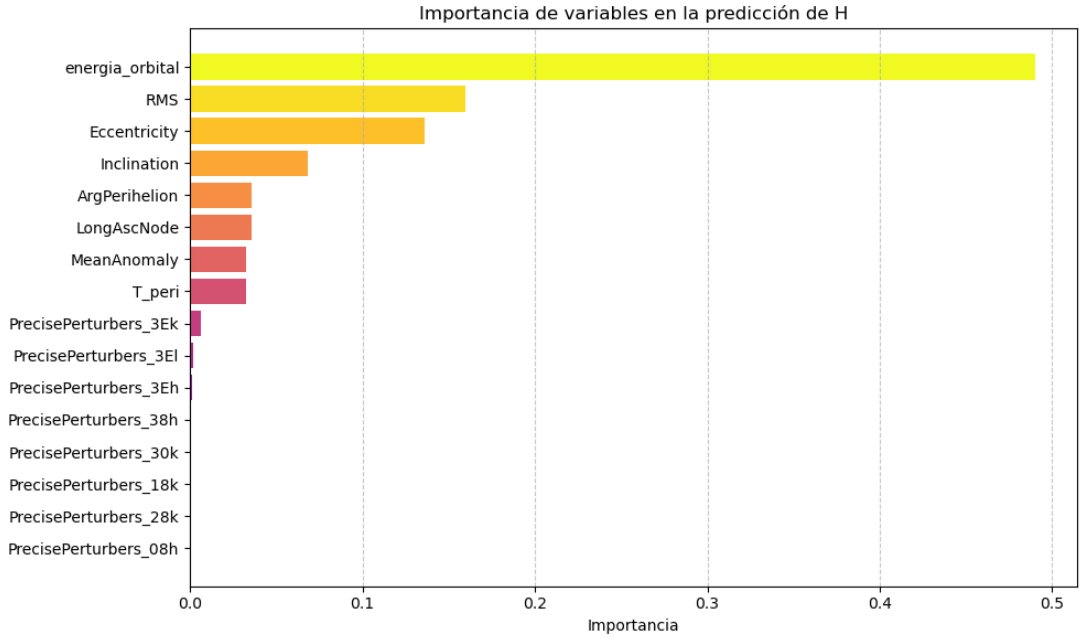
Se obtuvo:

- $R^2 = 0{,}723$
- $RMSE = 0{,}981$



La gráfica muestra que el modelo no tiene gran sesgo, es decir, predice bastante bien a H.

La siguiente gráfica muestra las variables más importantes para la predicción de H.



Conclusiones

- La limpieza y preparación de datos fue esencial para el análisis.
- Las correlaciones observadas concuerdan con principios orbitales conocidos.
- K-Means permitió que se identificaran posibles familias de asteroides.

- El ajuste de la Tercera Ley de Kepler fue consistente con la teoría.
- El modelo Random Forest tuvo un desempeño adecuado para estimar H agregando la energía orbital.

Referencias

- Minor Planet Center: `https://minorplanetcenter.net`
- `https://github.com/Mayte13/Datos/blob/main/The%20Minor%20Planet%20Center.ipynb`
- Documentación de **pandas**, **numpy**,**skyfield.api**, **matplotlib.pyplot**, **seaborn**.