



Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Química

**ESTUDIO DE LAS ESPECIES QUIMICAS PRESENTES
EN EL MEDIO INTERESTELAR MEDIANTE
INTELIGENCIA ARTIFICIAL**

**Autora: Verónica González Alonso
Tutoras: Camilla Calabrese / Celina Bermúdez
Año: 2023/2024**

ÍNDICE

1.	RESUMEN / ABSTRACT	4
2.	INTRODUCCIÓN	5
3.	METODOLOGÍA	9
3.1.	PYTHON	9
3.1.1.	INTRODUCCIÓN A PYTHON	9
3.1.2.	OBJETOS ESCALARES. TIPOS Y OPERACIONES.....	11
3.1.3.	TIPOS DE ESTRUCTURAS. TUPLAS, LISTAS Y DICCIONARIOS	14
3.1.4.	NUMPY.....	16
3.1.5.	PANDAS.....	17
3.2.	MACHINE LEARNING.....	19
3.2.1.	SCIKIT-LEARN.....	24
4.	TRATAMIENTO DE DATOS.....	25
4.1.	PREPARACION DE LA BASE DE DATOS	25
4.1.1.	COLUMNA DE DENSIDAD	25
4.1.2.	CONSTANTES DE ROTACIÓN	26
4.1.3.	FUNCIONES DE PARTICIÓN	28
4.1.4.	FUNCIONES DE BASE	30
4.2.	ANÁLISIS DE DATOS	30
5.	RESULTADOS.....	33
6.	CONCLUSIONES	39
7.	BIBLIOGRAFÍA	40
8.	ANEXOS.....	42

1. RESUMEN / ABSTRACT

El estudio del medio interestelar y la detección de moléculas en las nubes moleculares es una parte muy importante en la astroquímica para conocer y entender el espacio y comprender el origen de la vida.

En este trabajo nos vamos a centrar en el estudio de la nube molecular de Tauro 1 (TMC-1) y para poder explicar la química de las nubes moleculares se va a crear un modelo de aprendizaje automático. Para poder llevar a cabo este primer modelo, se han recopilado datos de moléculas con presencia y tentativa de presencia en dicha nube para la creación de bases de datos con los parámetros astroquímicos de interés de dichas moléculas. Con estas bases de datos se van a intentar predecir qué moléculas son potencialmente presentes en TMC-1 y cuales no lo son.

ABSTRACT

The study of the interstellar médium and the detection of molecules in molecular clouds is a very important part of astrochemistry to know and undertand space and the origin of life.

In this work, we are going to focus on the study of the Taurus 1 molecular cloud (TMC-1) and in the order to explain the chemistry of molecular cloud we are going to create a machine leaning model. In order to carry out this preliminar model, data of molecules with presence and tentative presence in this cloud have been collected for the creation of databases with the astrochemical parameters of interest of these molecules. These databases Will be used to try to predict which molecules are potentially present inTMC-1 and which are not.

2. INTRODUCCIÓN

La astroquímica es la ciencia que tiene por objetivo entender los procesos químicos y físicos que determinan la composición química de gas y las partículas de polvo interestelar. La composición química del gas depende de varios factores: densidad, temperatura, grado de ionización y la radiación presente en el entorno. Se hace uso de las moléculas como herramientas para el estudio del Universo a todos los niveles, desde una galaxia a una atmósfera planetaria.

Tener conocimiento sobre la composición química del gas y las partículas de polvo es de gran importancia ya que las partículas son la materia principal a partir de la cual se forman los planetas. En el interior de las nubes moleculares las partículas de polvo se encuentran recubiertas de una capa de hielo con alto porcentaje de agua y monóxido de carbono donde se da lugar la formación de moléculas orgánicas complejas. Estas especies podrían sobrevivir hasta formar parte de los inicios de los planetas por lo que conocer el destino de las moléculas formadas en estas partículas de polvo tiene gran importancia por la directa relación con el origen de la vida.

El medio interestelar es el espacio que existe entre las estrellas que tiene una gran cantidad de materia difusa que se distribuye por el medio en el que se encuentra. Esta materia se distribuye en diferentes fases según la temperatura en la que se encuentra desde zonas extremadamente frías próximas al cero absoluto hasta donde tiene lugar la formación de las estrellas hasta zonas con elevadísimas temperaturas donde quedan restos de supernovas. Según las condiciones de densidad y temperatura, la materia se puede encontrar en distintos estados que no son permanentes ya que esta materia se encuentra en continua circulación entre las distintas fases del medio interestelar. De tal manera que uno de los objetivos principales de la astroquímica es buscar una explicación sobre la naturaleza y abundancia de las moléculas que se encuentran en el medio interestelar al igual que a nivel experimental se establezcan los procesos por los cuales se identifiquen los procesos que tengan una mayor influencia en la abundancia de las especies observadas.

Las estrellas que se observan en el espacio son estrellas que se han formado hace mucho tiempo, pero hay zonas del espacio en las que se están formando, dichas zonas son nubes de gran tamaño de gas interestelar que son consideradas acumulaciones de materia difusa procedente de anteriores estrellas que al morir expulsaron hacia el exterior. A estas nubes se las conoce como nubes oscuras frías o nubes moleculares.¹

Las nubes oscuras frías o nubes moleculares son miembros cercanos de la fase más densa (con una densidad de 10^4 cm^{-3} que se correspondería aproximadamente a 10.000 moléculas por centímetro cúbico) y fría (con temperaturas que tienen un amplio rango desde los 10 a 300K)

del medio interestelar. Estas nubes son representativas de los lugares más accesibles donde nacen las estrellas actualmente debido a las condiciones en las que se encuentran. Están formadas de material molecular muy diverso con H_2 como constituyente dominante y se denominan con estos nombres debido a la composición molecular y a su aspecto opaco que se debe a la presencia de sólidos diminutos (granos de polvo) que absorben la luz óptica de las estrellas reduciendo los efectos de calentamiento de la radiación externa dando lugar a temperaturas muy poco elevadas por encima de los 2.7K (entre 10 y 20K aproximadamente). Si la nube molecular tiene zonas donde se pueden formar estrellas, se denominan viveros estelares.²

Una de las principales teorías de formación de estrellas es la teoría del colapso gravitacional. Esta teoría explica que los núcleos son regiones donde el polvo interestelar y el gas se acumulan debido a la atracción gravitatoria. La gran densidad de materia existente en los núcleos hace que se necesite una mayor fuerza gravitatoria, de tal manera, que, si ésta última es lo suficientemente fuerte para que el gas y polvo colapsen, se dará el inicio de la formación estelar (ver Ilustración 1).³

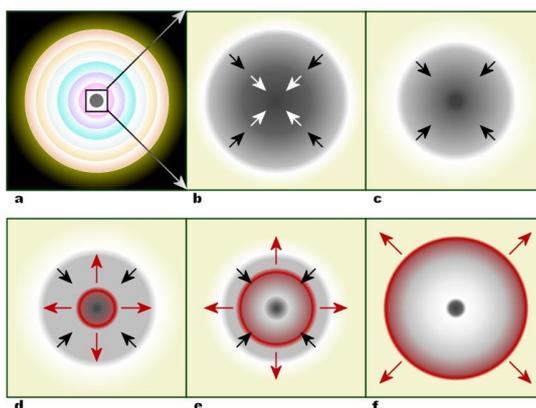


Ilustración 1. Esquema colapso gravitacional⁴

En las últimas décadas, el esfuerzo económico y el desarrollo instrumental han permitido avances significativos tanto en el estudio de las nubes moleculares como en la comprensión de la evolución de las moléculas orgánicas. La identificación de nuevas moléculas en el espacio interestelar es una de las ramas más dinámicas y con mayor impulso en investigación. A fecha de mayo 2024 se han catalogado 310 moléculas en la sección de moléculas encontradas en los espacios interestelares o circunestelares⁵. Estas moléculas desempeñan un papel crucial en la química interestelar, ya que permiten trazar los caminos evolutivos necesarios para comprender el origen de la vida.

La nube molecular de Tauro (TMC-1) es una de las regiones del espacio que mejor se conocen a nivel astroquímico y se considera que es un vivero estelar que se encuentra situado en la constelación de Tauro a 140 parsec (430 años luz) de nuestro planeta Tierra (ver Ilustración

2). La química existente en TMC-1 es de gran interés debido a la formación de especies tanto neutras como radicales, aniones y cationes e incluso moléculas con heteroátomos y ciclos. De esta forma, se han detectado un gran número de moléculas en esta nube, así como se especula de la existencia de muchas otras que facilitarán el estudio de la química en el Universo.

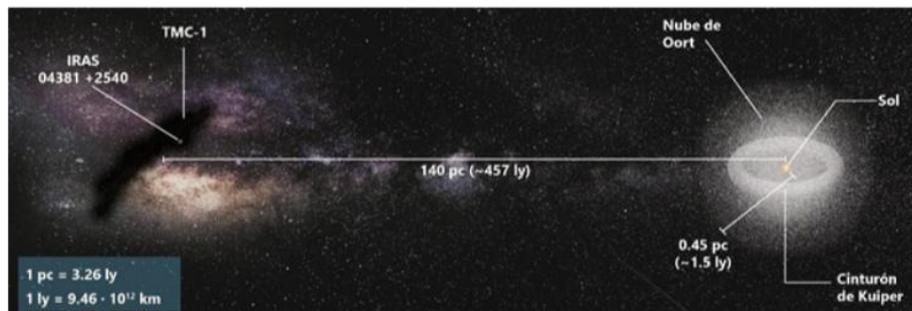


Ilustración 2. Distancia de la nube molecular Tauro 1 (TMC-1) respecto a nuestro Sol³

Debido a la baja temperatura del espacio las moléculas presentes en este medio tienen una energía muy baja pero sí la suficiente como para poder rotar, de tal manera que la espectroscopía rotacional es la técnica de alta resolución que se encarga de la identificación de las especies químicas ya que mide las energías de las transiciones que tienen lugar entre estados rotacionales cuantizados de las moléculas en fase gas. Cada molécula tiene un espectro rotacional único y es el medio por el cual pueden ser estudiadas.⁶

De hecho, hoy en día, la principal técnica para la detección de especies moleculares en el espacio interestelar es a través de observaciones astronómicas en las regiones de frecuencias centimétricas, milimétricas y submilimétricas, que permiten observar transiciones rotacionales.

Las 310 moléculas detectadas se han identificado de manera concluyente mediante el uso de esta técnica espectroscópica, comparando los espectros obtenidos en los laboratorios con las observaciones astronómicas. La identificación de nuevas moléculas en el espacio interestelar representa uno de los retos más fascinantes de la astroquímica actual. Las posibilidades de éxito aumentan con la obtención de más datos de laboratorio y el análisis exhaustivo de las moléculas ya detectadas, lo que permite la identificación de patrones que puedan indicar cuál será la próxima molécula a buscar y encontrar. Con este propósito, es importante la organización de todos los datos ya publicadas para, posteriormente, intentar implementar nuevas técnicas de *machine learning* que van a permitir el desarrollo de nuevos algoritmos capaces de identificar las variables más significativas para la detección de moléculas haciendo uso de los datos de laboratorio disponibles. De igual manera, estos algoritmos pueden ayudar a predecir moléculas que aún no se han estudiado de forma experimental pero que podrían ser detectadas en el espacio interestelar.

Para alcanzar estos objetivos, en este TFG se ha realizado una recopilación de las moléculas presentes y con sospecha de presencia en la nube interestelar de Tauro 1 (TMC-1), creando una base de datos con la que se va a trabajar posteriormente.

3. METODOLOGÍA

3.1. PYTHON

3.1.1. INTRODUCCIÓN A PYTHON

Para poder abordar la realización de este trabajo se ha tenido que aprender lenguaje de programación. El que se ha usado principalmente ha sido Python: se trata de un lenguaje de gran versatilidad que se usa de forma habitual para la creación de sitios web y software, análisis y visualización de datos.⁷

Se han seguido varios cursos para el aprendizaje de este lenguaje a través de la organización edX.⁸ El principal objetivo de estos cursos es que, aparte de aprender dicho lenguaje, se empiece a pensar de forma computacional y algorítmica, de tal forma que cuando se presente un problema se sea capaz a indicar a un ordenador cómo resolverlo, qué etapas hay que seguir para que sea la máquina quien nos dé un resultado, ya que un ordenador tiene dos principales funciones, hacer cálculos y recordar cosas, donde la rapidez con la que nos va a proporcionar un resultado va a ser mucho mayor que si se realiza de forma manual.

Al ir avanzando con este tipo de pensamiento, se va a poder hacer una clasificación de las distintas clases de algoritmos que se van a usar en distintos problemas para buscar y clasificar contenido. Para empezar, un ordenador para resolver una incógnita debe tener un conocimiento en el cual va a basar la respuesta. Se van a poder distinguir dos tipos de conocimiento:

- **Conocimiento declarativo:** Se refiere al enfoque que se encarga de describir qué se debe hacer sin entrar en especificaciones de cómo se tiene que hacer. Este tipo de conocimiento se centra en las relaciones y restricciones entre los datos.
- **Conocimiento imperativo:** Se define como el conocimiento sobre cómo se lleva a cabo un proceso de forma efectiva; cómo conocer u obtener información, de tal forma que se centra en la elaboración de una secuencia de instrucciones detalladas para realizar una tarea.

El conocimiento que más interesa a la hora de programar es el conocimiento imperativo ya que para poder crear un código hay que indicar la secuencia de operaciones que hay que realizar para poder resolver un problema. Si se introduce esta lista de pasos en un ordenador es para pasar de tener la descripción de un proceso a un conjunto específico de declaraciones que la máquina va a evaluar, de forma que nos va a proporcionar un conjunto de operaciones primitivas. A esto se le denomina lenguaje de programación.⁸

Las operaciones primitivas se unen mediante un concepto conocido como “expresiones” y se definen como una combinación de valores, variables, operadores y llamadas a funciones que

se van a evaluar para dar lugar a un resultado. De esta forma el lenguaje de programación es un conjunto de primitivos, una forma de poder combinarlos para crear nuevas declaraciones y de igual manera, una forma con la que se va a poder tomar una expresión compleja para poder tratarla como primitivo. Los primitivos en el caso de la programación van a ser números, cadenas o secuencias de caracteres y operaciones simples (suma, resta, comparaciones...). Cuando se combinen tienen que hacerlo de forma que tenga sentido a nivel sintáctico. Si una expresión es correcta a nivel sintáctico se puede valorar la semántica, si es una semántica estática o semántica completa, siendo la primera la que hace referencia a cómo se determina el significado que tiene un programa, qué cadenas tienen realmente un significado, y la segunda es aquella que hace referencia a que todas las expresiones tienen un significado que se encuentra bien definido, sin ambigüedades.

Una expresión que sea sintácticamente correcta y con una semántica estática apropiada va a tener significado siempre pero no tiene por qué ser el que uno espera, por eso es importante revisar el código para evitar tres tipos de errores:

- **Error sintáctico:** Este tipo de errores se presentan cuando en un código no se ha escrito bien una expresión en el lenguaje de programación que se esté usando. Generalmente, son fáciles de detectar y fáciles de corregir. Suelen estar relacionados con la falta de algún punto o coma al final de una instrucción, haber usado mal los paréntesis, corchetes o llaves...
- **Errores relacionados con la semántica estática:** Hacen referencia a todos aquellos errores que se centran en el significado del código. Aparecen cuando hay problemas con el significado lógico del problema. Se pueden considerar errores de este tipo a todas aquellas expresiones donde se intenten realizar operaciones con datos que no sean compatibles entre sí, usar identificadores que no se han declarado previamente...Cuesta más identificarlos y solucionarlos que a los errores sintácticos.
- Se puede tener una problemática mucho más grave que es que no haya ningún tipo de error semántico pero que el resultado que se obtiene no es el realmente esperado, esto puede dar lugar a una serie de consecuencias como que el programa falle y deje de ejecutarse, o que por el contrario no deje de ejecutarse hasta que no se pare el programa de forma manual ya que no genera ningún resultado. Aunque el peor de los casos, es que el programa de una solución y que ésta sea diferente a la que se espera de tal manera que se utilice para sacar conclusiones erróneas.

De esta manera, si se juntan todas las expresiones de una forma correcta, sin ningún tipo de error, se va a conseguir una secuencia de definiciones (formas que se tiene de asignar nombres a valores, de crear procedimientos que se van a tratar como primitivos) y comandos

(declaraciones que instruyen al interprete a hacer algo, como operaciones aritméticas) que en su conjunto dan lugar al lenguaje de programación Python, un lenguaje de programación libre.⁸

3.1.2. OBJETOS ESCALARES. TIPOS Y OPERACIONES.

Se ha estado hablando de primitivos, también llamados objetos, y tienen por función representar los datos. Cada objeto está tipificado y se encarga de definir qué es cada expresión usada. Esta tipificación tiene una gran importancia ya que gracias a ella los programas saben si pueden actuar o no sobre ella (si se espera un número y se tiene una cadena, el programa no va a realizar ningún tipo de cálculo sobre ello). Los objetos se pueden clasificar como escalares, no se van a poder dividir, o no escalares, que en este caso van a tener una estructura interna a la cual se va a poder acceder y extraer distintas partes (listas, diccionarios, tablas...)⁷.

Los principales objetos escalares que se clasifican en:

- **Números enteros:** Se van a representar por el acrónimo *int*.
- **Números decimales:** En programación, se va a nombrar como *float*.
(ver Ilustración 3)
- **Booleano:** Hace referencia a un tipo de dato que únicamente va a tener dos valores: verdadero o falso. Se suelen usar como herramienta para representar condiciones lógicas y se suelen usar en estructuras de control, como pueden ser instrucciones condicionales (*if, else*), bucles (*for, while*) además de en otras condiciones lógicas que se ejemplificarán más adelante.
- **No tipo (*NoneType*):** Representa la ausencia o falta de un valor o la definición de un objeto. Generalmente, se usa como indicativo de la ausencia de un valor que sea válido

(ver Ilustración 4).

```
In [9]: 3+7
Out[9]: 10

In [10]: 3.0+7
Out[10]: 10.0

In [11]: 3+5.0
Out[11]: 8.0

In [12]: round(2.6)
Out[12]: 3

In [13]: int(5.2)
Out[13]: 5

In [14]: float(2)
Out[14]: 2.0
```

Ilustración 3. Ejemplos de operaciones con números enteros y decimales

```
In [22]: 3 > 4
Out[22]: False

In [23]: 2 + 2 == 4
Out[23]: True

In [24]: b = 10

In [25]: c = b > 9

In [26]: c
Out[26]: True

In [27]: d = 8

In [28]: e
Traceback (most recent call last):

  Cell In[28], line 1
    e
NameError: name 'e' is not defined
```

Ilustración 4. Ejemplos de booleanos y NonType

A partir de este punto se van a poder realizar combinaciones entre objetos y operadores para dar lugar a expresiones. Estas combinaciones van a tener una estructura simple de un objeto seguido de un operador y luego otro objeto de tal manera que se considera una expresión sintácticamente válida.

Existe una herramienta por la cual se va a poder dar nombres a distintos elementos de tal manera que se va a poder reusar ese nombre para recuperar el valor o elemento asociado a él, dicha herramienta es la abstracción. En programación, generalmente se representa mediante el símbolo de igualdad. El nombre que se va a asociar al elemento recibe la denominación de variable, mientras que dicho elemento se va a conocer como valor. Este valor se va a guardar en la memoria del ordenador y se va a poder recuperar en el momento en el que se llame a esa variable.

El uso de esta herramienta se debe fundamentalmente a dos motivos: nombrar un valor de cierta manera facilita la recuperación del dato en vez de tener que realizar el cálculo de nuevo, sobre todo en casos donde se tiene una expresión aritmética compleja. A su vez el uso de esto va a hacer que el código sea mucho más entendible ya que se deben usar nombres que tengan que ver o que den una pista sobre lo que se está trabajando, deben ser nombres informativos.

Esta asignación no tiene por qué ser definitiva, se van a poder hacer renombramientos, pero si se hace hay que tener en cuenta que el primer nombre usado se va a perder.

Poniendo en conjunto la información que se ha visto hasta ahora se va a poder tener un código de mayor complejidad con problemas lógicos donde hay que seguir una serie de órdenes y especificaciones para poder resolverlo, como puede ser la sangría, ya que es un elemento importante que aparte de ayudar visualmente con el código se encarga de indicar la jerarquía de las estructuras de control (ver Ilustraciones 5 y 6).⁸

```

1 x = float (input ("Enter a number for x: "))
2 y = float (input ("Enter a number for y: "))
3 if x == y:
4     print ("x and y are equal.")
5     if y != 0:
6         print ("therefore, x / y is" , x/y)
7 elif x < y:
8     print ("x is smaller")
9 else:
10    print ("y is smaller")
11

```

Ilustración 5. Ejemplo código

```

In [9]: runfile('C:/Users/gonza/.spyder-py3/untitled0.py', wdir='C:/Users/gonza/.spyder-py3')
Enter a number for x: 4
Enter a number for y: 7
x is smaller

In [10]: runfile('C:/Users/gonza/.spyder-py3/untitled0.py', wdir='C:/Users/gonza/.spyder-py3')
Enter a number for x: 9
Enter a number for y: 2
y is smaller

In [11]: runfile('C:/Users/gonza/.spyder-py3/untitled0.py', wdir='C:/Users/gonza/.spyder-py3')
Enter a number for x: 5
Enter a number for y: 5
x and y are equal
therefore, x / y is 1.0

```

Ilustración 6. Resultado ejemplo código

Otra de las estructuras que se usan en programación son los bucles, estos se consideran como un tipo de configuración que permiten ejecutar un bloque de un código de forma repetida, siempre que se cumpla una condición específica. La principal función de los bucles es la automatización de tareas que son repetitivas de tal forma que se pueden procesar un conjunto de datos de una forma rápida y eficiente. Los bucles más usados son el bucle “while” y el bucle “for” (ver Ilustración 7). El primero se encarga de ejecutar un bloque completo siempre que se cumpla una condición concreta, dicha condición se evalúa antes de cada iteración; mientras que el segundo se usa más cuando hay que iterar sobre una secuencia concreta y ejecutar un bloque de código específico para cada elemento de la secuencia.

```

In [32]: n = 0

In [33]: while (n < 5):
...:     print (n)
...:     n = n+1
...:
0
1
2
3
4

In [34]: for n in range (5):
...:     print (n)
...:
0
1
2
3
4

```

Ilustración 7. Ejemplos de bucles con "while" y "for"

Como se ve, los programas de bifurcación o programas de decisión se usan de tal forma que se realicen diferentes acciones según el valor que se tenga de una determinada condición. Es muy común en este tipo de programas el uso de flujos de control que hacen uso de declaraciones como “if” “while” “else” ...⁷

3.1.3. TIPOS DE ESTRUCTURAS. TUPLAS, LISTAS Y DICIONARIOS

Con todo lo que se ha visto anteriormente se consigue una gran variedad de posibilidades de programación, pero algo que realmente interesa es ser capaz de estudiar un número elevado de datos. De igual manera se van a poder agrupar para poder tratarlos como primitivos de tal forma que va a tener una gran relevancia. Esto lleva a hablar de tres estructuras nuevas de datos que son las listas, las tuplas y los diccionarios.

Una tupla se define como una estructura de datos que se usa para guardar de forma ordenada una colección de datos. Ordenada no hace referencia a que estos elementos sigan una secuencia determinada de menor a mayor, sino que la secuencia como tal tiene un orden de tal forma que se puede acceder a las distintas partes de la secuencia para poder registrarlos y formar un índice con ellos. Las tuplas se caracterizan porque no son inmutables, es decir, una vez que se han creado no se van a poder agregar, eliminar ni modificar los datos más adelante en dicha tupla, aunque si se puede crear otra que contenga todos los datos modificados. Una tupla se representa entre paréntesis (ver Ilustración 8).

```
In [3]: tuple = (2, "one", 3)
In [4]: tuple
Out[4]: (2, 'one', 3)
In [5]: tuple [0]
Out[5]: 2
In [6]: tuple [2] = 5
Traceback (most recent call last):
  Cell In[6], line 1
    tuple [2] = 5
TypeError: 'tuple' object does not support item assignment
```

Ilustración 8. Ejemplo tupla

Por otro lado, se tienen las listas, estas son muy similares a las tuplas solo que se diferencian de ellas en un aspecto. De igual forma, se definen como una estructura de datos que se usa para poder almacenar un conjunto de elementos de forma ordenada.⁷ La diferencia con las tuplas es que las listas si son mutables por lo que se van a poder modificar tras su creación. Una lista se representa mediante los corchetes de tal forma que así se puede diferenciar una lista de una tupla (ver Ilustración 9).

```

In [40]: list = [1, 3, 5, 7]
In [41]: print(list[2])
5
In [42]: list.append(9)
In [43]: print (list)
[1, 3, 5, 7, 9]
In [44]: list.remove (1)
In [45]: print (list)
[3, 5, 7, 9]

```

Ilustración 9. Ejemplo lista

Por último, se tienen los diccionarios, la última de estas tres estructuras que se usa como un diccionario convencional, pero en programación, de tal manera que sirve para guardar valores en él. En Python se almacenan pares de datos juntos, una clave y un valor, donde la clave hace referencia a lo que se tiene que buscar. Sobre esta estructura se va a poder realizar variedad de modificaciones: añadir entradas, borrarlas, hacer pruebas y test sobre ellas... El diccionario se representa mediante un conjunto de elementos que se encuentran entre corchetes (ver Ilustraciones 10, 11 y 12).

```

notas= {'Ana': 'Notable', 'Lucia': 'Sobresaliente', 'Sara': 'Suspenso', 'Henar': 'Notable' }

```

Ilustración 10. Ejemplo diccionario

```

In [3]: notas
Out[3]:
{'Ana': 'Notable',
 'Lucia': 'Sobresaliente',
 'Sara': 'Suspenso',
 'Henar': 'Notable'}

In [4]: notas ['Henar']
Out[4]: 'Notable'

In [5]: notas ['Daniel'] = 'Aprobado'

In [6]: notas ['Daniel']
Out[6]: 'Aprobado'

In [7]: 'Lucia' in notas
Out[7]: True

```

Ilustración 11. Ejemplo operaciones en diccionarios

```
In [8]: notas ['Alberto']
Traceback (most recent call last):

  Cell In[8], line 1
    notas ['Alberto']
KeyError: 'Alberto'

In [9]: del (notas['Ana'])

In [10]: notas
Out[10]: {'Lucia': 'Sobresaliente',
'Sara': 'Suspenso',
'Henar': 'Notable',
'Daniel': 'Aprobado'}

In [11]: notas.keys()
Out[11]: dict_keys(['Lucia', 'Sara', 'Henar', 'Daniel'])

In [12]: notas.values()
Out[12]: dict_values(['Sobresaliente', 'Suspenso', 'Notable', 'Aprobado'])
```

Ilustración 12. Ejemplo operaciones en diccionarios

Se puede ver que esta herramienta es muy útil en las situaciones en las que se tenga que almacenar datos que se encuentren asociados entre sí, convirtiéndose en una estructura fundamental en programación.⁷

3.1.4. NUMPY

Como se ha dicho anteriormente, Python es un lenguaje de programación libre y uno de los módulos adicionales más usados es Numpy (*Numeric Python*) que proporciona rutinas básicas que facilitan la manipulación de grandes matrices de un número de datos determinado. Facilita dos objetos fundamentales: un objeto array N-dimensional (*ndarray*) y un objeto función universal (*ufunc*).⁹

Un objeto array N-dimensional se considera una colección homogénea de datos indexados mediante un número determinado de números enteros N.⁹ Se puede definir mediante dos datos esenciales: la forma de la matriz y el tipo de elementos que componen dicha matriz. La forma de la matriz es una tupla de N enteros (uno por cada dimensión) que se encarga de proporcionar información sobre cuánto puede variar dicho índice a lo largo de esa dimensión. Por otra parte, se tiene la información importante que describe un array que viene definido por el elemento que forma parte de la matriz (ver Ilustración 13).

Cada *ndarray* se considera una colección homogénea del mismo tipo de datos donde cada elemento ocupa el mismo tamaño de bloque de memoria, y cada bloque de memoria en el array se interpreta de la misma manera, es decir, si se usan arreglos tipo *object*, se va a poder tener arrays heterogéneos, pero el sistema va a englobar cada elemento del array en una misma referencia a un objeto en Python.

```

In [19]: data_list = [
        [0,1,2,3,4],
        [10,11,12,13,14],
        [20,21,22,23,24],
        [30,31,32,33,34],
        [40,41,42,43,44],
        ]
        data_np = np.array(data_list)
        data_np

Out[19]: array([[ 0,  1,  2,  3,  4],
               [10, 11, 12, 13, 14],
               [20, 21, 22, 23, 24],
               [30, 31, 32, 33, 34],
               [40, 41, 42, 43, 44]])

In [20]: data_list[2][1:4]

Out[20]: [21, 22, 23]

In [21]: data_np[2, 1:4]

Out[21]: array([21, 22, 23])

In [24]: selection = []
        for index, row in enumerate(data_list):
            if index > 0:
                selection.append(row[4])
        selection

Out[24]: [14, 24, 34, 44]

In [25]: data_np[1:, 4]

Out[25]: array([14, 24, 34, 44])

```

Ilustración 13. Ejemplo Numpy

Se ha visto que Numpy se considera una herramienta con gran versatilidad y poderosa para el cálculo numérico en Python, aunque tiene ciertos límites que se deben tener en cuenta:

- **Consumo de memoria:** Si se llega a trabajar con un conjunto de datos muy elevado, a pesar de que sea una herramienta muy eficiente en lo que a uso de memoria para arreglos grandes se refiere, puede ser limitante sobre todo en entornos restrictivos de memoria.
- **Tipo de datos limitados:** El tipo de datos numéricos que admite Numpy se considera muy amplio y también puede operar con datos heterogéneos, pero cuando se usa el arreglo de tipo “object” se puede ver limitado frente a otros sistemas de tratamiento de datos más complejos.
- **Capacidad de cálculo simbólico:** El principal objetivo de Numpy son los cálculos numéricos y no ofrece facilidades a la hora del cálculo simbólico.

3.1.5. PANDAS

Todas estas problemáticas se pueden resolver con Pandas, una herramienta de análisis y manipulación de datos de código abierto rápida, potente, flexible y de fácil uso. Se construye sobre el lenguaje de programación de Python. Esta herramienta proporciona estructuras de datos flexibles y potentes que se utilizan para trabajar con datos tabulares y marcados de una forma intuitiva.

Los dos tipos de estructuras de datos básicas de manejo de datos en Panda son:

- **Series:** Una estructura de datos (un array) unidimensional que tiene cierta similitud a un arreglo unidimensional de NumPy solo que se va a tener un índice que se encuentra asociado a cada uno de los elementos presentes (ver Ilustración 14). Las

series de Pandas proporcionan una variedad de métodos y funcionalidades para poder tanto manipular como analizar datos de una manera eficiente.

```
In [26]: import pandas as pd
s = pd.Series (data=[1,2, 'three'], index=['id1', 'id2', 'id3'])
s = pd.Series ({'id1': 1, 'id2': 2, 'id3': 'three'})
s

Out[26]: id1      1
         id2      2
         id3     three
         dtype: object
```

Ilustración 14. Ejemplo Pandas - Series

- **DataFrame:** Este tipo de estructuras de datos bidimensional que contiene datos y es equivalente a un array bidimensional de Numpy. Se puede definir como una tabla de filas y columnas, donde cada columna puede ser de distintos tipos de datos (ver Ilustración 15). Los DataFrames se reconocen gracias a que tienen etiquetas asociadas tanto para las columnas como para las filas, proporcionando una mayor facilidad para el acceso y manipulación de los datos.

```
In [27]: import pandas as pd
df = pd.DataFrame(
[[4, 7, 10],
 [5, 8, 11],
 [6, 9, 12]],
index=['fila_1', 'fila_2', 'fila_3'],
columns=["col_a", "col_b", "col_c"])
df

Out[27]:
```

	col_a	col_b	col_c
fila_1	4	7	10
fila_2	5	8	11
fila_3	6	9	12

Ilustración 15. Ejemplo Pandas – DataFrame

Los DataFrames se pueden considerar como un diccionario de Series donde cada Serie representa cada una de las columnas en el DataFrame con un nombre determinado (lo que se ha visto anteriormente como la clave del diccionario). Por otra parte, las filas del DataFrame son las que se encargan de representar las observaciones ¹⁰ (ver Ilustración 16).

Una de las principales ventajas de tratar los DataFrames como diccionarios es la fácil manipulación de los datos porque se pueden aplicar operaciones a algunas columnas en concreto o a todo el DataFrame según se necesite.

```
In [32]: import pandas as pd
serie1 = pd.Series(data=[1, 2, 3], index=["fila_1", "fila_2", "fila_3"])
serie2 = pd.Series(data=[4, 5, 6], index=["fila_1", "fila_2", "fila_3"])
d = {
    "serie1": serie1,
    "serie2": serie2,
}
pd.DataFrame(d)
```

```
Out[32]:
```

	serie1	serie2
fila_1	1	4
fila_2	2	5
fila_3	3	6

Ilustración 16. Ejemplo DataFrame como diccionario de Series

Otro de los usos principales de la herramienta de Pandas es el Análisis Exploratorio de Datos (EDA). Se trata de una etapa de gran peso ya que se centra en explorar y comprender los datos en un análisis de estos antes de que se realice cualquier cambio. Con este tipo de análisis se va a poder obtener información del DataFrame, visualizar, agrupar y resumir datos, así como tener un manejo de valores que no se encuentren en la base de datos¹¹ (ver Ilustración 17). El conjunto de todos estos procesos va a hacer que se obtenga una comprensión completa de los datos y se pueda extraer la información deseada para futuros estudios.

```
In [33]: curl -s -L https://1ew.ag/countries-dataset > countries.csv
head -n 3 countries.csv

Cell In[33], line 1
curl -s -L https://1ew.ag/countries-dataset > countries.csv
^
SyntaxError: invalid syntax
```

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: df = pd.read_csv("countries1.csv", decimal=',')
```

```
In [5]: df.head(-5)
```

```
Out[5]:
```

	Country	Region	Population	Area (sq. mi.)	Pop. Density (per sq. mi.)	Coastline (coast/area ratio)	Net migration	Infant mortality (per 1000 births)	GDP (\$ per capita)	Literacy (%)	Phones (per 1000)	Arable (%)	Crops (%)	Other (%)	Climate	Birthrate
0	Afghanistan	ASIA (EX. NEAR EAST)	31056997	647500	48.0	0.00	23.06	163.07	700.0	36.0	3.2	12.13	0.22	87.65	1.0	46.1
1	Albania	EASTERN EUROPE	3581655	28748	124.6	1.26	-4.93	21.52	4500.0	86.5	71.2	21.09	4.42	74.49	3.0	15.1
2	Algeria	NORTHERN AFRICA	32930091	2381740	13.8	0.04	-0.39	31.00	6000.0	70.0	78.1	3.22	0.25	96.53	1.0	17.1
3	American Samoa	OCEANIA	57794	199	290.4	58.29	-20.71	9.27	8000.0	97.0	259.5	10.00	15.00	75.00	2.0	22.1
4	Andorra	WESTERN EUROPE	71201	468	152.1	0.00	6.60	4.05	19000.0	100.0	497.2	2.22	0.00	97.78	3.0	8.1

Ilustración 17. Ejemplo Análisis Exploratorio de Datos

3.2. MACHINE LEARNING

Con todo este previo conocimiento del tratamiento de datos se va a poder introducir la herramienta de *Machine Learning*: se puede definir como el área de la ciencia computacional que se encarga del análisis e interpretación de patrones y estructuras en los datos que va a permitir posteriormente su aprendizaje, razonamiento y toma de decisiones dejando al margen la interacción del ser humano. Se trata de un campo de la inteligencia artificial.¹²

A pesar de que *Machine Learning* y *Python* se consideran herramientas útiles por las cuales se consigue la resolución de problemas y desarrollo de aplicaciones, difieren en que la programación general se centra en la implementación de algoritmos y reglas que han sido definidas por una persona con el fin de resolver un tipo de problema en específico, mientras que el aprendizaje automático se encarga de el desarrollo de algoritmos encargados de aprender de los datos para mejorar el rendimiento¹³ (ver Ilustración 18).

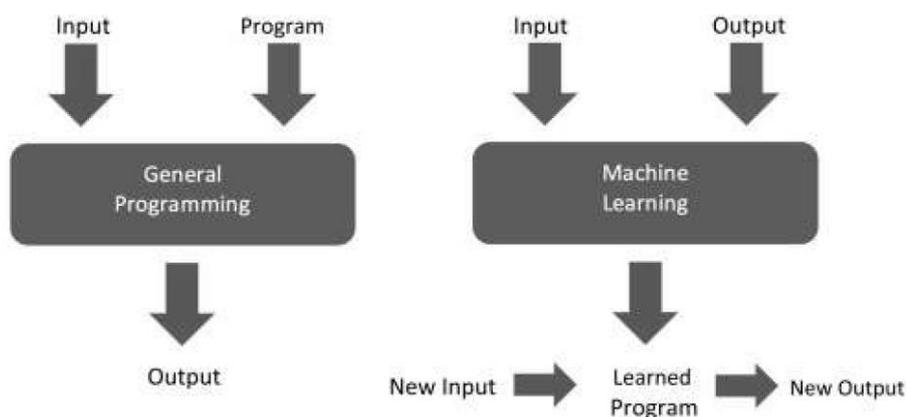


Ilustración 18. Programación general vs Machine Learning

Esta área de conocimiento se puede dividir en dos grandes apartados: aprendizaje supervisado y aprendizaje no supervisado, siendo el primero el que se encarga de desarrollar un modelo predictivo que se basa tanto en los datos de entrada como los de salida, mientras que el segundo se centra más en agrupar e interpretar los datos que se basan exclusivamente en los datos de entrada.

El aprendizaje supervisado es en el que se centra este trabajo, en concreto en la regresión, aunque haya también otros tipos de aprendizaje supervisado como la clasificación y el ranking, de tal manera que para poder entenderlo un poco mejor se tienen que definir ciertos conceptos:¹³

- **Características:** se posicionan en las columnas, y hacen referencia a las X, las variables. Se pueden tener tantas columnas como variables que se estudian. Para el estudio que se realiza en este TFG son variables como el número de átomos, momentos dipolares, etc...
- **Objetivo:** se refiere al objetivo que se quiere calcular. En lo que se refiere a este trabajo, el objetivo es la columna de densidad de las moléculas que se encuentran en la nube molecular TMC-1 en Tauro.
- **Muestras:** son las filas u observaciones.

Machine Learning engloba una serie de pasos que hacen que se pueda desarrollar el modelo. Estos pasos se pueden dividir en la adquisición de los datos, su preparación y posterior procesamiento. El proceso más largo y que conlleva la mayor parte del tiempo es la preparación de los datos ya que se deben tener limpios y ordenados para su estudio ya que los algoritmos de *Machine Learning* tienen algunas restricciones en cuanto a los datos de entrada.

Se puede tener un plan de lectura para la preparación de los datos que se puede englobar en tres grupos: ¹³

- **Limpieza:** Se centra en la preparación de los datos:
 - Eliminación de los duplicados: este paso se considera de gran importancia ya que estos datos pueden desacreditar la evaluación del rendimiento de un modelo.
 - Datos faltantes: La ausencia de datos se puede dar por razones varias como un error en la programación, un fallo de medición o sucesos aleatorios. Suelen estar representados por expresiones como NaN (“*not a number*”), grandes números negativos, interrogaciones o infinitos. Que se tenga una falta de datos no va ligado siempre a que se tenga una falta de información, pero hay que tener en cuenta que, si se tiene más de un 30% de datos faltantes, esa característica se podría eliminar mientras que si es menor del 30% se recomienda imputar un valor con sentido (media, moda, mediana) aunque la introducción de una aproximación conlleva la creación de ruido, se introduce el sesgo humano en el conjunto de datos y se puede perder relación entre las columnas de datos.
 - Valores atípicos: Se consideran a todos aquellos datos que se desvían del resto. Se pueden llegar a producir con un error a la hora de la introducción de datos, con un error de medición, de manipulación o de preprocesamiento de los datos. Encontrar los valores atípicos es de gran importancia ya que pueden llegar a afectar tanto las distribuciones como los patrones de la base de datos asimismo como al rendimiento de los modelos del *Machine Learning*. Por ello, la mejor forma de tratar estos datos es siempre asegurarse de conocer bien el significado de qué es un valor atípico, hay que asegurarse de que es evidentemente falso, que no se trata de una novedad y también asegurarse que no se pueda usar como una característica.
 - Escalado de características: Se trata de la transformación de los rasgos continuos en una gama más pequeña y se trata de un paso con importancia ya que las características con grandes magnitudes pueden superar de forma

incorrecta a las características que tengan una menor magnitud. De igual manera, escalar a magnitudes que tienen un valor más pequeño ayuda a mejorar la eficiencia computacional e incrementa la interpretabilidad de los coeficientes de las características.

Hay dos formas de escalar: mediante la estandarización y la normalización (ver Ilustración 19). La estandarización se caracteriza por ser muy eficaz siempre y cuando los datos se distribuyan de forma normal y es sensible a los datos atípicos, aunque puede distorsionar las distancias relativas entre los valores de las características, mientras que la normalización es eficaz independientemente de la distribución que se tenga, aunque no reduzca el efecto de los datos atípicos ni tampoco corrige la asimetría de la distribución.

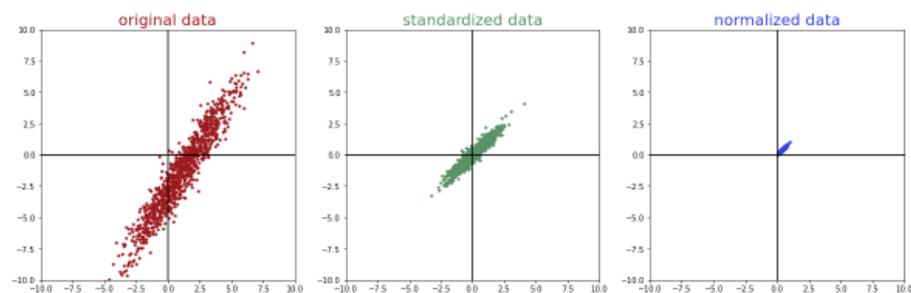


Ilustración 19. Normalización y estandarización de datos

- **Equilibrado de datos:** En un conjunto de datos, estos suelen ser desiguales o encontrarse desequilibrados para distintas clases. Es importante este paso ya que el aprendizaje automático aprende con el ejemplo y se tiende a predecir mal la clase infrarrepresentada. Las mejores estrategias del equilibrado se pueden englobar en 3 tipos: sobre muestreo de la clase minoritaria, submuestreo de la clase mayoritaria y el cálculo de nuevas instancias de la clase minoritaria. Las dos primeras técnicas se encargan de duplicar instancias de la clase minoritaria o sobre muestrear la clase mayoritaria; esto se hace para compensar las clases que se encuentran mucho mas o mucho menos representadas. La principal estrategia de sobre muestreo es SMOTE (*Synthetic Minority Over-sampling TEchnique*), o técnica de sobre muestreo de minorías sintéticas (ver Ilustración 20) que se encarga de generar nuevas instancias a partir de las ya existentes basándose en combinaciones lineales de puntos existentes. Es importante que el equilibrado se realice únicamente en el conjunto de entrenamiento ya que se quiere que el conjunto de pruebas sea lo más representativo posible de la realidad.

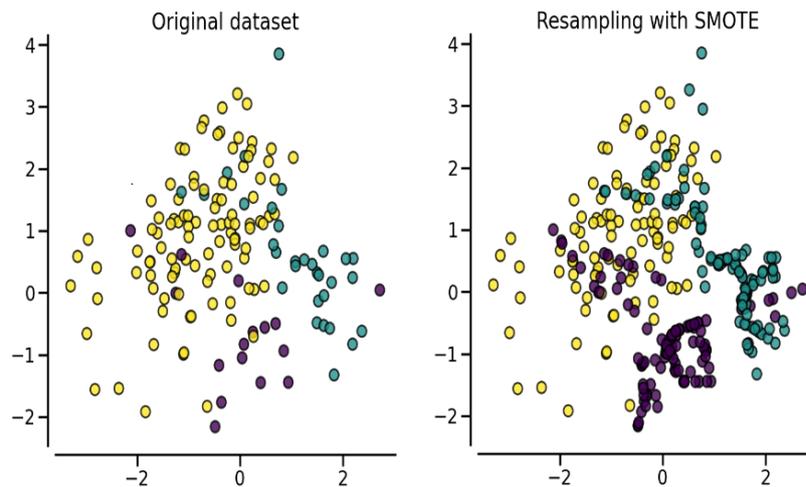


Ilustración 20. Ejemplo SMOTE

- **Ingeniería de características:** Es el proceso que se encarga de seleccionar, transformar y crear nuevas características a partir de los datos brutos con el fin de mejorar el rendimiento de los del *Machine Learning*. Se puede dividir en tres procesos importantes:
 - Codificación: Se centra en la transformación de todos los datos no numéricos en datos numéricos equivalentes, esto es importante ya que muchos de los algoritmos que se usa en el aprendizaje automático solo son capaces de procesar datos numéricos.
 - Discretización: Proceso que se encarga de transformar los datos continuos en datos concretos.
 - Creación de características: De esta forma se puede crear información adicional que se encarga de ayudar en la mejora del rendimiento del modelo.
- **Selección de características:** A través de este último paso se eliminarán todos aquellos rasgos que no proporcionen ningún tipo de información y para llevarlo a cabo se puede usar una selección estadística univariable, como la correlación de rasgos, o la selección estadística multivariable donde se puede tener la permutación de rasgos. Lo mejor es eliminar el mayor número de rasgos sobre todo aquellos que tengan una alta correlación hasta que el rendimiento del modelo caiga de forma significativa.

3.2.1. SCIKIT-LEARN

Una herramienta muy útil en el aprendizaje automático es *Sklearn* (*Scikit-learn*). Se trata de una biblioteca que provee diferentes herramientas de procesamiento de datos, como puede ser el manejo de valores faltantes y la selección de características, y procesado y selección de modelos que otorgan funciones para evaluar y comparar modelos mediante el uso de técnicas de validación cruzada o curvas de aprendizaje. Se organiza en módulos donde cada módulo contiene distintos tipos de clases.

4. TRATAMIENTO DE DATOS

4.1. PREPARACION DE LA BASE DE DATOS

Se van a crear dos bases de datos distintas: En la primera de ellas se van a introducir todas las especies que se vayan recopilando que tengan columna de densidad detectada con los datos químico-físicos (ver Anexos – Tabla A1) mientras que la segunda base de datos que se ha creado se centra más en datos espectrales de las moléculas recopiladas en la primera (ver Anexo - Tabla A2). Ambas bases se relacionan a través de un número denominado número *Tag*¹⁴ y no es un dato aleatorio, sino que se trata de un criterio establecido por el CDMS (*The Cologne Databae for Molecular Spectroscopy*) donde recopila en un catálogo (*The Catalog Directory*) todas las especies. El número *Tag* está formado por seis dígitos, los tres primeros de la etiqueta representan el peso molecular de la especie en unidades de masa atómica, el cuarto dígito es el número “5” que se usa para evitar conflictos con el catálogo del JPL (*Jet Propulsion Laboratory*) ya que usan un esquema equivalente y, los dos últimos dígitos, se utilizan para contar las especies con el mismo peso molecular.¹⁵

4.1.1. COLUMNA DE DENSIDAD

La primera base de datos va a recopilar todas las moléculas que se vayan encontrando con presencia y posible presencia en TMC-1. Esta presencia o no presencia se va a determinar gracias a la densidad de columna de las moléculas que se define como la cantidad de materia (moléculas, partículas o átomos) que se encuentra contenido en un cilindro imaginario (normalmente con una sección transversal del área de 1cm²) entre un observador y un objeto astronómico.

$$N = \int_0^D n(s) ds$$

Donde N es la densidad de columna, $n(s)$ es indicativo de la densidad local del material en un punto s a lo largo de ese cilindro imaginario (línea de visión) y D es la profundidad de la nube. La unidad más comúnmente usada en la densidad de columna es la de partículas por centímetro cuadrado (cm^{-2}).

Se trata de una medida integral de la cantidad total de materia que hay presente en una columna de una determinada área que se extiende a lo largo de la profundidad de una nube molecular o una parte del espacio. Es un parámetro con gran importancia ya que deriva directamente de las observaciones que se realizan en diferentes longitudes de onda (IR, UV, Visible...) y que proporcionan datos sobre la cantidad de material que hay en una línea de visión. Así mismo, es un parámetro crucial para poder predecir tanto la dinámica como la química del medio interestelar y comprender la estructura y la evolución de las galaxias.

4.1.2. CONSTANTES DE ROTACIÓN

Otros datos que se tienen que buscar para el estudio de las moléculas son las constantes de rotación. Un modelo aproximado que se considera de gran utilidad es el modelo del rotor rígido donde los núcleos de una molécula diatómica se encuentran unidos mediante un enlace que se considera una varilla rígida y sin peso, de tal manera que la molécula rota sin que las distancias internucleares se vean afectadas.

Para entender este modelo se definen el momento de inercia a través de la siguiente ecuación:

$$I = \sum_i m_i r_i^2$$

Y también el momento angular:

$$L = I\omega$$

Si se considera una única partícula, tanto la velocidad angular (ω) como el momento angular (L) son vectores que apuntan hacia fuera en el plano de rotación y, en este caso, ambos apuntan en la misma dirección (ver Ilustración):

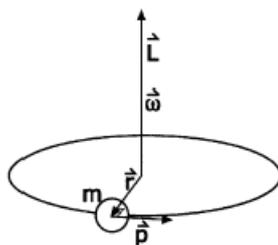


Ilustración 21. Movimiento circular de una partícula con masa m ¹⁶

Mientras que, si un objeto extendido está girando, L y ω no tienen por qué apuntar en la misma dirección.

Teniendo en cuenta una molécula plana, el momento de inercia fuera del plano es igual a la suma de los dos momentos de inercia que hay dentro del plano. Según sea la magnitud de los momentos de inercia se tiene un esquema de etiquetado para los ejes que pasan a ser a , b o c que es el siguiente:

$$I_A \leq I_B \leq I_C$$

Donde el momento de inercia I_C es siempre mayor que el momento de inercia I_A .

Según los valores que tomen cada uno de los momentos de inercia se va a tener una clasificación de las moléculas:

- Molécula lineal: $I_B = I_C, I_A = 0$
- Trompo esférico: $I_A = I_B = I_C$
- Trompo simétrico oblate: $I_A < I_B = I_C$
- Trompo simétrico prolata: $I_A = I_B < I_C$

- Trompo asimétrico: $I_A < I_B < I_C$

Para las moléculas lineales y los trompos simétricos, la posición de los ejes principales puede ser representadas en expresiones analíticas y el cálculo de los niveles de energía es sencillo. Pero la mayoría de las moléculas son trompos asimétricos y por tanto el cálculo no es tan sencillo.

Lo niveles energéticos rotacionales de las moléculas lineales vienen dados por los términos de los números cuánticos J y M_J , que son el número cuántico del momento angular y a la proyección del momento angular en los ejes fijos en el espacio respectivamente y toman unos valores determinados donde $J = 0, 1, 2, \dots$ y $M_J = -J, \dots, 0, \dots, +J$.

Por otra parte, la energía de rotación solo depende del número cuántico J , por lo que los niveles de rotación se encuentran degenerados en $2J+1$ veces. La expresión para la energía y funciones de onda se puede escribir de la siguiente manera:

$$E_{rot} = E(J) = \frac{\hbar^2}{2I} J(J+1) = B(J+1)$$

Donde B es la constante de rotación y se expresa como:

$$B = \frac{h^2}{8\pi^2 I_B}$$

El estado de rotación de un trompo simétrico depende de la componente de momento angular sobre el eje de simetría de la molécula (K) por lo que la función de onda depende los tres números cuánticos: $\psi(J, K, M_J)$.

La energía de rotación depende del tipo de molécula:

- Molécula prolata (alargada): $E(J, K) = BJ(J+1) + (A - B)K^2$
- Molécula oblate (achatada): $E(J, K) = BJ(J+1) + (C - B)K^2$

Las constantes de rotación restantes se pueden obtener con las siguientes expresiones:

$$A = \frac{h^2}{8\pi^2 I_A} \quad C = \frac{h^2}{8\pi^2 I_C}$$

Si se tiene un trompo asimétrico la descripción del movimiento de la molécula como una rotación en torno a un único eje molecular no es posible. Debido a esto, los estados de rotación se tienen que describir teniendo en cuenta un diagrama de correlación (ver Ilustración 22) entre

los casos límite de los niveles de energía del tropo simétrico prolata, lado izquierdo del diagrama, (J, K_a) y los niveles del tropo simétrico oblata, lado derecho del diagrama, (J, K_c) .

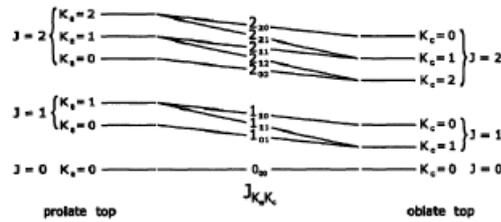


Ilustración 22. Diagrama de correlación¹⁶

Cada estado de rotación del tropo asimétrico viene dado por el número cuántico del momento angular total J y dos pseudonúmeros cuánticos K_a y K_c que hacen referencia a los números cuánticos de los casos extremos del tropo simétrico de tal forma que cuando se da una transición se va a indicar como $J_{K_a', K_c'} \leftarrow J_{K_a'', K_c''}$

Un tropo asimétrico va a tener un grado de asimetría, el cual se puede cuantificar mediante el parámetro de asimetría de Ray (k), que toma unos valores desde la unidad negativa, que corresponde al límite superior prolata, hasta la unidad positiva que referencia al límite superior oblata. Este parámetro de asimetría es otro de los datos a introducir en la base de datos y que se calcula como:

$$k = \frac{2B - A - C}{A - C}$$

4.1.3. FUNCIONES DE PARTICIÓN

A parte de recopilar los datos de las constantes de rotación de la primera base y los momentos dipolares de las moléculas, se va a tener un parámetro de gran importancia que es la función de partición de cada molécula a distintas temperaturas. La función de partición se trata de la herramienta por la cual se describe la distribución de los niveles energéticos de una molécula, es decir, cuales de los estados son accesibles a una temperatura determinada.

$$Q = \sum_i g_i e^{-\frac{\epsilon_i}{k_B T}}$$

Donde g_i es degeneración del i -ésimo estado, ϵ_i es la energía del estado i , k_B la constante de Boltzmann y T la temperatura.

Cuanto mayor sea el valor de la función de partición mayor es el número de estados que se encuentran poblados de forma significativa (ver Ilustración 23). Este aumento de la función de partición se puede dar tanto por el aumento de la temperatura como por la disminución de la diferencia energética que hay entre los niveles energéticos.



Ilustración 23. Estados poblados según el valor de la función de partición

La función de partición total de una molécula tiene tres contribuciones principalmente: la función de partición rotacional, la vibracional y la electrónica. En este trabajo, como se ha dicho anteriormente, las moléculas van a tener una energía tan baja debido a la baja temperatura que el movimiento principal que se va a dar es la rotación de tal manera que se va a centrar únicamente en la función de partición rotacional.

Teniendo en cuenta que existen moléculas lineales y no lineales ambas van a tener funciones de partición distintas.

Para una molécula lineal, la función de partición se va a poder aproximar a altas temperaturas y se expresa como:

$$Q_r = \frac{8\pi^2 I k_B T}{\sigma h^2} = \frac{T}{\sigma \theta_r} = \frac{k_B T}{h B}$$

Donde B es la constante de rotación de la molécula, θ_r es la temperatura característica rotacional y σ es el centro de simetría de la molécula, en este caso si la molécula lineal tiene centro de simetría, $\sigma = 2$, mientras que si no existe ese centro de simetría $\sigma = 1$.

Por otro lado, se tienen las moléculas no lineales poliatómicas. Hay distintos tipos de trompos como se ha visto en el apartado anterior donde para cada uno de ellos los momentos de inercia toman distintos valores. Se van a tener tres momentos de inercia y tres temperaturas características de rotación.

$$Q_r = \frac{\pi^{1/2}}{\sigma} \left(\frac{T^3}{\theta_{r,a} \theta_{r,b} \theta_{r,c}} \right)^{1/2}$$

4.1.4. FUNCIONES DE BASE

La elección de una función de base es crucial ya que se encarga de representar de manera similar a la función real y a la hora de realizar cálculos computacionales pueden tardar mucho en resolverse o, en el peor de los casos, no llegar a la convergencia.^{17,18}

Se ha elegido el funcional B3LYP a la hora de la optimización de la energía ya que se trata de un funcional híbrido, además de que es el funcional más usado. Utiliza el funcional de correlación Becke's 3, con tres parámetros para introducir la energía de correlación e intercambio y el funcional LYP (Lee Yang y Parr).¹⁸

La base elegida de donde se han sacado todos los datos en este trabajo es la base 6-311G++ (2d,p)¹⁹. Es la base que representa de forma más precisa las moléculas ya que tiene una complejidad mayor que cualquier otra base. Se introducen tanto funciones de polarización como funciones difusas. Está constituida por 6 funciones gaussianas para los orbitales internos. Por otro lado, cada orbital de valencia consta de 3 funciones (triple-zeta), una de ellas es contracción de 3GTO y las restantes tienen una única gaussiana primitiva. Los dos símbolos "+" hacen referencia a que a la base se le añaden funciones difusas s y p (ver Ilustración 24) a los átomos pesados como pueden ser los átomos de C, O y N; y funciones s a los átomos de hidrogeno. El término (2d,p) significa que se añaden dos funciones de polarización d a los átomos pesados y una única función de polarización p a los átomos de hidrógeno (ver Ilustración 25).²⁰



Ilustración 24. Esquema utilización de una función difusa²¹

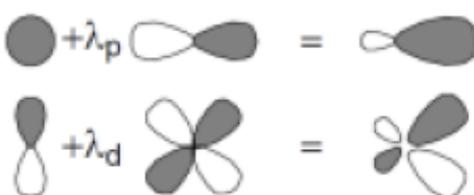


Ilustración 25. Funciones de polarización²¹

4.2. ANÁLISIS DE DATOS

Tras la creación de las bases de datos ya se va a poder empezar a desarrollar el código para el análisis de los datos obtenidos, es el proceso más largo como se ha visto anteriormente ya que en este apartado hay que realizar una limpieza y tratamiento de los datos. El desarrollo de dicho código se va a englobar en una serie de pasos que se enumeran y explican a continuación:

- **Importar las bases de datos:** Las dos tablas que se han creado (las denominadas “Moléculas TMC-1” y “Molecules-Spectral-Data”) se tienen que importar en formato Pandas desde Jupyter Notebook.
- **Agrupar tablas:** Una vez importadas las tablas se han agrupado mediante el número *Tag* del CDMS, este agrupamiento es importante ya que se va a querer únicamente las moléculas que se tienen en la primera tabla.
- **Limpieza de datos:** Se mira el porcentaje que se tiene de NaN (“*not a number*”) en cada columna. En columnas como por ejemplo las del momento dipolar o las de las constantes de rotación hay datos que no se tienen por lo que habría que sustituirlos por ceros o por el valor máximo que exista entre los datos que se tengan. En los datos correspondientes a la función de partición a distintas temperaturas se eliminan todas las columnas con más de un 70% de NaN teniendo en cuenta que se tiene que dejar al menos una función de partición a alta temperatura y otra a baja temperatura.
- **Creación de columnas:** Para complementar la información de las tablas se van a crear columnas con distinta información: pesos moleculares, grado de insaturación, tipo de molécula...

Una vez se han limpiado todos los datos innecesarios para el análisis se realiza una selección de características. Se trata de un proceso por el cual se eliminan las características que no aportan ningún tipo de información relevante. Esta selección de rasgos es importante ya que, si se tiene una entrada con una calidad mala, inducirá ruido y desestabilizará el modelo dando lugar a una salida de la misma calidad que la entrada. De igual manera hay que tener cuidado de que no se tenga el número suficiente de datos por el cual se establezca una relación significativa entre las características. En este caso, si aumenta el número de características, existe una crecida tipo exponencial de la cantidad de datos que se necesitan para poder generalizar con la precisión adecuada.²²

En este TFG, se aplica una primera técnica de selección que consiste en eliminar las características que se encuentren altamente relacionadas entre sí y se conoce como la *correlación de Pearson* que se encarga de eliminar esa alta correlación debido a que la información que proporciona es redundante. Cabe destacar que se pueden eliminar tantas características como uno crea conveniente (siempre empezando con las de mayor correlación) hasta que el rendimiento del modelo comience a disminuir de forma significativa.

En la *correlación de Pearson* se analiza la fuerza de correlación entre características que se visualizan gráficamente. El valor de la correlación varía entre 0 y 1, tomando distintos

significados según el valor que se tenga de r , siendo 0 la nula correlación entre las características y 1 la correlación elevada que no proporciona ningún tipo de información relevante (ver Tabla 1).²³

Valor de r	Fuerza de correlación
0.0 < 0.1	No hay correlación
0.1 < 0.3	Poca correlación
0.3 < 0.5	Correlación media
0.5 < 0.7	Correlación alta
0.7 < 1.0	Correlación muy alta

Tabla 1. Fuerza de correlación

5. RESULTADOS

A partir de las bases de las tablas que recogen los datos de TMC y de moléculas se ha procedido a importarlas en Python para proceder a su preparación con el fin de que estén preparadas para lanzar los modelos de *machine learning*. Al juntar las tablas queda el siguiente *DataFrame* (ver Ilustraciones

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Formula                                   141 non-null    object
1   Smiles                                    82 non-null    object
2   Tag CDMS                                  81 non-null    float64
3   Column Densitylog10 cm-2                 140 non-null    float64
4   References                                45 non-null    object
5   Dipole momentDebye                        8 non-null     float64
6   Cte rot a_x                               41 non-null    object
7   Cte B_x                                   81 non-null    float64
8   Cte C_x                                   41 non-null    float64
9   mua                                       76 non-null    float64
10  Mub                                       17 non-null    float64
11  MuC_x                                    1 non-null     float64
12  Mol formula                               80 non-null    object
13  Name                                      80 non-null    object
14  Cte rot a_y                               41 non-null    object
15  Cte B_y                                   80 non-null    float64
16  Cte C_y                                   41 non-null    float64

17  MuA                                       75 non-null    float64
18  MuB                                       17 non-null    float64
19  MuC_y                                    1 non-null     float64
20  MuTot                                    0 non-null     float64
21  Q(300.0)                                80 non-null    float64
22  Q(225.0)                                79 non-null    object
23  Q(150.0)                                80 non-null    float64
24  Q(75.00)                                 79 non-null    float64
25  Q(50.00)                                 2 non-null     float64
26  Q(37.50)                                 79 non-null    float64
27  Q(25.00)                                 1 non-null     float64
28  Q(18.75)                                 79 non-null    float64
29  Q(12.50)                                 1 non-null     float64
30  Q(9.375)                                 79 non-null    float64
31  Q(5.000)                                 51 non-null    float64
32  Q(2.725)                                 51 non-null    float64
33  B3LYP/6311**                             35 non-null    object
34  Parametro de asimetria de Ray             40 non-null    float64
35  Número de insaturaciones (G.I.= (2C-H+N-X+2)/2) 0 non-null    float64
dtypes: float64(27), object(9)
memory usage: 39.8+ KB
```

Ilustración 26. *DataFrame* inicial

A partir del *DataFrame* se proceder a la limpieza de los datos. En primer lugar, se tienen que limpiar los datos *null*. En caso de los momentos dipolares, se han rellenado los huecos con ceros y se ha empleado la aproximación de utilizar el momento dipolar más grande entre los tres presentes como el momento dipolar total. Así mismo, se han eliminado las columnas 25, 27 y 29 que se corresponden con las funciones de partición a 50'0, 25'0 y 12'50 K respectivamente ya que sólo se tienen una o dos filas con datos. En el caso de los datos vacíos en las constantes de rotación, que corresponden con las moléculas lineales, se han incluido ceros en todos esos huecos.

Por otra parte, se ha desglosado la fórmula molecular para incorporar columnas que incluyan el número de átomos de ese elemento que contiene la especie química. Se ha añadido el peso molecular de la especie y se ha calculado el parámetro de asimetría de Ray y el número de insaturaciones de la molécula.

En cuanto al *target* del modelo, se corresponde con la columna de densidad. Se ha tratado de dos maneras. Por un lado, se han asignado 0 para todas aquellas moléculas que no se han detectado y 1 para las detectadas. Esto es la “y” que se utilizará en nuestros modelos de regresión logística. Por otro lado, se tiene el caso de los modelos de regresión línea; en este caso se necesita adaptar el *target* para que refleje si la especie está presente o no y su abundancia. Por ello, en las moléculas detectadas, se ha dejado el valor tal cual mientras que en las no detectadas se ha restado al valor de la columna de densidad del hidrógeno H₂²⁴, el límite de detección y se le ha dado el valor negativo.

Por último, se han normalizado los datos incluidos en cada una de las columnas para evitar tener diferente peso en los coeficientes de las columnas.

Tras todas estas operaciones el *DataFrame* ha quedado como se indica a continuación (ver Ilustración 27).

```
data_numeric.head()
```

]:

	Cte A	Cte B	Cte C	MuA	MuB	MuC	MuTot	Q(300.0)	Q(225.0)	Q(150.0)	...	Parametro de asimetria de Ray	C	H	O	N	S
0	0.173243	0.001432	0.022725	0.144231	0.0	0.0	0.144231	0.000146	0.035083	0.109716	...	0.112522	0.636364	0.500	0.0	0.000000	0.0
1	0.173243	0.004389	0.059446	0.116058	0.0	0.0	0.116058	0.000056	0.013389	0.041852	...	0.112522	0.454545	0.500	0.0	0.000000	0.0
2	0.172110	0.001432	0.022719	0.519231	0.0	0.0	0.519231	0.000440	0.105706	0.330639	...	0.112522	0.545455	0.375	0.0	0.333333	0.0
3	0.172110	0.004460	0.060322	0.456731	0.0	0.0	0.456731	0.000055	0.013238	0.041378	...	0.112522	0.363636	0.375	0.0	0.333333	0.0
4	0.000000	0.109139	0.000000	0.326923	0.0	0.0	0.326923	0.000001	0.000382	0.001427	...	NaN	0.000000	0.125	0.0	0.666667	0.0

Ilustración 27. *DataFrame* final

A partir de esta base de datos se procede a realizar un análisis de los datos. En primera instancia, se hace una primera selección de características a través de la correlación de Pearson (ver Ilustración 28).

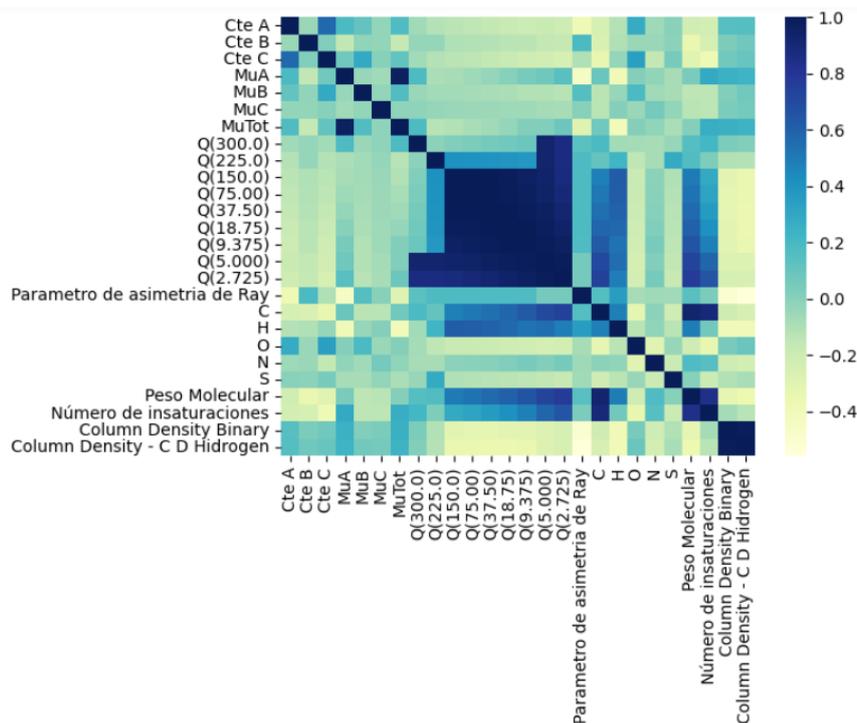


Ilustración 28. Correlación de Pearson inicial.

Analizando la imagen se pueden observar zonas oscuras de muy alta correlación en la zona central, en la diagonal y en algún punto suelto de la gráfica, como puede ser el momento dipolar total con el momento dipolar A ya que en este caso se ha tomado el mayor valor (o único) valor del momento dipolar como el total, también es el caso de las funciones de partición entre sí, pese a estar a distintas temperaturas no aportan información relevante por lo que se van a eliminar todas menos a las funciones de partición a mayor y menor temperatura.

De esta forma al eliminar todos aquellos datos que no proporcionan ningún tipo de información de interés, el gráfico de la correlación de Pearson queda de la siguiente manera (ver Ilustraciones 29 y 30).

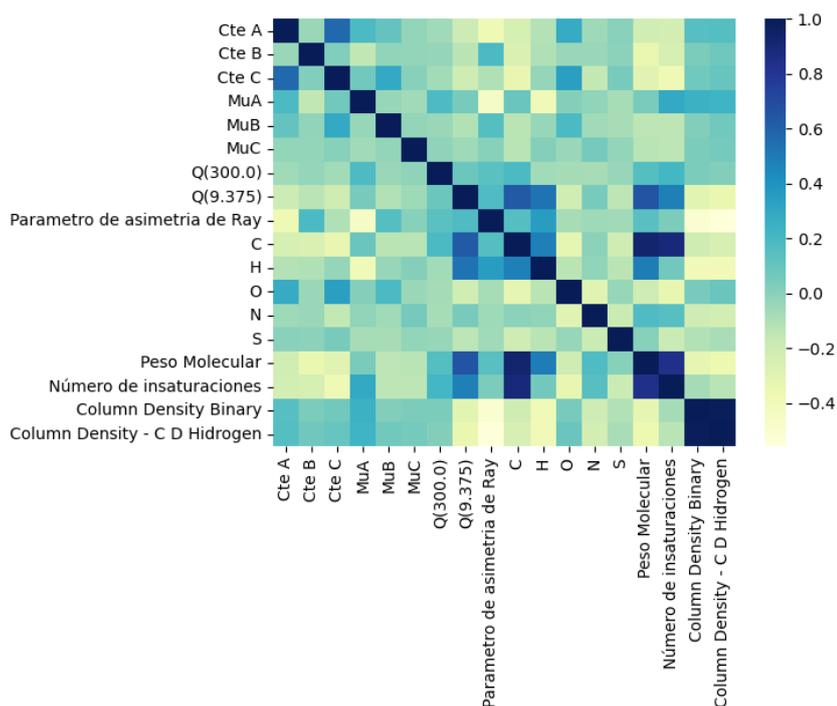


Ilustración 29. Correlación Pearson final

feature_1	feature_2	correlation
Column Density Binary	Column Density - C D Hidrogen	0.993202
Column Density - C D Hidrogen	Column Density Binary	0.993202
Peso Molecular	C	0.928004
C	Peso Molecular	0.928004
C	Número de insaturaciones	0.896352
Número de insaturaciones	C	0.896352
Peso Molecular	Número de insaturaciones	0.837167
Número de insaturaciones	Peso Molecular	0.837167
Q(9.375)	Peso Molecular	0.656139
Peso Molecular	Q(9.375)	0.656139
C	Q(9.375)	0.624367
Q(9.375)	C	0.624367
Cte A	Cte C	0.572602
Cte C	Cte A	0.572602
Q(9.375)	H	0.530288
H	Q(9.375)	0.530288
Peso Molecular	H	0.492371
H	Peso Molecular	0.492371
C	H	0.482659
H	C	0.482659

Ilustración 30. Correlaciones numéricas finales

La Ilustración 29 para poder visualizar los datos de forma gráfica está muy bien, pero hay que fijarse mejor en la Ilustración 30 ya que se pueden ver las correlaciones en formas numéricas

para poder hacer un estudio más exhaustivo. Se observa que las correlaciones vienen dadas por parejas.

La mayor correlación que se tiene, fuera a parte de la diagonal, es la existente entre la columna de densidad binaria e hidrógenos ya que una viene dada por la otra debido a que una es en números binarios mientras que la otra es en positivo/negativo. Ambas columnas se mantienen para poder hacer los modelos y así elegir el objetivo que más interese (regresión o clasificación).

El peso molecular tiene una fuerte correlación con el número de carbonos, lo cual es lógico ya que a mayor número de carbonos más peso molecular se tiene. Lo mismo pasa con el número de insaturaciones, cuanto mayor sea el número de carbonos mayor es la posibilidad de que se formen insaturaciones en la molécula.

Todas las características se tienen que referir a la columna de densidad ya que en casos de fuertes correlaciones siempre se tiene que descartar la característica que tiene menos relación con el *target*.

Entre estas tres características (peso molecular, carbonos e insaturaciones) en relación con la columna de densidad, el número de insaturaciones tiene más peso que las otras dos, afectando más al resultado de la columna de densidad, y, por tanto, a la posibilidad de encontrar con mayor facilidad moléculas con insaturaciones en TMC-1 que aquellas que no tienen ningún tipo de insaturación.

Si se hace lo mismo con el resto de las características, se ve que las más significativas con respecto a la columna de densidad son cuatro: las constantes rotacionales que, como se ha explicado en la teoría (ver apartado 5.1.2), al estar en el medio interestelar los principales movimientos moleculares debido a la temperatura existente en dicho medio son los de rotación. Luego también tenemos los momentos dipolares, la función de partición a 300K y el oxígeno. Éste último está correlacionado en un alto grado con los momentos dipolares ya que hace que las moléculas sean más polares lo cual tiene sentido ya que dicho átomo hace que aumente la polaridad de los enlaces de una molécula.

Por otro lado, el resultado obtenido para la función de partición resulta curioso. Por lógica, se deberían tener resultados más informativos a 10K que a 300K dado que la nube molecular de Tauro 1 se encuentra a una temperatura media de 20K. Esto se puede explicar por la influencia del tamaño y del peso molecular de las moléculas sobre la función de partición. En el caso de las moléculas pequeñas a temperaturas bajas, los niveles de rotación accesibles son bastante limitados, resultando en una función de partición baja. En cambio, a temperaturas más altas, los niveles rotacionales se vuelven más accesibles, aumentando significativamente el valor de la función de partición. Para moléculas de gran tamaño, a bajas temperaturas, la función de

partición puede ser mayor que para moléculas pequeñas debido a que los niveles de rotación son más densos y se pueden poblar con mayor facilidad. A temperaturas elevadas, la función de partición toma valores aún mayores por el gran número de niveles rotacionales accesibles.

6. CONCLUSIONES

Gracias a este TFG se ha podido comprobar que existe una sobre abundancia de oxígeno con respecto a otros átomos como el carbono o el nitrógeno en la nube molecular TMC-1. Este resultado preliminar es significativo porque aumenta la probabilidad de obtener resultados positivos al buscar nuevas moléculas en TMC-1. Al enfocar las búsquedas en moléculas que contienen oxígeno en su estructura, se incrementa la facilidad de detección. Todavía queda mucho trabajo por delante. Es necesario implementar un modelo de *machine learning* capaz de predecir moléculas específicas que puedan ser detectadas y evaluar la eficacia del modelo para mejorar los algoritmos. Además, al ampliar las bases de datos a otras nubes moleculares en el espacio interestelar, el programa podría predecir la probabilidad de encontrar una molécula específica cuando se le consulte. Esto guiaría los estudios de laboratorio necesarios y aumentaría la probabilidad de detección de moléculas.

En general, el uso de nuevos métodos de investigación, como la inteligencia artificial, facilita el estudio de la química en el medio interestelar. Analizar una gran cantidad de datos relacionados por múltiples variables y clasificar y definir cuáles son las más importantes para predecir nuevos objetivos moleculares es una tarea difícil sin ayuda computacional. Por ello, buscar herramientas que ayuden a racionalizar las especies químicas presentes en el medio interestelar es de fundamental importancia para discernir los caminos que han llevado al origen de la vida en nuestro planeta.

7. BIBLIOGRAFÍA

- ¹“Astronomía : Acerca de,” (n.d.). Recuperado de: <https://astronomia.ign.es/web/guest/oan/acercade>
- ² S.L. Johansen, “Nitrogen-Heterocycles in the interstellar medium: Experimental and computational approaches to an astrochemical mystery,” (2021).
- ³“Vista de EL GRAN ROMPECABEZAS QUÍMICO DE NUESTRO UNIVERSO: LA NUBE OSCURA TMC-1,” (n.d.). Recuperado de: <http://rd.buap.mx/ojs-dm/index.php/rdicuap/article/view/1243/1284>
- ⁴“Colapso gravitatorio - Wikipedia, la enciclopedia libre,” (n.d.). Recuperado de: https://es.wikipedia.org/wiki/Colapso_gravitatorio
- ⁵“Molecules in Space,” (n.d.). Recuperado de <https://cdms.astro.uni-koeln.de/classic/molecules>
- ⁶“McGuire Research Group,” (n.d.). Recuperado de: <https://mcguirelab.mit.edu/chemistry.html>
- ⁷ Brad. Dayley, *Python Phrasebook: Essential Code and Commands* (Sams Pub, 2006). ISBN: 9780672329104
- ⁸“Cursos online gratis de Harvard, MIT y más | edX,” (n.d.).
- ⁹ T.E. Oliphant, *Guide to NumPy* (2006). Recuperado de <https://numpy.org/doc/stable/user/index.html#user>
- ¹⁰“User Guide — pandas 2.2.2 documentation,” (n.d.). Recuperado de: https://pandas.pydata.org/docs/user_guide/index.html#
- ¹¹“Creating DataFrames Reshaping Data-Change layout, sorting, reindexing, renaming,” (n.d.). Recuperado de: <http://pandas.pydata.org/>
- ¹²“Machine Learning: definición, funcionamiento, usos,” (n.d.). Recuperado de: <https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos>
- ¹³“User Guide — scikit-learn 1.5.0 documentation,” (n.d.). Recuperado de: https://scikit-learn.org/stable/user_guide.html
- ¹⁴“The Cologne Database for Molecular Spectroscopy,” (n.d.). Recuperado de: <https://cdms.astro.uni-koeln.de/classic/entries/>
- ¹⁵ H.S.P. Müller, F. Schlöder, J. Stutzki, and G. Winnewisser, “The Cologne Database for Molecular Spectroscopy, CDMS: a useful tool for astronomers and spectroscopists,” (2005). DOI: 10.1016/j.molstruc.2005.01.027
- ¹⁶ P.F. Bernath, *Spectra of Atoms and Molecules*, 2ª (2005). ISBN: 0-19-507598-6
- ¹⁷ J. Andrés, and J. Beltrán, *QUIMICA TEÓRICA Y COMPUTACIONAL* (2000). ISBN: 978-84-15443-27-8
- ¹⁸ M.E. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, and D.J. Fox, “Gaussian 16,” (2016). Recuperado de: <https://gaussian.com/dft/>

¹⁹“CCCBDB List Calculated Energies,” (n.d.). Recuperado de:
<https://cccbdb.nist.gov/energy1x.asp>

²⁰ M.J. Frisch, J.A. Pople, and J.S. Binkley, “Self-Consistent Molecular Orbital Methods 25. Supplementary Functions for Gaussian Basis Sets,” *J.Chem. Phys.* **80**, 3265–3269 (1984). DOI: 10.1063/1.447079

²¹“Conceptos básicos de mecánica cuántica,” (n.d.). Recuperado de:
https://www.unirioja.es/cu/enriquez/docencia/Quimica/LCAO-MO-SCF1112_partA.pdf

²²“The Curse of Dimensionality in Classification,” (n.d.). Recuperado de:
<https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/>

²³“Correlación de Pearson - Explicación sencilla - DATAtab,” (n.d.). Recuperado de:
<https://datatab.es/tutorial/pearson-correlation>

²⁴ A.A. Ramos, ★ C Westendorp Plaza, D. Navarro-Almaida, P. Rivière-Marichalar, V. Wakelam, and A. Fuente, “A fast neural emulator for interstellar chemistry,” *MNRAS* **000**, 1–12 (2024).

8. ANEXOS

Tabla A1. Recopilación de moléculas presentes en TMC-1

Fórmula	Tag	Columna de densidad (log ₁₀ cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
CH ₃ C ₆ H	88502	12,4914	159140	778,245	778,245	1,5		
CH ₃ C ₄ H	64507	13,4771	159140	2035,747	2035,747	1,207		
CH ₃ C ₅ N	89501	11,9243	158099	778,04	778,04	5,4		
CH ₃ C ₃ N	65503	12,2553	158099	2065,74	2065,74	4,75		
N ₂ H ⁺	29506	12,6990		46586,87		3,4		
NH ₃	17506	14,6998	10454,26	4041,15	2972,077			1,4719
CH ₃ OH	32504	13,1614	127523,4	24692,5	23760,3	0,899	-1,44	
C ₃ H ⁺	37505	13,4800		11244,95		3		
C ₃ H	37501	12,7497		11189,1		3,55	0,5	
1-C ₃ H ₂	38501	11,7701	288775	10588,64	10203,97	4,1		
c-C ₃ H ₂	38508	13,1695	35092,51	32212,995	12764,2823		3,27	
C ₃ H ₂	38501	12,3979	288775	10588,64	10203,97	4,1		
CH ₃ CCH	40502	14,0607	159140	8545,877	8545,877	0,784		
C ₂ O		12,5705						
CH ₂ CN	40505	13,5798	285130	10246,23	9876,56	3,5		

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
CH ₃ CN	41505	12,6096	158099	9198,8992	9198,8992	3,922		
HNCO	43511	13,0294	918594	11071,768	10909,89	1,58	1,35	
CS	44510	13,4594		24495,56		1,958		
CH ₃ CHO	45529	12,4298	55365,1	10124,79	8838,413	2,516	1,07	
HCS ⁺	45506	12,7589		21337,14		1,958		
H ₂ CS	46509	13,6201	291613,3	17698,994	16652,498	1,6491		
SO	48501	13,6702		21523,556		1,535		
C ₄ H	49503	13,4298		4758,656		2,1		
C ₄ H ₂	50503	13,3365	286234	4503,31	4428,61	4,1		
C ₃ N	50511	13,5502		4947,62		2,85		
HNC ₃	51528	11,6803		4668,336			5,665	
C ₃ O	52501	11,9201		4810,8864		2,391		
HC ₃ NH ⁺	52503	11,8698		4328,997		1,61		
CH ₂ CHCN	53515	12,8102	49850,697	4971,164	4513,877	3,815	0,894	
HCCCHO	54510	11,2601	68035,3	4826,22	4499,59	2,359	1,468	
C ₂ S	56502	14,0086		6477,75		2,88		
OCS	60503	13,2601		6081,492		0,7152		
C ₅ H	61505	12,2695		2395,127		4,88		

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
C ₃ S	68503	13,1399		2890,38		3,704		
C ₆ H	73501	12,7404		1391,187		5,54		
HC ₃ N	51501	14,243		4549,059		3,73172		
HCCNC	75503	13,8256		1331,3327		4,33		
HC ₅ N	75501	11,5173		1401,182		3,25		
HC ₄ NC	99501	13,5623		564,0011		4,82		
HC ₇ N	99504	11,6064		582,52		3,49		
HC ₆ NC	123501	13,3345		290,5183		5,2		
HC ₉ N	147501	12,0170		169,063		5,47		
HC ₁₁ N		11,9191	8352,981	1904,2522	1565,3652			8352,981
C ₅ H ₅ CN		11,2788	8235,592	1902,0748	1559,6472			8235,592
C ₅ H ₅ CN		11,8663						
C ₁₁ H ₇ N		11,8482						
C ₁₁ H ₇ N	103501	12,2380	5655,27	1546,875	1214,405	4,5152		5655,27
C ₆ H ₅ CN	65514	11,9643	19820,19	2909,594	2573,225	3,23	2,34	19820,19
HCCCH ₂ CN	77507	11,3874	46226	1472,154	1426,249	3,82		46226
H ₃ C ₅ N	77506	11,0719	39931	1376,633	1329,747	5,09		39931
H ₃ C ₅ N	77508	11,3032	7098,2	2682,6	1943,54	2,382	3,527	7098,2

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
H ₃ C ₅ N	97501	11,6628		587,264		6,5		
C ₈ H	97502	10,3222		583,34		10,4		
C ₈ H-	73502	11,0752		1376,863		8,2		
C ₆ H-	49509	10,9294		4654,944		6,19		
C ₄ H-	75503	13,8256		1331,3327		4,33		
H ₂ CCO	42501	12,7118	282089	10293,321	9915,904	1,42215		
CN	26504	12,8899		56693,47		1,45		
HCN	27501	12,6201		44315,976		2,9852		
HC ₇ O	101501	11,8921		549,2054		2,17		
HC ₅ O	77512	12,2304		1293,6046		2,16		
H ₂ CN	28502	11,1761	284343	39158,3	34245,5	2,54		
H ₂ CO	30501	13,0792	281970,56	38833,987	34004,244	2,3317		
HC ₃ O ⁺	53522	11,3222		4460,588		3,26		
HCO ⁺	29507	11,6021		44594,43		3,9		
H ₂ COH ⁺	31504	11,4710	197581,56	34350,55	29172,65	1,387	1,776	
H ₂ NCO ⁺	44516	10,6021	319800	10278,68	9948,9	4,13		
HCNO	43509	10,8451		11469,05		3,099		
HOCN	43510	11,0414	681000	10577,01	10398,49	3,7		

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
C ₄ O		11,0792						
t-HCOOH	46506	12,1461	77512,24	12055,021	10416,2	1,421	0,21	
c-HCOOH	46507		86461,62	11689,1	10284,07	2,65	2,71	
HC ₂ O	41506	12,0000		10831,37		1,59		
HC ₃ O	53507	11,3010	261120	4577,477	4489,053	2,39		
HC ₄ O	65513	11,4710		2279,914		1,73		
1-H ₂ C ₃ O	54503	11,0414	153430,9	4387,33	4257,84	2,156	0,7914	
c-H ₂ C ₃ O	54504	11,6021	32040,686	7825,001	6280,728	4,39		
CH	13502	14,1461		425476,4		1,46		
CNCN		11,9542						
NCCNH ⁺	53519	10,9345		4438,011		6,45		
C ₆ H ₂	74502	10,3284	268400	1348,0891	1341,3519	6,2		
CH ₃ CHCH ₂	42516	13,6021	46280,29	9305,243	8134,227	0,36	0,05	
CH ₂ C ₂ HCN	65506	11,6532	25981	2689,27	2474,82	4,07	1,33	
HCN	27501	12,3892		44315,976		2,9852		
C ₉ H ₈	116502	12,9823	3775,037	1580,864	1122,248	0,5	0,37	
CH ₂ CHCCH	52506	13,0792	50300,1	4744,943	4329,773	0,43		
HCCNC	51531	11,6435		4967,838			2,93	

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
CH ₃ CH ₂ CN	56506	11,0414	27342,643	4598,041	4133,767	3,816	1,235	
C ₃ N ⁻	50514	11,1139		4851,622		3,1		
C ₅ N ⁻	74513	11,4149		1388,86		5,23		
HC ₅ NH ⁺	76524	11,8751		1295,816		3,26		
NC ₄ NH ⁺	77515	10,9542		1293,908		9,1		
HCCNCH ⁺	52539	10,4771		4664,432		3,45		
CH ₂ CCCN	64509	11,2041	288	2195,08	2177,78	4,43		
o-CH ₂ C ₃ N		11,0414						
P-CH ₂ C ₃ N		10,6627						
CH ₂ C ₃ N		-11,3010						
C ₅ H ⁺		10,9444						
C ₃ H ⁺	37505	10,3802		11244,95		3		
C ₈ H ₆	102502	12,4771	5680,35	1529,741	1204,956	0,656		
NCCCNC	76501	-12		1409,9753		1,11		
HDCS	47504	12,079	202715	16110,81	14891,55	1,6536		
C ₂ H ₅ OH	46524	12,0414	34921	9351,2	8138,5	0,05	1,44	
CH ₃ COCH ₃		11,1461						
C ₂ H ₅ CHO	58505	11,2787	16669,63	5893,504	4598,982	1,71	1,85	

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
c-C ₅ H ₄ CCH ₂	90502	12,4313	8180,7	1886,672	1549,114	0,69		
c-C ₅ H ₄ CH ₂	78509	-12,544	8186,18	3802,743	2596,44	0,4326		
CCH	25501	14,8129		43674,52		0,77		
C ₂ H ₄		13,4771						
CH ₃ CH ₃		12,4771						
H ₂ C ₃	38501	12,2787	28775	10588,64	10203,97	4,1		
CH ₂ CCH	39505	13,9395	288055	9523,36	9207,2	0,14		
H ₂ C ₄	50503	12,5185	286234	4503,31	4428,61	4,1		
c-C ₅ H	61505	10,9542		2395,127		4,88		
c-C ₃ HCCH	62532	11,4913	34638,81	3424,88	3113,64	2,04	2,89	
H ₂ C ₅	62501	10,2552	277600	2304,7845	2285,805	5,9		
CH ₂ CCHCCH	64520	13,0792	19076,7	2859,29	2520,86		0,516	
c-C ₅ H ₆	66520	12,0792	8426,111	8225,639	4271,438		0,416	
o-C ₆ H ₄	76503	11,6989	6989,73	5706,8	3140,37		1,38	
c-C ₆ H ₆		12,7781						
C ₇ H	85501	10,8129		875,484		5,945		
CH ₂ CCHC ₄ H	88503	11,0792	15655,72	947,907	898,299	1,2	0,2	
1-c-C ₅ H ₅ CCH		12,1461						

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
2-c-C ₅ H ₅ CCH		12,301						
c-C ₆ H ₅ CH ₃		-12,7781						
c-C ₆ H ₅ CHCH ₂		-14						
C ₉ H	109501	-10,544		413,237		7		
c-C ₁₀ H ₈	128501	12,544	2841,953	1254,843	870,713	0,799		
C ₆ H ₅ CCH	102502	-12,3979	5680,35	1529,741	1204,956	0,656		
c-C ₂ H ₄ NH	43502	-13,29	22736,193	21192,461	13383,164		0,97	1,357
CHNCH ₂		-12,1	35615,5986	22224,02407	15064,61324			
c-C ₂ H ₄ S	60509	-12,33	21973,631	10824,925	8026,218	1,84		
c-C ₂ H ₄ O	44504	-12,26	25483,863	22120,873	14097,826		1,9	
c-C ₄ H ₄ NH	67504	-12	9130,632	9001,364	4532,11	1,74		
C ₄ H ₄ S		-13,1	8041,594988	5418,264739	3235,779214			
c-C ₄ H ₄ O	68507	-12,93	9447,123	9246,744	4670,825	0,661		
C ₃ H ₄ N ₂		-12,09	9725,30651	9373,99203	4771,91599			
C ₃ H ₃ NS		-12,85	8529,447921	5505,77936	3344,301222			
c-C ₅ H ₅ N	79501	-11,88	6039,249	5804,909	2959,21	2,215		
C ₄ H ₄ N ₂		-12,33	6242,943	5961,085	3048,706			
C ₄ H ₄ N ₂		-12,62	6276,82775	6067,165761	3084,44922			

Fórmula	Tag	Columna de densidad (log10 cm ⁻²)	A (MHz)	B (MHz)	C (MHz)	μ _a (D)	μ _b (D)	μ _c (D)
C ₈ H ₇ N		-11,6	3877,8396	1636,046217	1150,900598			
C ₈ H ₆ S		-13,11	3153,78424	1309,71106	925,5185			
C ₈ H ₆ O		-12,01	3916,5648	1660,7946	1166,4178			
C ₉ H ₇ N		-12,38	3145,533013	1271,577972	905,739406			
C ₈ H ₆ N ₂		-13,06						
C ₈ H ₆ N ₂		-10,61						
C ₈ H ₆ N ₂		-10,94						

Tabla A2. Datos espectroscópicos moléculas TMC-1.

Fórmula	Nombre	Tag	Q (300.0)	Q (225.0)	Q (150.0)	Q (75.0)	Q (37.50)	Q (18.75)	Q (9.375)	Q (5.000)	Q (2.725)
CH ₃ C ₆ H	1,3,7-Heptatriyne	88502	119072,9605	77324,9012	42082,6251	14876,7106	5272,0513	1923,214	745,2487	325,2694	155,4775
CH ₃ C ₄ H	1,3-Pentadiyne	64507	45530,2862	29566,5165	16091,1324	5688,9493	2016,5653	736,0174	285,5123	125,6155	59,878
CH ₃ C ₅ N	1-Cyano-2,4-pentadiyne	89501	358481,436	232795,1947	126695,1947	44788,4264	15871,0298	5786,7912	2241,2499	983,8953	467,0488
CH ₃ C ₃ N	Methylpropionitrile	65503	45017,1092	29233,2219	15909,7327	5624,8234	1993,6887	727,3156	282,0002	124,0342	59,0815
N ₂ H ⁺	Diazenylium	29506	1300,9224	931,5116	608,6347	304,9598	153,9969	78,5512	40,8884	23,4212	15,5633
NH ₃	Ammonia	17506	1766,345	1140,9791	623,7662	225,415	84,0965	34,7977	17,2096	12,535	12,0104
CH ₃ OH	Methanol	32504	37027,3292	20991,81	9750,0398	2924,3023	920,9637	274,988	78,17336	26,719	11,8899
C ₃ H	Propynylidynium	37505	556,6328	417,4838	278,3832	139,3329	69,8273	35,0801	17,7092	9,6057	5,3964
C ₃ H	Propynylidyne	37501	6291,9546	4622,9411	2961,053	1323,5198	547,0203	210,2569	84,0099		
C ₃ H	Propadienylidene	38501	9949,3796	6458,9772	3514,8039	1243,251	439,8252	150,8054	45,3947		
C ₃ H ₂	Cyclopropenylidene	38508	12764,2823	8289,7732	4513,2525	1597,8907	566,856	201,8429	72,4028	28,9244	12,0171
C ₃ H ₂	Propadienylidene	38501	9959,3796	6458,9772	3514,8039	1243,251	439,8252	150,8054	45,3947		
CH ₃ CCH	Propyne	40502	17826,072	9357,5685	4209,3392	1362,3596	481,7197	176,2982	68,7664		
CH ₃ CN	Acetonitrilo	41505	15130,4545	8130,4079	3811,9477	1267,6705	449,0811	164,3168	64,0955	28,4924	13,8355
HNCO	Isocyanic acid	43511	2695,3359	1742,4282	943,7057	331,9879	117,3039	42,8291	18,4492	9,8228	5,5129
CS	Carbon Monosulfide	44501	256,3136	191,9014	127,9818	64,1447	32,2373	16,2876	4,6029	2,6827	
CH ₃ CHO	Ethanal	45529	53426,1		12184,2						
HCS ⁺	Thiomethylum	45506	293,4689	2201526	146,8481	73,5856	36,9581	18,6478	9,496		
H ₂ CS	Thioformaldehyde	46509	5996,0476	3893,3025	2119,1954	750,1191	265,7132	91,2543	27,5538	8,7424	3,7879
SO	Sulfur Monoxide	48501	850,2167	632,2255	414,5014	197,5147	90,3536	38,8764	15,9033	7,4104	4,0346
C ₄ H	Butadiynyl	49503	15542,213	8848,3936	4095,2962	1420,6195	660,0312	329,7446	165,5392	88,9165	49,0805
C ₄ H ₂	Butatrienylidene	50503	23244,4913	15090,1929	8210,7728	2902,7544	1025,7567	351,1983	105,4252		

Fórmula	Nombre	Tag	Q (300.0)	Q (225.0)	Q (150.0)	Q (75.0)	Q (37.50)	Q (18.75)	Q (9.375)	Q (5.000)	Q (2.725)
C ₃ N	Cyanoethynyl	50511	14945,4583	10032,1148	5585,5305	2110,0444	955,612	475,8115	238,9055		
HNC ₃	Iminopropadienyldiene	51528	4019,4725	3014,5867	2009,8795	1005,3511	503,1541	252,0733	126,5389	67,9602	37,505
C ₃ O	Oxopropadienyldiene	52501	1300,2125	975,1423	650,1387	325,2023	162,7596	81,545	40,9399		
HC ₃ NH ⁺	Protonated Cyanoacetylene	52503	4334,2345	3250,6814	2167,2915	1084,0647	542,5128	271,753	136,3786		
HCCCHO	Propynal	54510	22987,6098	14901,4008	8095,6491	2857,5599	1010,1924	357,6532	126,904		
OCS	Carbonyl Sulfide	60503	1028,6544	771,4892	514,3807	257,3289	128,8244	64,5779	32,4567		
C ₅ H	Pentadiynylidyne	61505	19751,0455	14550,0788	9369,4793	4259,7283	1828,129	760,1392	338,6937	178,0943	98,5154
C ₃ S	Thiopropadienyldiene	68503	2163,7408	1622,7529	1081,8557	541,0492	270,6801	135,5043	67,9188	36,3801	19,9812
C ₆ H	Hexatriynyl	73501	79587,4706	57489,502	35681,9005	14787,7675	5599,3939	2014,6292	732,6493	325,3491	171,9242
HC ₃ N	Cyanoacetylene	51501	1374,9098	1031,1809	687,5086	343,8929	172,1063	86,2186	43,2767	23,2385	12,8204
HC ₄ NC	Isocyanobutadiyne	75501	13387,439	10040,3097	6693,5219	3347,0855	1673,9991	837,4891	419,2429	224,0641	122,5732
HC ₇ N	Cyanohexatriyne	99501	33256,0622	24941,308	16627,2126	8313,7765	4157,3059	2079,1325	1040,0615	555,1656	303,0213
HC ₆ NC	Isocyanohexatriyne	99504	32200,1036	24149,1315	16098,9571	8049,5801	4025,1906	2013,0706	1007,0295	537,5483	293,4198
HC ₉ N	Cyanooctatetrayne	123501	64558,7258	48418,0026	32277,9796	16138,9774	8069,8607	4035,3984	2018,1914	1076,8338	587,3296
HC ₁₁ N	Cyanodecapentayne	147501	36976,8168	27732,6126	18488,4084	9244,1341	4622,1729	2311,2379	1155,7819		
C ₆ H ₅ CN	Benzonitrile	103501	1075880,204	698715,4828	380279,663	134440,6577	47539,5151	16815,616 4	5951,2363	2322,1453	937,3381
HCCCH ₂ CN	Propargyl Cyanide	65514	217480,3995	140996,7859	76605,9327	27037,0993	9554,4398	3379,9531	1197,3107	468,0865	189,6262
C ₈ H	Octatetraynyl	97501	814128992	60175,214	38992,1292	18005,0283	7869,6366	3275,2896	1407,8848		
C ₈ H ⁺	Octatetraynylide ion	97502	10717,8486	80,8,1533	5358,6689	2679,3957	1339,8382	670,0792	335,2048		
C ₆ H ⁺	Hexatriynylide ion	73502	4541,3128	3405,8862	2270,5807	1135,3965	567,8499	285,088	142,2101		
C ₄ H ⁺	Butadiynylide ion	49509	1343,656	1007,7399	671,8809	336,0788	168,1992	84,2651	42,3		
H ₂ CCO	Ethenone	42501	10354,5331	6722,6934	3658,5256	1294,1016	457,8772	157,3867	47,6347	14,9078	6,2126
CN	Cyanogen	26504	664,0904	498,4498	332,9077	167,4335	84,7308	43,4081	22,7963	13,2693	8,5178

Fórmula	Nombre	Tag	Q (300.0)	Q (225.0)	Q (150.0)	Q (75.0)	Q (37.50)	Q (18.75)	Q (9.375)	Q (5.000)	Q (2.725)
HCN	Hydrogen Cyanide	27501	453,4944	325,2472	213,0967	106,8116	53,9106	27,4718	14,2716	8,1453	5,03
HC ₇ O	Hexatriynylformyl	101501	86912,8177	64207,3025	41564,4463	19144,4986	8339,2643	3464,1408	1493,3659	765,1787	417,2258
HC ₅ O	Butadiynylformyl	77512	36908,9143	27267,3789	17652,8709	8133,2185	3544,9262	1474,232	636,7423	327,1914	179,2889
H ₂ CN	N-Methaniminyl	28502	17101,2962	11106,5132	6048,6137	2144,7082	764,3675	283,12	118,3487		
H ₂ CO	Formaldehyde	30501	283,0163	1872,6221	1019,9706	361,7195	128,6492	44,6812	13,8008	4,4832	2,0166
HC ₃ O ⁺	Ethynyloxomethylium	53522	1402,1553	1051,6166	701,1335	350,7058	175,5129	87,922	44,1284	23,6926	13,0679
HCO ⁺	Oxomethylium	29507	145,8285	106,5508	70,4887	35,3833	17,8601	9,1023	4,7298	2,7007	1,6691
H ₂ COH ⁺	Hydroxymethylium	31504	1975,4721	1283,1604	698,8572	247,7383	88,1185	31,5464	11,4423		
H ₂ NCO ⁺	N-protonated	44516	9712,0323	6306,1768	3432,1904	1214,1566	429,0428	145,1951	42,6428	13,6029	6,0872
HCNO	Fulminic acid	43509	545,5841	409,23	272,9036	136,605	68,4665	34,4005	17,3697	9,4247	5,298
HOCN	Cyanic acid	43510	3217,0765	2087,4819	1135,3539	401,374	142,1413	50,8159	20,1244	10,3038	5,7601
t-HCOOH	Formic cis trans conformer	46506	8901,5559	5779,9698	3145,8404	1112,914	394,2552	139,997	49,9396		
c-HCOOH	Formic acid, cis conformer	46507	8614,9881	5593,7108	3044,3589	1076,9647	381,5034	135,4602	48,3155		
HC ₃ O	trans-Proptnonyl	53507	51832,5975	33163,0913	17697,0608	6118,6184	2141,0708	754,3412	267,4002		
HC ₄ O	Butadiynonyl	65513	10985,914	8236,3427	5489,163	2744,158	1372,4689	686,8277	344,059	184,1137	100,9486
1-H ₂ C ₃ O	Propadienone	54503	65374,8365	42478,9722	23127,5278	8177,356	2890,7362	1021,5999	360,7321		
c-H ₂ C ₃ O	Cyclopropenone	54504	44220,6696	28716,2465	15629,6707	5527,8732	1965,712	693,6091	246,5439	96,9081	39,1677
CH	Methylidyne	13502	120,8417	91,364	61,9974	32,9221	18,8471	12,1999	8,9599	7,9666	7,7726
NCCNH ⁺	Protonated cyanogen	53519	1409,3153	1056,9809	704,7058	352,49	176,4046	88,3677	44,3512	23,8115	13,1326
C ₆ H ₂	Hexapentaenylidene	74502	77002,2797	50020,2431	27226,8996	9626,7289	3400,3418	1162,7194	348,0722		
C ₉ H ₈	Indene	116502	338635,6551	219934,0709	119710,4979	42325,1521	14967,0832	5294,0742	1873,5109	730,939	294,9724
CH ₃ CH ₂ CN	Ethyl cyanide	56506	38581,6719	25037,5616	13617,6743	4812,0461	1701,9844	602,6742	213,7954		
HC ₅ NH ⁺	Protonated pentadiynenitrile	76524	4825,2803	3618,8597	2412,5617	1206,3863	603,3446	301,8353	151,0837	80,7339	44,1526

Fórmula	Nombre	Tag	Q (300.0)	Q (225.0)	Q (150.0)	Q (75.0)	Q (37.50)	Q (18.75)	Q (9.375)	Q (5.000)	Q (2.725)
NC ₄ NH ⁺	Protonated butynedinitrile	77515	43491,9906	32618,0005	21745,1723	10873,5059	5438,1086	2720,5195	1361,7537	727,6717	397,9551
HCCNCH ⁺	Protonated isocyanoacetylene	52539	1340,8693	1005,6603	670,5013	335,3925	167,857	84,0942	42,2146	22,6721	12,5118
C ₃ H ⁺	Propynylidinium	37505	556,6328	418,4838	278,3832	139,3329	69,8273	35,0801	17,7092	9,6057	5,3964
HDCS	Thioformaldehyde	47504	3986,043	2587,9431	1408,4993	498,4358	176,7266	62,8727	22,5182	9,0839	4,2391
c-C ₃ HCCH	Ethynylcyclopropenylidene	62532	45664,2438	29648,6695	16133,5424	5703,5162	2017,406	714,1338	253,1446	99,0573	40,1905
c-C ₅ H ₆	Cyclopentadiene	66520	407659,1983	264757,1093	144112,6948	50966,0143	18034,0928	6387,3843	2266,5031	888,4592	361,6137
C ₇ H	Heptatriynylidyne	85501	53771,1718	39554,5601	25401,6646	11470,749	4885,1213	2030,1189	912,9499		
C ₉ H	Nonatetraynylidyne	109501	113559,3834	83457,0047	53501,7734	24058,3344	10201,8431	4244,8312	1922,8438		
c-C ₁₀ H ₈	Azulene	128501	497348,5716	323007,9405	175809,9623	62157,3611	21978,8584	7773,3813	2750,313	1072,6131	432,5633
C ₆ H ₅ CCH	Phenylacetylene	102502	1083804,864	703836,7632	383057,736	135420,8105	47885,8638	16938,048	5994,5175	2339,0003	944,1177
C ₂ H ₄ S	Thiirane	60509	80363,3153	52182,7218	28400,6307	10045,3913	3556,8565	1261,6815	449,0929	177,0165	72,8335
C ₂ H ₄ O	Oxirane	44504	39396,2392	25584,6609	13927,7869	4929,4803	1747,6072	621,4504	222,307	88,3952	36,9571
C ₄ H ₄ O	Furan	68507	17602,2671	112761,4258	61380,2397	21708,4616	7682,1319	2721,3628	965,9875		
c-C ₅ H ₅ N	Pyridine	79501	343525,5348	223505,2382	121702,5429	43035,8922	15223,9181	5389,0681	1910,1174		
CH ₂ CN	Cyanomethyl	40505	62063,7438	40294,0105	21927,8109	7756,25	2750,4324	1008,4643	412,6877		
CH ₂ CHCN	Vinyl cyanide	53515	144388,7445	75261,8534	32682,398	9982,9176	3481,9456	1232,852	437,4991		
C ₂ S	Thioethenylidene	56502	2844,224	2120,0922	1396,4031	673,888	314,7808	138,8729	56,463		
HCCNC	Ethynyl isocyanide	51531	3777,0647	2832,8234	1888,7324	944,7917	472,8781	236,9363	118,9708	63,9244	35,3062
HC ₅ N	Cyanodiacetylene	75503	14089,814	10567,0498	7044,6594	3522,6429	1761,7749	881,3763	441,1863	235,7671	128,9512
H ₃ C ₅ N	(E)-Cyanovinylacetylene	77507	266728,3571	173255,9962	94316,81	33352,3118	11795,6268	4172,9619	1477,1705	576,5764	232,8727
H ₃ C ₅ N	Cyanoethynylethene	77506	307079,7116	199509,4768	108632,2712	38422,7701	13590,1446	4807,8881	1701,8338	664,1809	268,185
H ₃ C ₅ N	(Z)-Cyanovinylacetylene	77508	434796,7746	281960,5019	153244,1401	54105,3113	19122,0143	6763,6192	2394,8095	935,3941	378,3024
HC ₂ O	Ketenyl	41506	2310,8901	1733,2944	1155,8362	578,5155	289,9077	145,6193	73,4839	39,8312	22,3495

Fórmula	Nombre	Tag	Q (300.0)	Q (225.0)	Q (150.0)	Q (75.0)	Q (37.50)	Q (18.75)	Q (9.375)	Q (5.000)	Q (2.725)
CH ₃ CHCH ₂	Propene	42516	118676,7674	77057,8496	41937,4534	14833,023	5252,1206	1863,1314	663,2677	261,5009	107,5486
CH ₂ C ₂ H ₃ CN	Butadienenitrile	65506	202184,6122	130953,551	71083,4629	25066,0944	8854,139	3131,4701	1109,0872		
CH ₂ CHCCH	Butenyne	52506	27411,299	17780,4512	9666,2474	3414,2542	1207,3757	427,5098	151,6822		
C ₃ N ⁻	Cyanoethynylide ion	50514	3867,6977	2900,7591	1933,9967	967,4108	484,1843	242,5885	121,79663		
C ₅ N ⁻	Cyanobutadiynylide ion	74513	4502,0711	3376,46	2250,9667	1125,5912	562,9476	281,637	140,9846		
CH ₂ CCCN	Cyanopropynyl	64509	283909,5968	184326,6258	100296,0537	3545,7245	12552,3421	4591,7641	1869,3711		
C ₈ H ₆	Ethynylbenzene	102502	1083804,864	703836,7632	383057,736	135420,8105	47885,8638	16938,048	5994,5175	2339,0003	944,1177
NCCCNC	2-isocyano-3-propynenitrile	76501	4271,2306	3285,1332	2214,2604	1108,7394	554,5224	277,4243	138,8782		
C ₂ H ₅ OH	Ethanol	46524	45601,2912	33853,1288	28439,8942	19556,0038	9684,0548	1072,4178	95,597	37,5425	15,4354
C ₂ H ₅ CHO	Propanal	58505	82649,4356	53653,0617	29191,4761	10319,3299	3650,8629	1293,1024	458,9411	179,9812	73,3134
c-C ₅ H ₄ CCH ₂	Fulvenallene	90502	1434829,181	931724,7743	507058,8721	179254,0535	63388,3558	22424,4938	7938,4895	3099,1047	1252,0828
c-C ₅ H ₄ CH ₂	Fulvene	78509	780116,2457	506638,7303	275758,4162	97505,9932	34489,7962	12207,0053	4325,3342	1691,2414	685,2319
CCH	Ethynyl	25501	815,4124	520,4986	304,507	144,7246	72,9146	37,1437	19,2835	10,9935	6,7735
H ₂ C ₃	Propadienylidene	38501	9949,3796	6458,9772	3514,8039	1243,251	438,8252	150,8054	45,3947		
CH ₂ CCH	Propargyl	39505	44220,0314	28709,2505	15623,2625	5525,8952	1959,4212	718,9952	294,551		
H ₂ C ₄	Butatrienylidene	50503	23244,4913	15090,1929	8210,7728	2902,7544	1025,7567	351,1983	105,4252		
c-C ₅ H	Pentadiynylidyne	61505	19751,0455	14550,0788	9369,4793	4259,7283	1828,129	760,1392	338,6937	178,0943	98,5154
H ₂ C ₅	Pentatetraenylidene	62501	45839,1236	29771,1369	16204,7905	5729,9857	2024,777	694,9773	209,8012	64,829	26,2058
CH ₂ CCHCCH	1,4-Pentadiyne	64520	301341,3615	195347,4268	106126,0279	37452,5225	13234,3958	4681,5692	1658,3036	648,2572	262,5773
o-C ₆ H ₄	o-Benzynes	76503	313341,0336	203499,3259	110763,4304	39166,8312	13855,6009	4905,0509	1738,8258		
CH ₂ CCHC ₄ H	Propadienylbutadiyne	88503	246972,5821	159294,4149	86082,5492	30214,7821	10645,6238	3758,4096	1328,6156	517,9618	208,8537
c-C ₂ H ₄ NH	Aziridine	43502	32796,8719	21299,2404	11594,6815	4103,5443	1454,5679	517,0732	184,8428		

c-C ₄ H ₄ NH	Pyrrrole	67504	181678,7775	118011,8631	64238,8916	22719,4197	8039,6964	2847,8677	1010,773		
------------------------------------	----------	-------	-------------	-------------	------------	------------	-----------	-----------	----------	--	--

Fórmula	Energía B3LYP/6-311G**	Parámetro de asimetría de Ray	Fórmula	Energía B3LYP/6-311G**	Parámetro de asimetría de Ray
CH ₃ C ₆ H		-1	HC ₃ N		#¡DIV/0!
CH ₃ C ₄ H		-1	HC ₄ NC		#¡DIV/0!
CH ₃ C ₅ N		-1	HC ₇ N		#¡DIV/0!
CH ₃ C ₃ N		-1	HC ₆ NC		#¡DIV/0!
N ₂ H ⁺	-109,752619	#¡DIV/0!	HC ₉ N		#¡DIV/0!
NH ₃	-56,576036	-0,714235003	HC ₁₁ N		#¡DIV/0!
CH ₃ OH	-115,723965	-0,982032148	C ₆ H ₅ CN	-324,573245	-0,85026791
C ₃ H		#¡DIV/0!	HCCCH ₂ CN		-0,960993833
C ₃ H		#¡DIV/0!	C ₈ H		#¡DIV/0!
C ₃ H		-0,995414622	C ₈ H-		#¡DIV/0!
C ₃ H ₂		0,686039944	C ₆ H-		#¡DIV/0!
C ₃ H ₂		-0,997238263	C ₄ H-		#¡DIV/0!
CH ₃ CCH	-116,690771	-1	H ₂ CCO	-152,647754	-0,99722664
CH ₃ CN	-132,79333	-1	CN	-92,736868	#¡DIV/0!
HNCO	-168,732966	-0,999643316	HCN	-93,452019	#¡DIV/0!
CS	-436,246155	#¡DIV/0!	HC ₇ O		#¡DIV/0!
CH ₃ CHO	-153,876859	-0,944703692	HC ₅ O		#¡DIV/0!
HCS ⁺	-436,557434	#¡DIV/0!	H ₂ CN	-94,009283	-0,960712922
H ₂ CS	-437,501981	-0,992388035	H ₂ CO	-114,536341	-1,242938295
SO	-473,399366	#¡DIV/0!	HC ₃ O ⁺		#¡DIV/0!
C ₄ H		#¡DIV/0!	HCO ⁺	-113,58118	#¡DIV/0!
C ₄ H ₂		-0,999469847	H ₂ COH ⁺	-114,821049	-0,938508004
C ₃ N		#¡DIV/0!	H ₂ NCO ⁺		-0,997871365
HNC ₃		#¡DIV/0!	HCNO	-168,622603	#¡DIV/0!
C ₃ O	-189,44663	#¡DIV/0!	HOCN	-168,68748	#¡VALOR!
HC ₃ NH ⁺		#¡DIV/0!	t-HCOOH	-189,775289	-0,951149993
HCCCHO		-0,989718223	c-HCOOH	-189,775289	-0,963111704
OCS	-511,595272	#¡DIV/0!	HC ₃ O		-0,999310886
C ₅ H		#¡DIV/0!	HC ₄ O		#¡DIV/0!
C ₃ S		#¡DIV/0!	1-H ₂ C ₃ O		-0,998263896
C ₆ H		#¡DIV/0!	c-H ₂ C ₃ O		-0,880102832

Fórmula	Energía B3LYP/6-311G**	Parámetro de asimetría de Ray	Fórmula	Energía B3LYP/6-311G**	Parámetro de asimetría de Ray
CH	-38,491977	#jDIV/0!	H ₃ C ₅ N		-0,997570752
NCCNH ⁺		#jDIV/0!	H ₃ C ₅ N		-0,713245878
C ₆ H ₂		-0,999949545	HC ₂ O	-151,94461	#jDIV/0!
HCN	-93,452019	#jDIV/0!	CH ₃ CHCH ₂	-117,944051	-0,938603572
C ₉ H ₈	-347,847421	-0,654238615	CH ₂ C ₂ HCN		-0,981753735
CH ₃ CH ₂ CN	-172,117487	-0,959991686	CH ₂ CHCCH	-154,780801	-0,981937479
HC ₅ NH ⁺		#jDIV/0!	C ₃ N ⁻		#jDIV/0!
NC ₄ NH ⁺		#jDIV/0!	C ₅ N ⁻		#jDIV/0!
HCCNCH ⁺		#jDIV/0!	CH ₂ CCCN		-1,01830901
C ₃ H ⁺		#jDIV/0!	C ₈ H ₆	-308,473532	-0,854857472
HDCS		-0,987016957	NCCCNC		#jDIV/0!
c-C ₃ HCCH		-0,980254508	C ₂ H ₅ OH	-155,088116	-0,909440866
c-C ₅ H ₆	-194,15372	0,903496112	C ₂ H ₅ CHO	-193,202768	-0,785509113
C ₇ H		#jDIV/0!	c-C ₅ H ₄ CCH ₂		-0,898196902
C ₉ H		#jDIV/0!	c-C ₅ H ₄ CH ₂	-235,252005	-0,568386723
c-C ₁₀ H ₈	-385,930868	-0,61026562	CCH	-76,629459	#jDIV/0!
C ₆ H ₅ CCH	-308,473532	-0,854857472	H ₂ C ₃		-0,997281281
C ₂ H ₄ S	-476,832781	-0,59867726	CH ₂ CCH	-116,037356	-0,997732383
C ₂ H ₄ O	-153,830128	0,409278224	H ₂ C ₄		-0,999469847
C ₄ H ₄ O	-230,083412	0,916094431	c-C ₅ H		#jDIV/0!
c-C ₅ H ₅ N	-248,346873	0,847833096	H ₂ C ₅		-0,999862125
CH ₂ CN		-0,997313966	CH ₂ CCHCCH		-0,959237345
CH ₂ CHCN	-170,87964	-0,979827125	o-C ₆ H ₄	-230,969401	0,333432051
C ₂ S		#jDIV/0!	CH ₂ CCHC ₄ H		-0,993276874
HCCNC		#jDIV/0!	c-C ₂ H ₄ NH	-133,959526	0,669896886
HC ₅ N		#jDIV/0!	c-C ₄ H ₄ NH	-210,22608	0,943778458
H ₃ C ₅ N		-0,997950658			