UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE QUÍMICA

PREDICCIÓN DE ENERGÍAS MOLECULARES MEDIANTE MACHINE LEARNING

T E S I S

QUE PARA OBTENER EL TÍTULO DE

QUÍMICO

P R E S E N T A

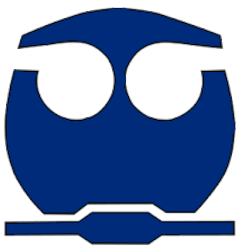
ISAI NEFTALI RODRÍGUEZ ROJAS

TUTOR

DR. JORGE MARTÍN DEL CAMPO RAMÍREZ

SUPERVISOR TÉCNICO

M. EN C. JUAN FELIPE HUAN LEW YEE



CIUDAD UNIVERSITARIA, CDMX, 2021

JURADO ASIGNADO

PRESIDENTE:	Dr. Álvarez Idaboy Juan Raul
VOCAL:	Dr. Flores Leonar Martha Magdalena
SECRETARIO:	Dr. Jorge Martín del Campo Ramírez
1er. SUPLENTE:	Dr. Dominguez Dueñas Laura
2° SUPLENTE:	M. en C. Lew Yee Juan Felipe Huan

SITIO DONDE SE DESARROLLÓ EL TEMA:

Departamento de Física y Química Teórica, Facultad de Química, UNAM

ASESOR DEL TEMA:

_____

Dr. Jorge Martín del Campo Ramírez

SUPERVISOR TÉCNICO:

_____

M. en C. Juan Felipe Huan Lew Yee

SUSTENTANTE:

_____

Isai Neftali Rodríguez Rojas

Agradecimientos

- A mi madre por su apoyo incondicional
- Al Dr. Jorge Martín del Campo Ramírez, por su confianza y las muchas oportunidades que me brindó para seguir aprendiendo y superarme.
- Al asesor técnico de este trabajo el M. en C. Juan Felipe Huan Lew Yee, por su amistad, diligencia y paciencia.
- A mi amigo y colega Alfonso Esqueda por su motivación y apoyo.
- A los miembros del jurado por sus sugerencias y tiempo para mejorar mi trabajo.

Resumen

En este trabajo se estudia el efecto de la representación molecular que ofrecen las matrices de Coulomb, y los *fingerprints* ACSF, SOAP y MBTR, sobre el aprendizaje de un modelo de regresión tipo kerner ridge (KRR por sus siglas en inglés). Mientras se predice la energía electrónica, el desempeño de cada descriptor es evaluado comparando la exactitud en la predicción que realiza el modelo. Para entrenar el algoritmo, se crearon tres bases de datos distintas, la primera, compuesta solamente por isómeros del pentadecano ($C_{15}H_{32}$), la segunda con isómeros de los primeros 15 alcanos, y la tercera con moléculas variadas provenientes de ChEMBL. Se encontró que el error absoluto es significativamente menor usando las matrices de Coulomb como descriptores para moléculas variadas, mientras que el descriptor ACSF ofrece un menor error en la predicción de isómeros de la misma molécula. Al final es posible notar que la métrica que mejor describe la energía son las matrices de Coulomb, aún cuando estas no cumplen con todas las características de un buen descriptor.

Abstract

In this work, the effect of the molecular representation offered by the Coulomb matrices, and the fingerprints ACSF, SOAP, and MBTR, on the learning of a ridge regression (KRR) model is studied. While predicting electronic energy, the performance of each descriptor is evaluated by comparing the accuracy of the result made by the model. To train the algorithm, three different databases were created. The first, composed only of isomers of pentadecane ($C_{15}H_{32}$), the second with isomers of the first 15 alkanes, and the third with diverse molecules from ChEMBL. It was found that the absolute error is significantly lower using Coulomb matrices as descriptors for diverse molecules, and homologous molecules, while the ACSF descriptor offers a lower error in the prediction of isomers of the same molecule. Finally, it is possible to note that the metric which best describes the energy is the Coulomb matrix, even though it doesn't meet all the characteristics of a good descriptor.

Índice

Índice de figuras

CAPÍTULO 1

Introducción

El aprendizaje de máquina, mejor conocido como *Machine Learning* (ML, por sus siglas en inglés), es una rama de la inteligencia artificial encargada de desarrollar algoritmos para que las computadoras sean capaces de aprender tendencias y comportamientos para poder realizar predicciones a partir de un conjunto de datos. Recientemente, los métodos de ML en conjunto con los métodos de *Deep Learning* (DL, por sus siglas en inglés) han generado especial interés por su efectividad para realizar predicciones en una gran cantidad de áreas, y se han desarrollado herramientas que facilitan su uso, como scikit-learn [1], keras [2], tensorflow[3], RDKit[4], DScribe[5] y DeepChem[6]. En el ámbito de la química, los métodos de ML y DL han sido usados de forma complementaria a los métodos de estructura electrónica tradicionales ayudando en la construcción de la función de onda [7, 8, 9, 10] o el diseño de funcionales de intercambio y correlación. [11] Por otra parte, también se han convertido en una alternativa completa para realizar predicciones rápidas y confiables de datos termodinámicos, propiedades moleculares,[12, 13, 14] interacciones atómicas, [15] optimización de estructuras, [16] así como de la reactividad química en general,[17, 18] acelerando el estudio y diseño de nuevos materiales[19] y fármacos.[20, 21, 22] En este ámbito es de particular interés la energía, pues sirve para determinar la termodinámica de una reacción y por lo tanto proporciona información sobre la reactividad química. Sin embargo, los métodos *ab initio* tradicionales son sustancialmente lentos y demandantes, lo que dificulta analizar moléculas con un gran número de átomos. Uno de los estudios que con frecuencia se abordan es el de la superficie de energía potencial (*PES* por sus siglas en inglés), en el que se asume que existe una relación entre las cargas nucleares, las posiciones atómicas y la energía. Tradicionalmente, esta relación suele ser descrita por la ecuación Schrödinger, sin embargo resolverla se vuelve cada vez más complicado conforme aumenta el tamaño del sistema. Gracias a la capacidad que tienen los algoritmos de ML de inferir una función basándose en un conjunto de datos, es posible encontrar una relación entre la carga, la posición y la energía lo suficientemente buena como para suplir la ecuación de Schrödinger sin las complicaciones que esta conlleva.

Esta idea, de manera similar puede ser usada para describir y relacionar otras propiedades químicas usando algoritmos de ML, en donde el objetivo principal sea establecer una relación entre la estructura atómica del sistema de estudio y sus propiedades, es decir una relación estructura-propiedad. [23, 24]

En ML se utilizan diversos tipos de modelos, y el aprendizaje se lleva a cabo al modificar sus parámetros para predecir correctamente un conjunto de datos de entrenamiento. El uso de estos métodos presenta tres grandes retos, el primero es que no existe un modelo universal que relacione de forma perfecta la estructura atómica con alguna propiedad molecular, por lo que en general se deben probar muchos modelos o construir nuevos para encontrar el que brinde el mejor desempeño para el problema de interés.[25] El segundo reto consiste en encontrar un conjunto de entrenamiento correcto que contenga suficiente información sobre el comportamiento que se desea predecir. El tercer reto consiste en encontrar la representación adecuada para los datos con los cuales interactúa el modelo, y que permita extraer las características que determinan el comportamiento de la propiedad a predecir.[26, 27] Por ello resulta de especial interés explorar cómo afecta la selección del conjunto de entrenamiento y la representación del espacio químico.

La selección de las moléculas que integran el conjunto de entrenamiento depende de conocer sus características químicas y de la información disponible para escoger aquellas que contengan las principales tendencias del comportamiento químico. Por otra parte, la forma en como se representa la información estructural es trascendental para que el modelo pueda extraer la información del conjunto de entrenamiento que usará para aprender. Para realizar la representación de una molécula se utilizan descriptores moleculares que codifican la información estructural en un vector numérico, siendo a la vez estos parte de un campo más grande conocido como ingeniería de características o *feature engi neering* cuyo principal objetivo es preparar los datos para que sean compatibles y mejoren el desempeño del modelo. [28] Algunas de las cualidades más importantes con las que debe contar un buen descriptor son [29]:

- La representación debe ser traslacional y rotacionalmente invariante.
- Las propiedades deben ser invariantes a permutaciónes en el orden atómico.
- El descriptor debe corresponder a una propiedad y su generación debe ser única.
- Emplear el descriptor debe ser más rápido que el cálculo directo de la propiedad.

Un caso destacable de descriptores moleculares son las huellas químicas o *fingerprints*, los cuales se basan en ideas como la asignación de un identificador numérico a cada átomo en una molécula, y contar la ocurrencia del tipo de átomos alrededor de una distancia fija o dentro de un radio circular.[30] Este tipo de representación permite extraer características que

reflejan la ausencia o presencia de subestructuras dentro de la molécula. [31] Algunos ejemplos son los *circular fingerprints*,[32] *MACCS keys*[33] y los *Extended-connectivity fingerprints*,[30] usados comúnmente en quimioinformática.[34] Particularmente, algunos descriptores de reciente creación considerados específicamente para su uso en ML y DL son las matrices de Coulomb, [24] *Atom-centered Symmetry Functions*[28] (ACSF, por sus siglas en inglés), *Smooth Overlap of Atomic Positions* (SOAP, por sus siglas en inglés)[27] y *Many-body Tensor Representation*[26] (MBTR, por sus siglas en inglés).

En este trabajo se estudia el proceso de construcción y entrenamiento de un modelo de ML que emplee información estructural para la predicción de energías calculadas mediante un método *ab initio*. Se utilizarán conjuntos de entrenamiento compuestos por moléculas de diversos tipos para evaluar la influencia de su composición sobre la capacidad del modelo. Cada conjunto de entrenamiento contiene como datos de entrada una representación de la información estructural de la molécula, y como datos de salida los resultados a predecir. También se explorará el uso de varios descriptores para la codificación de la información estructural, particularmente las matrices de Coulomb, ACSF, SOAP y MBTR . El funcionamiento del modelo se evaluará a partir de su capacidad para predecir la información de moléculas no vistas durante su construcción y entrenamiento.

CAPÍTULO 2

Marco Teórico

Los distintos métodos de ML se pueden clasificar dependiendo del problema que se busque resolver. Una de estas clasificaciones es conocida como aprendizaje predictivo o supervisado, cuyo objetivo es aprender una función que relacione los datos de entrada o *inputs* (x) con los datos de salida o *outputs* (y). Ambos, parte de dos conjuntos distintos, el primero es con el que se entrena el modelo (*Training set*), donde x e y son conocidos, mientras que en el segundo, llamado conjunto de prueba (*Test set*) sólo se conoce x y se desea predecir y.

Visto de otra forma se puede asumir que el problema a resolver queda descrito por una función desconocida $y = F(x)$ que se busca aproximar a partir de un conjunto de datos para posteriormente hacer predicciones. En el caso más sencillo el *input* x es un vector numérico que contiene la representación de alguna característica o atributo de un objeto, por ejemplo los píxeles de una imagen o los caracteres de una palabra. Por otra parte el *output* o respuesta es en la mayoría de los casos una variable categórica o nominal que constituye lo que se desea predecir. Si la respuesta es categórica, es decir que $y \in \{1, ..., C\}$ donde C es el número de clases, el problema se conoce como clasificación, por el contrario si la respuesta es de valor real y continua, el problema se conoce como regresión. Al final, se busca realizar predicciones para *inputs* que no se incluyeron en el proceso de aprendizaje, es decir, que el algoritmo sea capaz de generalizar lo aprendido hacia nuevos casos. [35]

2.1. Regresión lineal

La regresión es otro acercamiento al aprendizaje supervisado que a pesar de existir desde hace varios años todavía sigue siendo muy usado, además de que es el punto de partida para algoritmos más complejos. [36] Una de las formas de regresión más recurridas es la regresión lineal, la cual se basa en asumir una relación como la mostrada en la ecuación 2.1 entre

los *outputs* y los *inputs*, al buscar la función que mejor interpole los datos contenidos en el conjunto de entrenamiento.

$$y(x) = f(x) = w \cdot x + b \tag{2.1}$$

Cuando se trata con más de una dimensión, la ecuación anterior representa geométricamente un hiperplano donde w es un parámetro que define una dirección perpendicular al hiperplano y b es el parámetro que mueve el hiperplano paralelo a sí mismo. En el ámbito del aprendizaje de máquina w y b se denominan peso y sesgo respectivamente. Una forma de encontrar la línea que mejor se ajuste a los datos es minimizar la distancia entre los puntos de entrenamiento o lo que es lo mismo, usar el criterio de mínimos cuadrados (L):

$$L(x, b) = \sum_{i=1}^{l} (y_i - w \cdot x - b)^2. \tag{2.2}$$

A esta función se le conoce como función de pérdida cuadrada (*square loss function*), porque mide la pérdida asociada con un grupo de parámetros a través de la suma de cuadrados. Al ser b un vector independiente con variables aleatorias distribuidas de forma idéntica, y un gran número de datos de entrada y salida, estos se pueden representar de forma vectorial o matricial y la función de pérdida puede ser expresada como:

$$L(\hat{w}) = (y - \hat{X}\hat{w})^T (y - \hat{X}\hat{w}) \tag{2.3}$$

donde T denota la transpuesta de una matriz, $\hat{w} = (w, b)$, y el vector columna que contiene las etiquetas y $\hat{X}$ la matriz con los datos de entrenamiento. [37]

Para poder minimizar esta perdida se puede diferenciar L respecto a $(\hat{w})$ e igualar a cero, [37]

$$\frac{\partial L}{\partial \hat{w}} = -2\hat{X}^T y + 2\hat{X}^T \hat{X}\hat{w} = 0 \tag{2.4}$$

$$\hat{X}^T \hat{X}\hat{w} = \hat{X}^T y \tag{2.5}$$

de donde esta última expresión (2.5) es conocida como ecuaciones normales. Bajo la premisa de que el inverso de $\hat{X}^T \hat{X}$ existe, la solución de la función de mínimos cuadrados se convierte en:

$$\hat{w} = (\hat{X}^T \hat{X})^{-1} \hat{X}^T y \tag{2.6}$$

2.2. Regresión de cresta

Mejor conocido en inglés como *ridge regresión* o *penalized least squares*, este método remplaza la función de peso 2.6 por:

$$\hat{w} = (\hat{X}^T\hat{X} + \lambda I_n)^{-1}\hat{X}^T y \tag{2.7}$$

la cual se obtiene al sumarle a la matriz $\hat{X}^T\hat{X}$ un múltiplo λ de la diagonal de la matriz identidad I_n, ofreciendo como ventaja una mayor estabilidad numérica respecto a la regresión lineal, por lo que ahora la nueva "función de pérdida penalizada", se puede definir como:

$$L(w, b) = \lambda(w \cdot w) + \sum_{i=1}^{l}(w \cdot x + b - y_i)^2 \tag{2.8}$$

en donde el término $\lambda(w \cdot w)$, que se agregó es conocido como penalización por contracción o *shrinkage penalty*. Dicha expresión se vuelve pequeña cuando los coeficientes w son cercanos a cero, por lo que λ sirve para controlar el impacto relativo que tiene este término en la regresión. Si $\lambda = 0$ la regresión simplemente será la función de mínimos cuadrados, pero si $\lambda \to \infty$ el impacto de la penalidad crece. [36]

2.3. Kernel

Muchas veces cuando el problema que se busca resolver no puede ser expresado como una mera combinación lineal de los atributos que se presentan, o dicho de otra forma no es posible aprender relaciones lineales entre los datos, es necesario utilizar características no lineales. Una opción es la función *Kernel* que toma los vectores *input* en el espacio original, y representa su producto punto en un espacio de dimensión superior. Si x, z son vectores del conjunto de entrenamiento X, y ϕ es una función que mapea X a un espacio dimensional superior, es decir $\phi : X \to R^N$, entonces la función *Kernel* se define como:

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle = \langle \phi(z) \cdot \phi(x) \rangle = K(z, x) \tag{2.9}$$

cuyas propiedades son que es simétrico y positivo semidefinido. Esta solución se hace evidente al observar que los datos de entrenamiento siempre aparecen en la forma del producto punto entre pares de ejemplos $(x^T x)$ y que al aplicar dicha transformación, los datos en esta nueva dimensión pueden ser linealmente separables. El producto punto de los vectores también puede ser representado como una matriz G por lo que la función *Kernel* así mismo se suele

expresar como una matriz.

$$G_{i,j} = K(x_i, z_j) \tag{2.10}$$

Existen varios tipos de *Kernel*, por ejemplo, un *Kernel* no lineal llamado también *Kernel* polinomial se define de la siguiente manera:

$$K(x, z) = (1 + \sum_{j=1}^{N} x_{ij} \cdot z_{ij})^d \tag{2.11}$$

donde d es el grado del polinomio. Otro tipo *Kernel* no lineal también muy conocido es el *Kernel* radial (*radial Kernel*) que adopta la forma:

$$K(x, z) = \exp(-\gamma \sum_{j=1}^{N} (x_{ij} - z_{ij})^2) \tag{2.12}$$

donde γ es una constante que modifica la varianza del modelo a través de la norma Euclidiana $\sum_{j=1}^{p} (x_{ij} - z_{ij})^2$, de manera que discrimina si los puntos de entrenamiento x juegan o no un papel importante para predecir z, al ser muy pequeño el valor $K(x, z) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - z_{ij})^2)$ cuando la norma es grande. [38]

2.4. Regresión Kernel Ridge

El método *ridge regression* puede ser modificado con el método Kernel si, tomando como base la ecuación(2.7), se remplaza $\hat{X}^T \hat{X}$ por la matriz K, y si se define la siguiente igualdad [36]:

$$\alpha \triangleq (K + \lambda I_n)^{-1} y \tag{2.13}$$

donde $\triangleq$ significa que alfa es igual por definición al resto de la expresión, hacer esto permite reescribir $\hat{w}$ de la siguiente forma:

$$\hat{w} = \hat{X}^T \alpha = \sum_{i=1}^{N} \alpha_i x_i \tag{2.14}$$

indicando que el vector solución (es decir la respuesta) es una suma lineal de los N vectores de entrenamiento, y por tanto partiendo de que la ecuación de regresión es $y = f(x) = wx$ la nueva función de regresión $g(x)$ queda definida como,

$$y \approx g(x) = \sum_{i=1}^{N} \alpha_i (x \cdot x_i^T) \tag{2.15}$$

$$\sum_{i=1}^{N} \alpha_i (x \cdot x_i^T) = \sum_{i=1}^{N} \alpha_i k(x, x_i) \tag{2.16}$$

$$g(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) \tag{2.17}$$

a este ajuste se le conoce en inglés como *kernel ridge regression* o KRR.

2.5. DScibe

DScribe es una librería que permite representar estructuras atómicas en distintos tipos de *fingerprints* o vectores que pueden ser empleados por ML, más no son dependientes de ningún modelo. Su principal ventaja es que ofrece descriptores de última generación, tanto globales enfocados en representar la información de toda la estructura, como locales, diseñados para codificar una región localizada en una estructura atómica . Entre las opciones que presenta se prestará especial atención a los siguientes cuatro descriptores. [5]

2.5.1. Matrices de Coulomb

Es un tipo de representación molecular que trata de imitar la interacción electrostática entre los núcleos de una molécula, usando las coordenadas cartesianas R_I y las cargas nucleares Z_I del i ésimo átomo.

$$M_{ij}^{\text{Coulomb}} = \begin{cases} 0.5 Z_i^{2.4} & \text{para } i = j \\ \frac{Z_i Z_j}{R_{ij}} & \text{para } i \neq j \end{cases} \tag{2.18}$$

En la matriz, los elementos de la diagonal codifican un ajuste polinómico de las energías atómicas a la carga nuclear, visto como la interacción de un átomo consigo mismo mientras que los elementos fuera de esta corresponden a la repulsión del i ésimo átomo y el j ésimo átomo. [24, 5]

2.5.2. Atom Centered Symmetry Functions (ACSFs)

Este descriptor codifica la configuración de los átomos vecinos alrededor del i-ésimo átomo usando funciones de simetría. Para detectar dichos vecinos, DScribe ofrece tres funciones de simetría de dos cuerpos,

$$G_i^{1,Z_1} = \sum_{j}^{|Z_1|} f_c(R_{ij}) \tag{2.19}$$

$$G_i^{2,Z_1} = \sum_j^{|Z_1|} e^{-\eta(R_{ij}-R_s)^2} f_c(r_{ij}) \tag{2.20}$$

$$G_i^{3,Z_1} = \sum_j^{|Z_1|} cos(\kappa R_{ij}) f_c(r_{ij}) \tag{2.21}$$

donde la suma corre para todos los átomos con número atómico Z_1. Los parámetros η, R_s y κ son modificables, mientras que $R_{ij} = |R_i - R_j|$ y f_c es una función de corte definida como,

$$f_c(r) = \frac{1}{2}\left[\cos\left(\pi\frac{r}{r_{cut}}\right) + 1\right] \tag{2.22}$$

donde r_{cut} es el radio de corte. Por otra parte para detectar motivos específicos, se pueden usar dos funciones de simetría de tres cuerpos, que toman en cuenta el ángulo entre tripletes de átomos dentro de cierto límite, así como la distancia entre estos. Las opciones que ofrece DScribe son,

$$G_i^{4,Z_1,Z_2} = 2^{1-\zeta} \sum_{j\neq i}^{|Z_1|} \sum_{k\neq i}^{|Z_2|} (1 + \lambda\cos\theta)^\zeta \cdot e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \tag{2.23}$$

y

$$G_i^{5,Z_1,Z_2} = 2^{1-\zeta} \sum_{j\neq i}^{|Z_1|} \sum_{k\neq i}^{|Z_2|} (1 + \lambda\cos\theta)^\zeta \cdot e^{-\eta(R_{ij}^2+R_{ik}^2)} f_c(R_{ij}) f_c(R_{ik}) \tag{2.24}$$

donde las sumas de j y k corren sobre todos los átomos con número atómico Z_1 o Z_2, así mismo ζ, λ y η son parámetros definidos por el usuario lo mismo que θ es el ángulo entre los tres átomos. Al final, el *fingerprint* de un átomo se construye concatenando el resultado de las diferentes funciones de simetría. [28, 5]

2.5.3. Smooth Overlap of Atomic Positions (SOAP)

Es un descriptor que codifica ambientes locales dentro de una estructura expandiendo la densidad atómica de cada átomo con armónicos esféricos y funciones de base radial. Para ello, primero se transforma cada especie dentro de la estructura, en campos de densidad atómica ρ^z con funciones gaussianas no normalizadas centradas en cada átomo,

$$\rho^z(r) = \sum_i^{|Z|} e^{-\frac{1}{2\sigma^2}|r-R_i|^2} \tag{2.25}$$

donde la suma corre para todos los átomos con número atómico Z y σ es el ancho de la gaussiana. La expansión de la densidad se hace al elegir $r = 0$ como el centro del punto de interés junto con un conjunto de funciones de base radial ortonormales y armónicos esféricos de acuerdo a la siguiente expresión,

$$\rho^z(r) = \sum_{min} C^Z_{nlm} g_n(r) Y_{lm}(\theta, \phi) \tag{2.26}$$

donde los coeficientes se pueden obtener a través de:

$$C^Z_{nlm} = \iiint_{R^3} dV g_n(r) Y_{lm}(\theta, \phi) \rho^z(r) \tag{2.27}$$

Los armónicos esféricos reales se definen de la siguiente forma,

$$Y_{lm}(\theta, \phi) = \begin{cases} \sqrt{2}(-1)^m Im[Y_l^{|m|}(\theta, \phi)] & \text{si } m < 0 \\ Y_l^0 & \text{si } m = 0 \\ \sqrt{2}(-1)^m Re[Y_l^m(\theta, \phi)] & \text{si } m > 0 \end{cases} \tag{2.28}$$

donde Y_l^m es el complejo ortonormalizado del armónico esférico establecido como,

$$Y_{lm}(\theta, \phi) = \pi \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(cos\theta) e^{im\phi} \tag{2.29}$$

y P_l^m son los polinomios asociados de Legendre.

Las funciones de base radial usadas son,

$$g_{nl}(r) = \sum_{n'=1}^{n_{max}} \beta_{nn'l} \phi_{n'l}(r) \tag{2.30}$$

$$\phi_{nl}(r) = r^l e^{-\alpha_{nl} r^2} \tag{2.31}$$

las cuales permiten soluciones analíticas hasta $l \leq 9$. Los parámetros de decaimiento α_n, se elijen de tal forma que cada función no ortonormalizada ϕ_{nl}, decaiga hasta un valor límite de 10^{-3}, en un radio de corte tomado en una cuadrícula uniformemente espaciada desde 1Å hasta r_{cut} con intervalos de $\frac{r_{cut}-1}{n_{max}}$ y donde r_{cut} controla el alcance máximo de cada base. Los pesos, representados por $\beta_{nn'l}$ se elijen de tal forma que las funciones de base radial sean ortonormales. Para cada valor de l, $\beta_{nn'l}$ se puede calcular con la ortogonalización de Löwdin:

$$\beta = S^{-1/2} \tag{2.32}$$

$$S_{nn'} = \langle \phi_{nl} | \phi_{n'l} \rangle = \int_0^\infty dr r^2 r^l e^{-\alpha_{nl} r^2} r^l e^{-\alpha_{n'l} r^2} \tag{2.33}$$

Al final el descriptor es un vector p rotacionalmente invariante donde cada elemento se define como,

$$p_{nn'l}^{Z_1,Z_2} = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{nlm}^{Z_1})^* c_{n'lm}^{Z_2} \tag{2.34}$$

El vector p se forma al concatenar cada $p_{nn'l}^{Z_1,Z_2}$ para todos los pares únicos de números atómicos Z_1, Z_2, junto con todos los pares únicos de funciones de base radial n, n' hasta n_{max} y los valores de grados angulares l hasta l_{max}.[27, 39, 5]

2.5.4. Many-body Tensor Representation (MBTR)

Codifica estructuras periódicas o moleculares usando la distribución de subestructuras de distintos tamaños, y agrupándolas a través de los elementos químicos que contienen. Para ello se usan funciones de geometría, g_k, que transforman una configuración de átomos k en un solo valor que lo represente. Las funciones de simetría disponibles en DScribe son: , $g_2(R_l, R_m)$ la distancia, $|R_l, R_m|$, el inverso de la distancia, $\frac{1}{|R_l, R_m|}$ el ángulo $g_2(R_l, R_m, R_n)$: $\angle(R_l - R_m, R_n - R_m)$ y el coseno del ángulo, $\cos(\angle(R_l - R_m, R_n - R_m))$.

Los escalares son ampliados usando estimación de la densidad kernel con un kernel gaussiano, dando la distribución D_k,

$$D_1^l(x) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x - g_1(Z_l))^2}{2\sigma_1^2}} \tag{2.35}$$

$$D_2^{l,m}(x) = \frac{1}{\sigma_2 \sqrt{2\pi}} e^{-\frac{(x - g_2(R_l, R_m))^2}{2\sigma_2^2}} \tag{2.36}$$

$$D_3^{l,m,n}(x) = \frac{1}{\sigma_3 \sqrt{2\pi}} e^{-\frac{(x - g_3(R_l, R_m, R_n))^2}{2\sigma_3^2}} \tag{2.37}$$

donde σ_k es la desviación estándar del kernel gaussiano y x corre sobre todo un rango predefinido de valores, que cubren los posibles valores para g_k. Posteriormente se hace una suma ponderada de las distribuciones D_k para cada posible combinación de las k especies presentes en los datos. Las distribuciones para k = 1,2,3 son las siguientes:

$$MBTR_1^{Z_1}(x) = \sum_l^{|Z_1|} w_1^l D_1^l(x) \tag{2.38}$$

$$MBTR_2^{Z_1,Z_2}(x) = \sum_{l}^{|Z_1|} \sum_{m}^{|Z_2|} w_2^{l,m} D_2^{l,m}(x) \tag{2.39}$$

$$MBTR_3^{Z_1,Z_2,Z_3}(x) = \sum_{l}^{|Z_1|} \sum_{l}^{|Z_2|} \sum_{m}^{|Z_3|} w_3^{l,m,n} D_3^{l,m,n}(x) \tag{2.40}$$

donde las sumatorias l, m y n corren para todos los átomos con los números atómicos Z_1, Z_2 y Z_3 respectivamente, mientras que w_k es un "peso", es decir una función que controla la importancia de los diferentes términos. Cuando $k = 1$ no se usan pesos y por tanto $w_1^l = 1$, en cambio cuando $k = 2$ y $k = 3$ las funciones de los pesos tienen la siguiente forma:

$$w_2^{l,m} = e^{-s_k|R_l - R_m|} \tag{2.41}$$

$$w_3^{l,m,n} = e^{-s_k(|R_l - R_m| + |R_m - R_n| + |R_l - R_n|)} \tag{2.42}$$

donde S_k sirve para ajustar la distancia de corte.[26, 5]

CAPÍTULO 3

Objetivos

El objetivo general de este trabajo es construir y entrenar un modelo de ML para la predicción de energía calculada mediante métodos *ab initio*. Una vez construido el modelo, se estudiará de manera particular:

- El efecto del conjunto de entrenamiento sobre la capacidad predictiva del modelo.
- El efecto que tienen distintas representaciones moleculares sobre el aprendizaje del modelo. En particular se probará la representación de las matrices de Coulomb y de los *fingerprints* ACSF, SOAP y MBTR.

CAPÍTULO 4

Metodología

Se usó *scikit-learn* para construir un modelo de regresión tipo KRR empleando como función de pérdida el error cuadrático medio y la función kernel de base radial. Los hiperparámetros λ y γ se calcularon entre doce puntos espaciados uniformemente dentro de una escala logarítmica de -12 a 12, los cuales se optimizan durante el entrenamiento y al final se escoge la combinación que proporcione los mejores resultados a través de un proceso conocido como validación cruzada.

El proceso de construcción de los conjuntos usados para el entrenamiento del modelo se realizó creando un *script* en python y se muestra en la Figura 4.1. El primer paso consiste en la elección de la fórmula de las moléculas que constituirán el conjunto. En el segundo paso se obtuvieron los SMILES correspondientes a las moléculas contenidas en PubChem [40] con la respectiva fórmula molecular deseada. En el tercer paso se realiza un mezclado aleatorio de cada SMILES [41, 42] para generar nuevos, que a su vez proporcionarán información de nuevas moléculas. Sin embargo, varios de los SMILES generados de esta manera no corresponderán a una molécula real, por lo que se aplica un filtro para conservar solamente los correctos y descartar los incorrectos. La representación de SMILES no es única, por lo que la lista puede contener varios que correspondan a la misma molécula. Para solucionar esto, cada SMILES se convirtió a InChI key [43] y se guardó en una lista, eliminando entradas repetidas. El InChI key sí es una representación única de cada molécula, por lo que se evita que existan moléculas repetidas en el conjunto de entrenamiento. Posteriormente, la molécula se construye a partir de cada InChI key, la geometría molecular se optimiza mediante el *Merck Molecular Force Field* [44] (MMFF) y se realiza un cálculo de punto simple empleando el funcional MN15-L [45] y la base STO-3G [46]. Finalmente, el resultado se integra en el conjunto de entrenamiento.

Para analizar el efecto del conjunto de entrenamiento sobre la capacidad predictiva del modelo se crearon tres conjuntos de entrenamiento, los cuales se muestran en la Tabla 4.1. En todos los casos la geometría corresponde a la optimización de MMFF y los resultados a

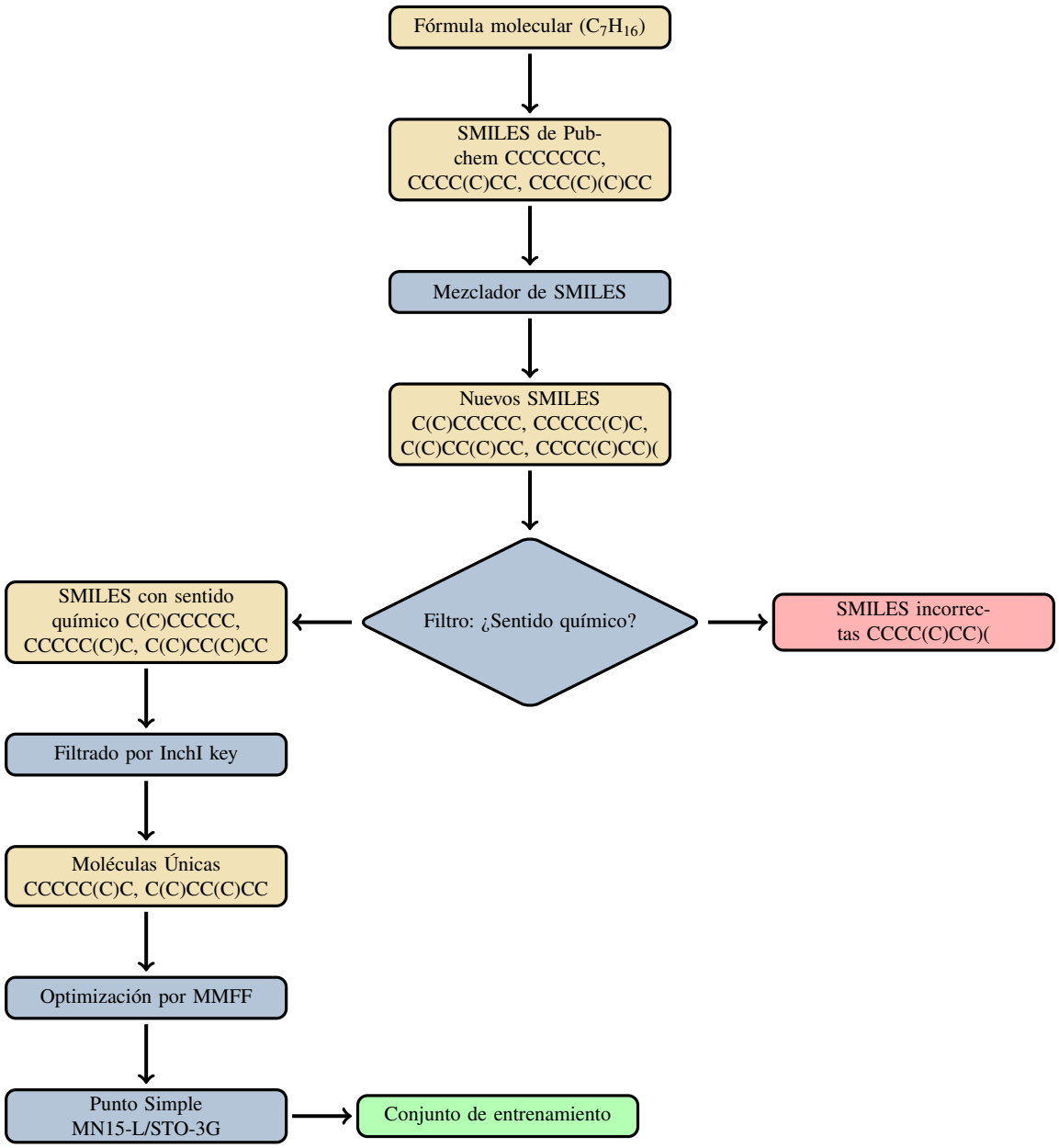


Figura 4.1: Proceso de construcción de los conjuntos de entrenamiento. La construcción de una base de datos se realiza transformando a SMILES un conjunto de fórmulas moleculares. Los caracteres de cada SMILES son permutados para obtener nuevas combinaciones que codifiquen posibles nuevas moléculas. La existencia de moléculas reales es verificada al transformar cada SMILES al identificador único InChI key y a partir de este se obtiene la geometría. Al final la geometría de cada molécula se optimiza con MMFF y se realiza un cálculo de punto simple con MN15-L/STO-3G.

Clave	Composición	Característica	Número de moléculas
C1	Isómeros de $C_{15}H_{32}$	Todas las moléculas contienen los mismos átomos y solo cambia su ubicación espacial	1500 moléculas
C2	Isómeros de los primeros 15 alcanos	Moléculas de diversos tamaños aunque con los mismos elementos	1500 moléculas
C3	Moléculas tomadas de ChEMBL	Moléculas de diversos tamaños con varios elementos y grupos funcionales	1500 moléculas

Tabla 4.1: Conjuntos de entrenamiento.

MN15-L/STO-3G. El primer conjunto de entrenamiento está integrado por los isómeros del pentadecano ($C_{15}H_{32}$), mientras que el segundo conjunto está conformado por los isómeros de los primeros quince alcanos. El tercer conjunto consiste en varias moléculas tomadas de ChEMBL [47, 48] (ChEMBL_SARS_CoV-2). Esta base de datos se eligió de forma aleatoria sin ningún propósito en particular. Para los tres conjuntos se utilizó el procedimiento descrito en la Figura 4.1. La información estructural fue codificada con los descriptores, ACSF, SOAP MBTR, y las matrices de Coulomb mediante DScribe. Hecho esto la nueva información se usó como datos de entrada (inputs), con el objetivo de determinar la representación más adecuada.

Los atributos de cada descriptor se conservaron de acuerdo a los valores preestablecidos excepto para el descriptor SOAP donde el número de funciones de base radial (n_{max}) y el máximo grado de armónicos esféricos (l_{max}) se ajustaron a 3 y 2 respectivamente para acortar el tiempo de cálculo.[29]

De cada conjunto se tomó el 70 % de las moléculas, como moléculas de entrenamiento y el 30 % restante como moléculas de prueba, en ambos casos la selección fue de manera aleatoria. Así mismo, se creó un conjunto de validación (CV) compuesto por 12 moléculas distintas, no presentes en ninguno de los tres conjuntos anteriores, las cuales se muestran en la Tabla 4.2.

Clave	Nombre	Fórmula Molecular
I	5,6-Bis(pentazol-1-yl)tetrazina	C_2N_{14}
II	5-Azido-1-(diazidometileneamino)-1H-tetrazol	C_2N_{14}
III	Bis(diazidometilen)-hidrazina	C_2N_{14}
IV	Tert-Butil 3-amino-7,8-dihidro-1,6-naftiridina-6(5H)-carboxilato	$C_{13}H_{19}N_3O_2$
V	Dodecaedrano	$C_{20}H_{20}$
VI	Ácido 2-dodecenodioico	$C_{12}H_{20}O_4$
VII	(2Z)-1,1,1,4,4,4-hexafluorobut-2-eno	$C_4H_2F_6$
VIII	Ácido Úrico	$C_5H_4N_4O_3$
IX	Geranilacetona	$C_{13}H_{22}O$
X	Arecolina	$C_8H_{13}NO_2$
XI	4-Metoxiestireno	$C_9H_{10}O$
XII	Acetato de vinilo	$C_4H_6O_2$

Tabla 4.2: Moléculas en el conjunto de validación (CV)

CAPÍTULO 5

Resultados y análisis

A continuación se muestran los resultados de entrenar el modelo de KRR con los tres conjuntos de entrenamiento, C1, C2 y C3, así como las predicciones sobre el conjunto de validación CV.

5.1. Predicción de la energía para el conjunto C1

En la Tabla 5.1 se muestran los resultados de entrenar el modelo KRR con el conjunto C1, compuesto por isómeros de alcanos con fórmula $C_{15}H_{32}$. Durante el entrenamiento el algoritmo trata de predecir los valores contenidos dentro del conjunto de prueba, la diferencia entre estos y el valor de la predicción puede tomarse como el error que comete el modelo. Al tratarse de isómeros de la misma molécula, todas las energías tiene un valor similar de -579.9444 ± 0.0087 Hartrees. Para cada métrica se realizaron 5 entrenamientos independientes, el error promedio y la desviación estándar son los valores que se muestran en la Tabla 5.1. Aunque no existe una diferencia sustancial entre los 4 descriptores, es posible notar que Coulomb es la métrica con el error promedio más grande de estos. Posiblemente porque al estar conformada la base de datos con isómeros de una sola molécula, la matriz que genera Coulomb no es capaz de captar con la misma sutileza que los demás descriptores, los detalles que diferencian a una molécula de otra, es decir la interacción entre dos átomos sólo está representada por un valor dentro de la matriz. En cambio ACSF, SOAP y MBTR ofrecen una mejor descripción del ambiente local de cada átomo al considerar más de un vecino dentro del mismo y a que muchas partes estructurales que componen cada molécula son muy parecidas entre sí. Por consiguiente la distribución que deriva de estas, concluyendo en una descripción numérica más amplia y detallada, así como facilitando al modelo la interpolación entre los datos.

Los resultados de aplicar los modelos obtenidos al conjunto de validación, se muestran en

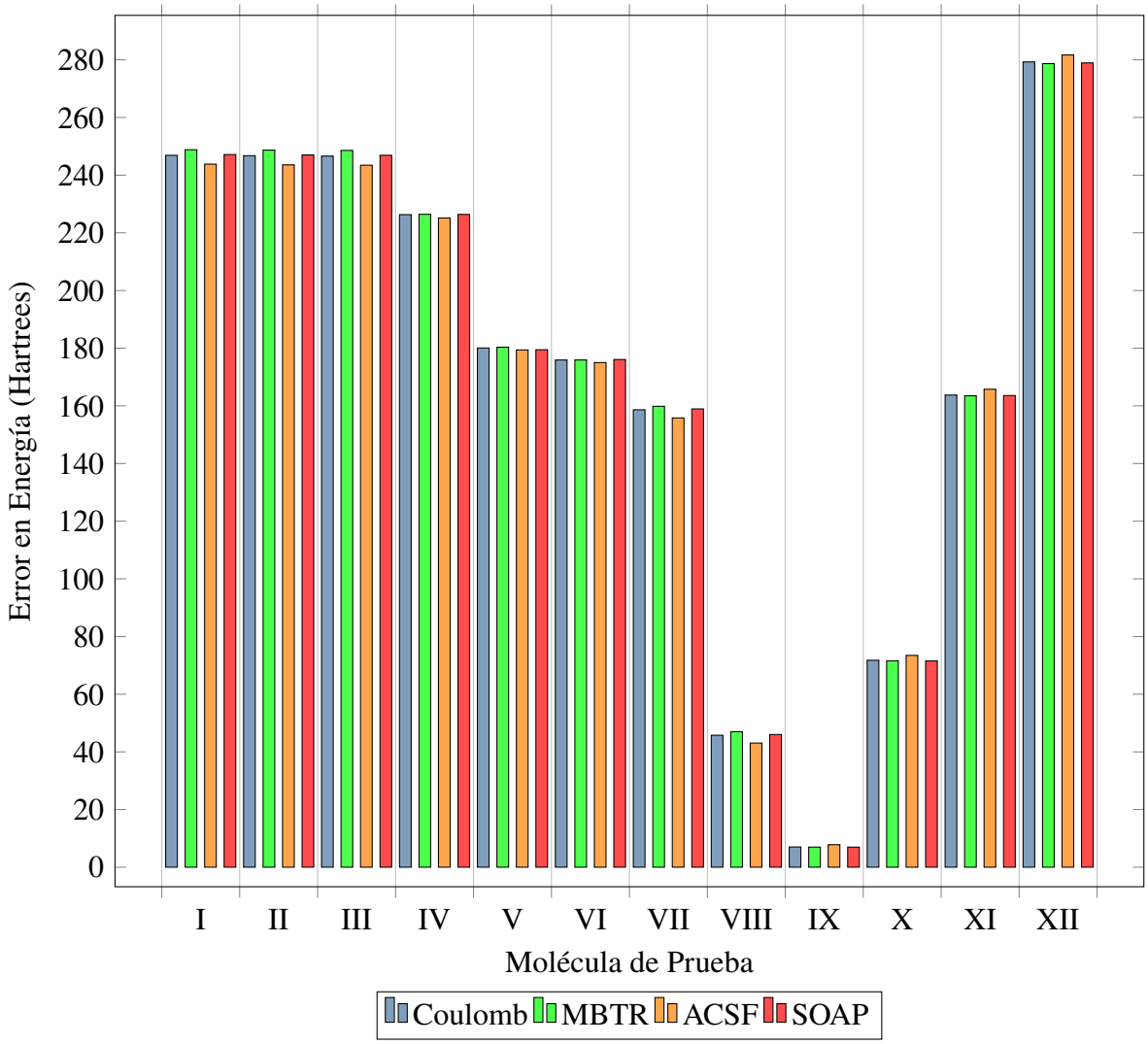
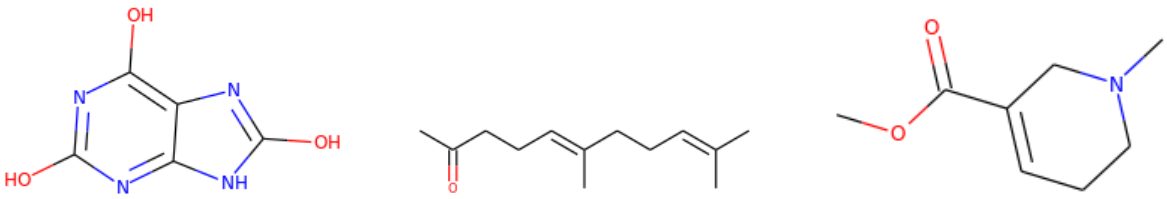


Figura 5.1: Errores de predicción del conjunto de validación CV utilizando el modelo entrenado mediante el conjunto C1. El error corresponde a la diferencia de energía entre el cálculo de MN15-L y la predicción del modelo de *machine learning*.



(a) VIII (b) IX (c) X

Figura 5.2: Moléculas del conjunto CV que presentan la menor diferencia entre el valor de energía y el predicho por la red entrenada con el conjunto C1.

Métrica	Error(Ha)
ACSF	0.001422 ± 0.000026
SOAP	0.001608 ± 0.000071
MBTR	0.001683 ± 0.000097
Coulomb	0.004270 ± 0.000063

Tabla 5.1: Error promedio que comete el modelo KRR durante el entrenamiento al evaluar distintas métricas (datos en unidades Hartree).

la Figura 5.1, donde el eje horizontal corresponde a las moléculas dentro del conjunto, y el eje vertical al error correspondiente. Para cada molécula se observan cuatro barras, cada una corresponde a usar alguna de las métricas como representación de los datos de entrada. En esta figura se muestran errores grandes en la mayoría de las moléculas, sin embargo, aquellas que corresponden a las barras VIII, IX, y X, tienen un error menor debido a la mayor similitud estructural y de energía en comparación con el resto de moléculas (Figura 5.2). Al probar el modelo frente a moléculas nuevas, se observa que sólo es capaz de identificar ambientes locales representados por vectores similares a aquellos con los que se entrenó. Como consecuencia sólo las moléculas más parecidas al pentadecano tuvieron un mejor resultado. La molécula IX que corresponde a la geranilacetona fue la que mejor predijo KRR bajo la métrica SOAP, con un error promedio de 6.980 Hartrees, posiblemente porque su cadena más grande compuesta por 11 átomos se asemeja a las cadena de los isómeros del pentadecano. Además de presentar pocos atributos nuevos en comparación con el resto de las moléculas de prueba, como son un átomo de oxígeno, dos dobles enlaces y un par de metilos, haciendo que sea relativamente más sencillo para la regresión hacer esta extrapolación.

5.2. Predicción de la energía para el conjunto C2

El conjunto C2 está compuesto por los isómeros de los alcanos lineales del metano al pentadecano. En la Tabla 5.2 se muestra el error promedio al entrenarse el modelo con el conjunto C2. Entre todas las métricas se observa que ACSF es la que presenta el menor error, siendo este de 0.1746 Hartrees, posiblemente porque la exposición del ambiente local sigue siendo el factor de mayor peso que describe el conjunto. Posteriormente se encuentran Coulomb y SOAP con valores similares, al ser los datos parecidos entre sí, Coulomb permite codificar lo suficientemente bien a pesar del tamaño de cada molécula.

De manera similar para el caso del descriptor MBTR, al ser las moléculas de entrenamiento semejantes, muchas de sus partes estructurales son similares y por lo tanto su distribución dentro la molécula también lo es, permitiendo al modelo aprender de moléculas distintas pero con atributos similares. Por otra parte, su diferencia tan marcada respecto a las demás

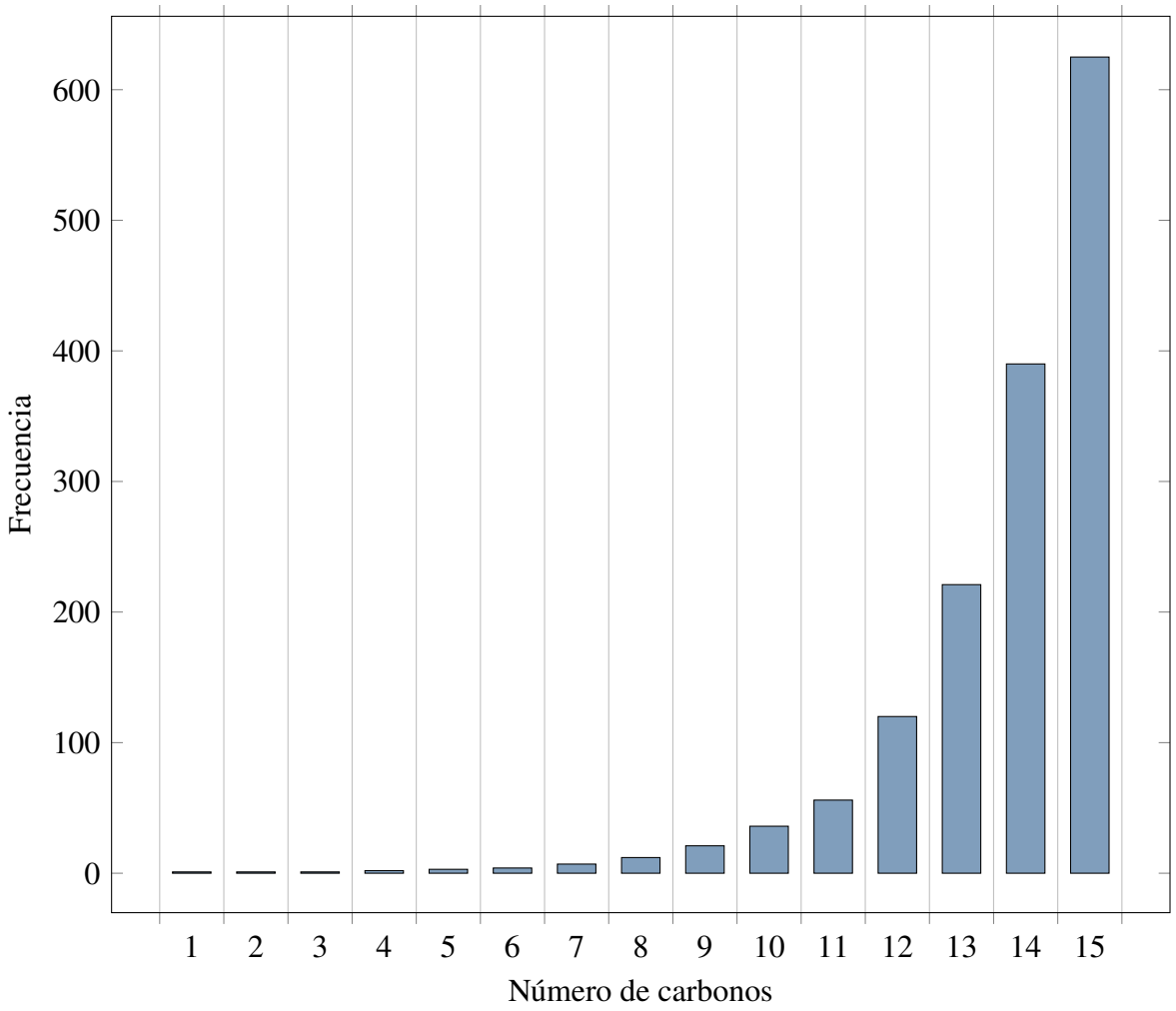


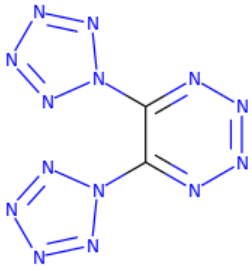
Figura 5.3: Número de moléculas por alcano dentro del conjunto C2.

métricas, se puede atribuir a la disparidad en la distribución del tipo de moléculas dentro del conjunto de entrenamiento, como se aprecia en la Figura 5.3, lo que dificulta la capacidad de KRR de distinguir entre moléculas y por tanto de hacer una buena generalización.

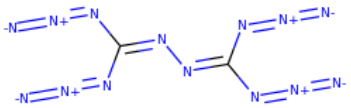
Métrica	Error (Ha)
ACSF	0.1746761 ± 0.0703653
Coulomb	0.6776926 ± 0.0424242
SOAP	0.6862915 ± 0.0698632
MBTR	1.1964202 ± 0.4904782

Tabla 5.2: Error promedio del modelo al predecir (datos en unidades Hartree).

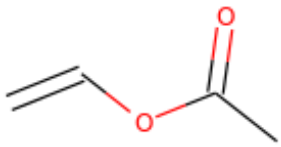
En la Figura 5.5 se muestra el error que comete la red entrenada con el conjunto C2 frente al conjunto de validación CV. La principal característica que destaca de la figura, es que la métrica de Coulomb correspondiente a las barras azules, comete errores significativamente



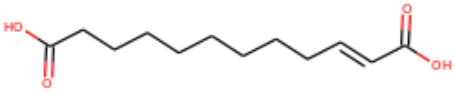
(a) I



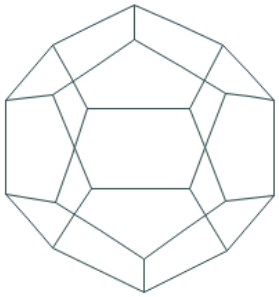
(b) III



(c) XII



(d) VI



(e) V

Figura 5.4: Moléculas del conjunto CV que presentan la menor diferencia entre el valor de energía y el predicho por la red entrenada con el conjunto C2.

menores al restos de las métricas para la mayoría de las moléculas, esto se podría justificar ya que Coulomb permite diferenciar mejor entre moléculas pues cada sistema queda codificado de forma única incluso para aquellas con atributos similares, por esta razón las moléculas que mejor predijo Coulomb fueron la I, III y VIII (Figuras 5.4a, 5.4b, y 5.2a respectivamente), pues son moléculas que destacan del resto por ser simétricas y con interacciones entre solo dos y tres tipos de átomos.

No obstante, también se pueden resaltar las moléculas que obtuvieron el menor error para el resto de las métricas. En el caso de MBTR, la molécula VI correspondiente al ácido 2-dodecenodioico (Figura 5.4d) fue la que mejor predijo la red, al ser esta un alqueno lineal de 12 carbonos, la distribución de algunos de sus motivos estructurales coinciden con los de los datos de entrenamiento y como consecuencia la predicción que hace KRR sobre esta molécula es más cercana al valor real, en comparación con las demás. Por otro lado las moléculas que mejor predijo la red bajo las métricas de SOAP y ACSF, fueron la XII, y la V (Figuras 5.4c, 5.4e), la primera al ser una molécula pequeña con pocos grupos funcionales fue más fácil de predecir en comparación con las demás moléculas. mientras que la segunda probablemente fue más sencilla de extrapolar pues es una molécula simétrica con sólo carbono e hidrógeno, los mismos elementos que componen la base de datos.

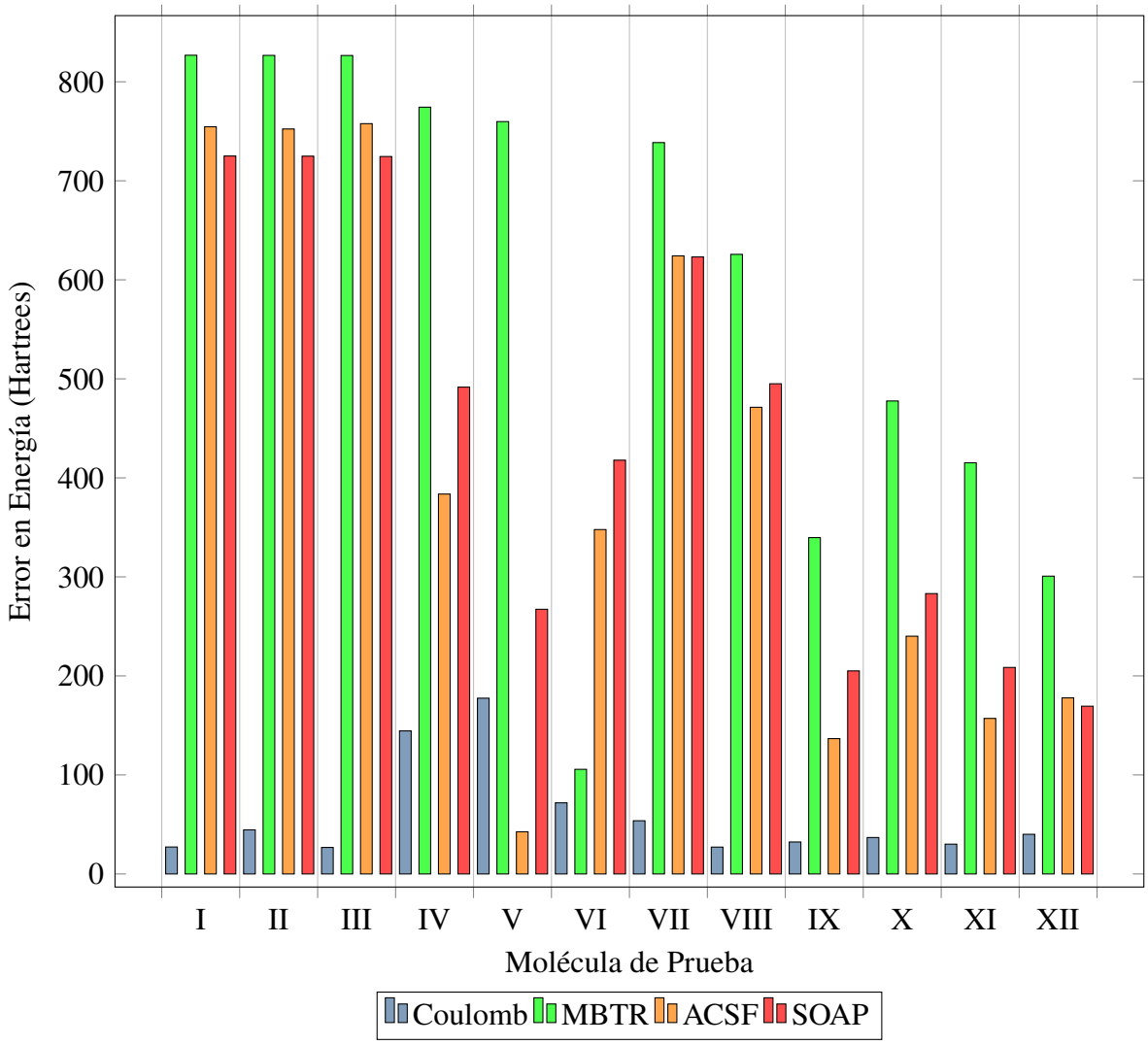
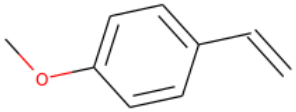


Figura 5.5: Errores de predicción del conjunto de validación CV utilizando el modelo entrenado mediante el conjunto C2. El error corresponde a la diferencia de energía entre el cálculo de MN15-L y la predicción del modelo de *machine learning*.

(a) VII



(b) XI

Figura 5.6: Moléculas del conjunto CV que presentan la menor diferencia entre el valor de energía y el predicho por la red entrenada con el conjunto C3.

5.3. Predicción con moléculas variadas del conjunto C3

El conjunto C3 es una base de datos más robusta pues está compuesta por moléculas con distinto número de átomos y grupos funcionales, lo que implica una mayor diferencia en los ambientes locales de cada molécula y una mayor variabilidad en la descripción que ofrecen las métricas, factores que dificultan al modelo aprender cambios locales y probablemente los principales motivos por los que ASCF y SOAP fallan. De forma semejante, la composición y el arreglo espacial de los átomos de cada molécula al ser tan distinto genera que la distribución de cada parte de la estructura sea radicalmente diferente respecto a las demás y por lo tanto también la respuesta que ofrece la métrica MBTR. Al ser los inputs tan distintos entre sí, al algoritmo le cuesta más trabajo aprender y generalizar. Coulomb es el descriptor que mejor representa la energía de un conjunto de moléculas tan diversas, pues cargas nucleares y posiciones atómicas pueden ser directamente relacionadas con la energía de un sistema cualquier sistema.

Métrica	Error (Ha)
Coulomb	0.654754 ± 0.110973
ACSF	13.627188 ± 1.011053
SOAP	14.146045 ± 0.353411
MBTR	31.208932 ± 1.343187

Tabla 5.3: Error promedio del modelo al predecir (datos en unidades Hartree).

En la Figura 5.7 se muestran las predicciones que realiza KRR sobre las moléculas de validación al entrenarse con el conjunto C3. De forma general, se puede observar que la representación de toda la molécula es el factor más importante y no el ambiente local de cada

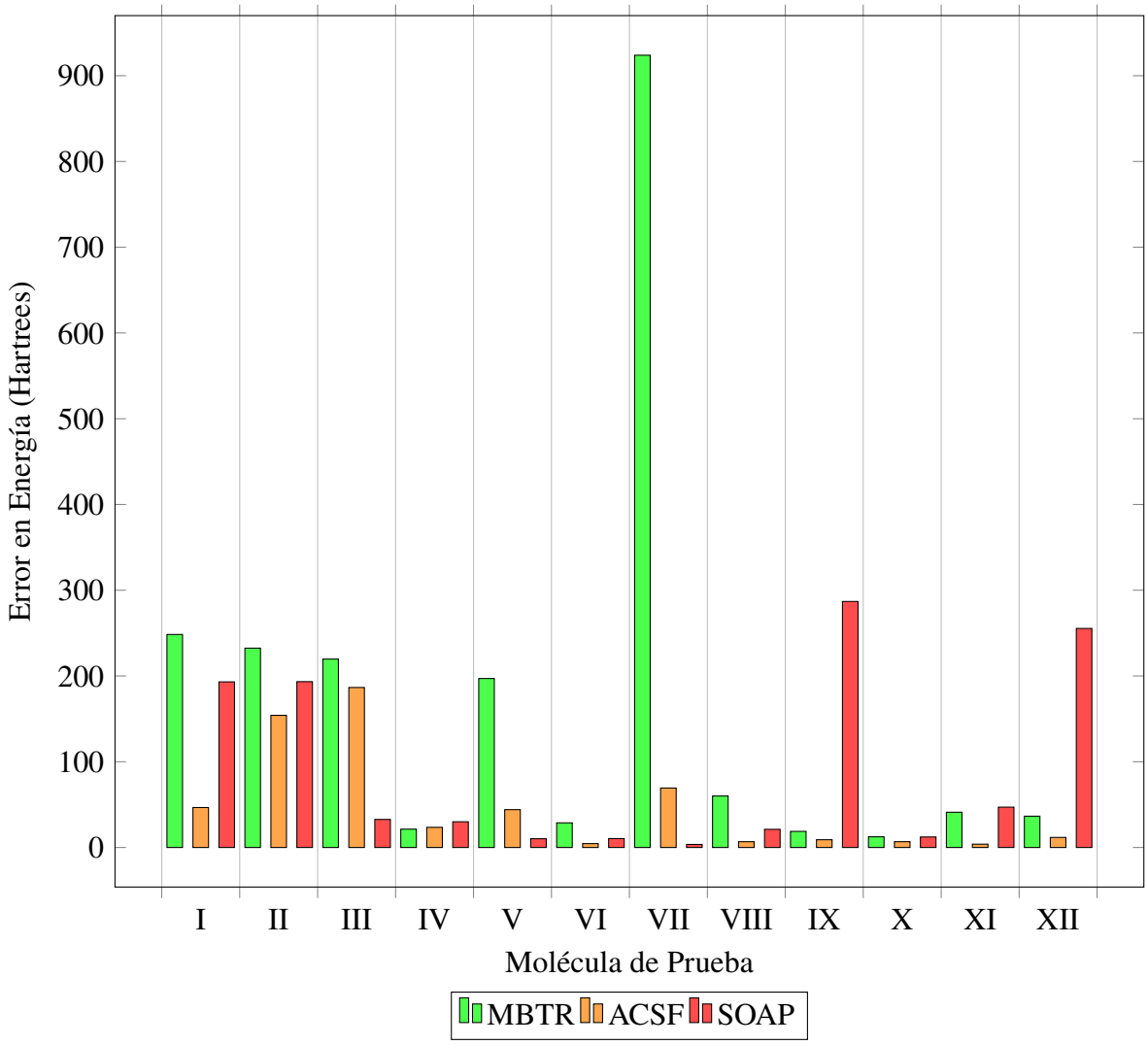


Figura 5.7: Errores de predicción del conjunto de validación CV utilizando el modelo entrenado mediante el conjunto C3. El error corresponde a la diferencia de energía entre el cálculo de MN15-L y la predicción del modelo de *machine learning*.

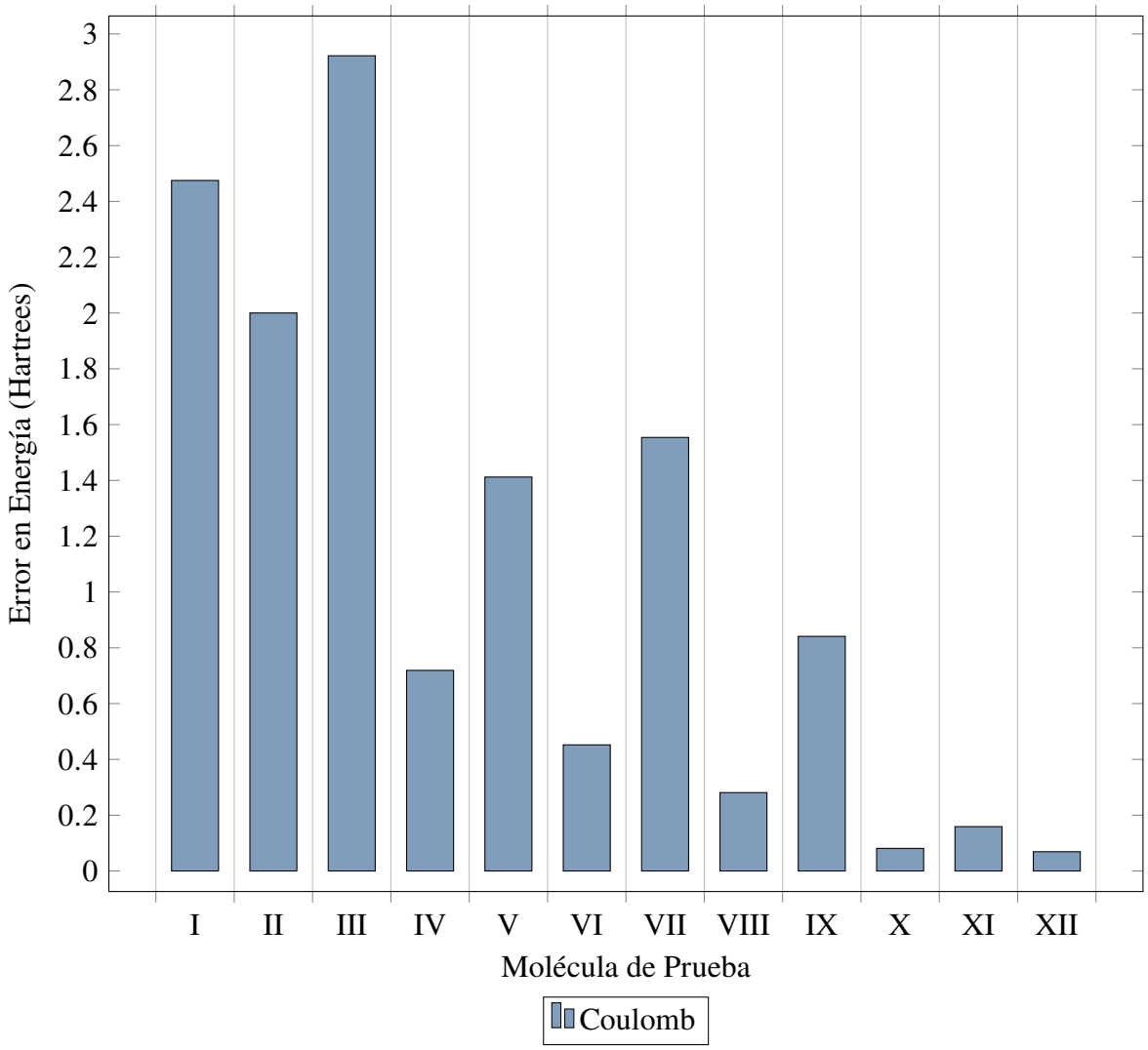


Figura 5.8: Error de predicción del conjunto de validación CV asociado a la métrica de Coulomb, utilizando el modelo entrenado con el conjunto C3.

átomo para que aprenda el modelo, pues la métrica de Coulomb es la que mejores predicciones aporta, en especial sobre la molécula XII (Figura5.4c), por ser esta la molécula más pequeña dentro del conjunto de prueba y con pocos grupos funcionales. No obstante se puede destacar que la molécula X (Figura 5.2c), con un error de 12.459 Hartrees, fue la que mejor describió MBTR posiblemente por que los grupos funcionales con los que cuenta, son muy comunes dentro de la base de datos, y por lo tanto la distribución de estos motivos es mayor que otros. La molécula VII (Figura 5.6a) con un error de 3.572 Hartrees, fue la mejor predicha por el algoritmo al entrenarse con los datos representados por SOAP, posiblemente porque al ser una molécula muy simétrica muchas de las vecindades de cada átomo son muy parecidas entre sí. Por otro lado la molécula XI (Figura 5.6b), fue la molécula mejor descrita por ACSF con un error de 3.867 Hartrees. El valor de la predicción se puede deber a que el 4-metoxiestireno que es la molécula correspondiente, es una molécula pequeña y sencilla con grupos funcionales comunes dentro de la base de datos por lo que la regresión puede generalizar más fácilmente.

CAPÍTULO 6

Conclusiones y perspectivas

Predecir la energía molecular a través de cálculos típicos de mecánica cuántica, en la mayoría de los casos suele ser tardado y no escalable a sistemas más complejos, por lo que encontrar una alternativa que brinde resultados rápidos y precisos es de suma importancia. Los algoritmos de ML son una herramienta que muchas veces es perfilada como una alternativa viable. Sin embargo, para alcanzar cada vez mejores resultados no sólo basta con el uso y desarrollo de algoritmos cada vez más complejos, sino entender cómo se relacionan los datos con el problema. En el caso particular de la energía molecular, cómo elegir la descripción que mejor relacione la estructura química con la energía que a su vez sea capaz de compensar el poder de generalización del algoritmo con la variabilidad de los datos. En este trabajo se estudió el desempeño de dos tipos de descriptores globales, las matrices de Coulomb y las matrices MBTR, junto con dos tipos de descriptores locales, las matrices ACSF y las matrices SOAP frente a tres bases de datos de moléculas orgánicas con características diferentes.

Observando los resultados obtenidos es posible notar que las métricas enfocadas en representar el ambiente local de cada átomo tienen la ventaja de brindar una descripción más detallada y resultados más exactos si lo que se busca es que la red aprenda a distinguir entre un grupo de isómeros o de moléculas con atributos similares. Sin embargo, si el conjunto de moléculas de las que se busca aprender tiene características muy diversas, es probable que la descripción que genere sobre una molécula varíe considerablemente respecto a la descripción de otra , dificultando que el algoritmo establezca una función que las relacione. De la misma forma, si el vector que se genera es demasiado grande, este puede contener elementos redundantes que ocasionen problemas de sobre ajuste en el algoritmo y por consiguiente de generalización.

Por otro, lado los descriptores globales y en particular las matrices de Coulomb brindaron los mejores resultados frente a datos con características diversas. La codificación de cada estructura con este descriptor cuenta con menos elementos en comparación con las demás

métricas, lo que permite reducir la dimensionalidad del problema. De igual forma, al derivarse cada elemento de la matriz, de las cargas y las posiciones atómicas, este puede ser más fácilmente asociado con la energía. Muchas veces las matrices de Coulomb suelen descartarse como opción frente a descriptores más complejos y nuevos, principalmente porque muchas matrices pueden ser asociadas con la misma molécula tan solo permutando el orden de los renglones y las columnas. [49]. Sin embargo como lo demuestra este trabajo, siguen siendo una opción que permite codificar y distinguir con buena precisión moléculas diversas si el objetivo es encontrar una relación entre la estructura y la energía.

Las redes neuronales de grafos son una clase de algoritmos que permite trabajar con *inputs* de dimensiones no fijas que han demostrado una gran habilidad para representar y predecir estructuras y propiedades químicas. Una propuesta interesante que ya ha empezado a ser estudiada en el trabajo de Yang y sus colaboradores, [12] es la creación de modelos híbridos basados en redes neuronales de grafos y características atómicas.

Continuando por esta linea de investigación, es posible que si se amplia la información de cada átomo con la que ofrecen descriptores locales, como los presentados en este trabajo, la descripción de cada molécula sea mas detallada y tanto el aprendizaje, como la generalización del algoritmo crezcan.

CAPÍTULO 7

Anexos

7.1. Predicción de la energía usando los datos del conjunto C1 como entrenamiento

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-577.970	248.821
II	-826.645	-577.959	248.687
III	-826.533	-577.956	248.577
IV	-806.235	-579.805	226.429
V	-759.991	-579.649	180.342
VI	-755.83	-579.925	175.905
VII	-738.583	-578.733	159.850
VIII	-625.707	-578.702	47.005
IX	-572.922	-579.902	6.980
X	-508.201	-579.785	71.584
XI	-416.233	-579.718	163.485
XII	-300.718	-579.402	278.684

Tabla 7.1: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor MBTR como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-579.640	247.151
II	-826.645	-579.637	247.008
III	-826.533	-579.637	246.896
IV	-806.235	-579.854	226.381
V	-759.991	-580.535	179.456
VI	-755.83	-579.769	176.061
VII	-738.583	-579.649	158.935
VIII	-625.707	-579.681	46.026
IX	-572.922	-579.886	6.964
X	-508.201	-579.740	71.539
XI	-416.233	-579.801	163.568
XII	-300.718	-579.640	278.922

Tabla 7.2: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor SOAP como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-582.971	243.82
II	-826.645	-583.075	243.571
III	-826.533	-583.088	243.446
IV	-806.235	-581.081	225.154
V	-759.991	-580.603	179.388
VI	-755.83	-580.818	175.012
VII	-738.583	-582.770	155.813
VIII	-625.707	-582.654	43.053
IX	-572.922	-580.730	7.808
X	-508.201	-581.649	73.448
XI	-416.233	-581.986	165.753
XII	-300.718	-582.401	281.683

Tabla 7.3: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor ACSF como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-579.904	246.886
II	-826.645	-579.895	246.75
III	-826.533	-579.912	246.621
IV	-806.235	-579.926	226.309
V	-759.991	-579.925	180.066
VI	-755.83	-579.935	175.895
VII	-738.583	-579.949	158.634
VIII	-625.707	-579.943	45.765
IX	-572.922	-579.945	7.023
X	-508.201	-579.966	71.765
XI	-416.233	-579.976	163.743
XII	-300.718	-579.994	279.276

Tabla 7.4: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando las matrices de Coulomb como métrica (datos en unidades Hartree).

7.2. Predicción de la energía usando los datos del conjunto C2 como entrenamiento

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-3.420E-28	826.791
II	-826.645	-3.329E-28	826.645
III	-826.533	-3.298E-28	826.533
IV	-806.235	-31.907	774.327
V	-759.991	-0.214	759.777
VI	-755.83	-689.757	105.647
VII	-738.583	-2.706E-15	738.583
VIII	-625.707	-5.498E-16	625.707
IX	-572.922	-912.570	339.648
X	-508.201	-30.568	477.633
XI	-416.233	-0.949	415.284
XII	-300.718	-0.002	300.716

Tabla 7.5: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor MBTR como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-101.752	725.039
II	-826.645	-101.757	724.888
III	-826.533	-102.014	724.519
IV	-806.235	-314.610	491.625
V	-759.991	-492.715	267.275
VI	-755.83	-337.978	417.852
VII	-738.583	-115.438	623.145
VIII	-625.707	-130.682	495.025
IX	-572.922	-367.879	205.043
X	-508.201	-225.081	283.12
XI	-416.233	-207.701	208.532
XII	-300.718	-131.323	169.394

Tabla 7.6: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor SOAP como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-72.234	754.557
II	-826.645	-74.309	752.337
III	-826.533	-68.844	757.690
IV	-806.235	-422.604	383.631
V	-759.991	-717.417	42.574
VI	-755.83	-408.042	347.788
VII	-738.583	-114.373	624.210
VIII	-625.707	-154.426	471.281
IX	-572.922	-436.255	136.667
X	-508.201	-268.117	240.084
XI	-416.233	-259.172	157.060
XII	-300.718	-122.922	177.796

Tabla 7.7: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor ACSF como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferecia (Abs.)
I	-826.791	-799.589	27.202
II	-826.645	-782.120	44.5252
III	-826.533	-799.753	26.7805
IV	-806.235	-661.761	144.473
V	-759.991	-582.496	177.495
VI	-755.83	-684.007	71.8226
VII	-738.583	-684.867	53.717
VIII	-625.707	-598.608	27.100
IX	-572.922	-540.655	32.2674
X	-508.201	-471.420	36.7808
XI	-416.233	-386.153	30.0795
XII	-300.718	-260.692	40.0262

Tabla 7.8: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando las matrices de Coulomb como métrica (datos en unidades Hartree).

7.3. Predicción de la energía usando los datos del conjunto C3 como entrenamiento

Molécula	Energía	Predicción	Diferencia (Abs.)
I	-826.791	-607.407	248.389
II	-826.645	-667.026	232.471
III	-826.533	-693.439	219.692
IV	-806.235	-827.590	21.355
V	-759.991	-562.945	197.046
VI	-755.83	-784.575	28.745
VII	-738.583	-1662.457	923.874
VIII	-625.707	-685.749	60.042
IX	-572.922	-591.698	18.776
X	-508.201	-520.659	12.459
XI	-416.233	-375.120	41.113
XII	-300.718	-264.259	36.495

Tabla 7.9: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor MBTR como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferencia (Abs.)
I	-826.791	-633.664	193.127
II	-826.645	-545.212	193.371
III	-826.533	-723.123	32.707
IV	-806.235	-602.917	29.995
V	-759.991	-626.348	10.208
VI	-755.830	-501.102	10.421
VII	-738.583	-419.273	3.572
VIII	-625.707	-321.875	21.158
IX	-572.922	-539.638	286.895
X	-508.201	-754.135	12.251
XI	-416.233	-853.291	47.056
XII	-300.718	-571.229	255.417

Tabla 7.10: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor SOAP como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferencia (Abs.)
I	-826.791	-780.205	46.586
II	-826.645	-672.527	154.118
III	-826.533	-639.973	186.560
IV	-806.235	-829.731	23.496
V	-759.991	-804.098	44.107
VI	-755.83	-758.848	4.536
VII	-738.583	-679.737	69.350
VIII	-625.707	-627.942	6.750
IX	-572.922	-565.723	9.051
X	-508.201	-508.271	6.760
XI	-416.233	-412.365	3.867
XII	-300.718	-312.458	11.740

Tabla 7.11: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando el descriptor ACSF como métrica (datos en unidades Hartree).

Molécula	Energía	Predicción	Diferencia (Abs.)
I	-826.791	-829.266	2.475
II	-826.645	-828.645	2.000
III	-826.533	-829.455	2.922
IV	-806.235	-805.516	0.719
V	-759.991	-761.402	1.412
VI	-755.83	-755.643	0.452
VII	-738.583	-737.029	1.554
VIII	-625.707	-625.988	0.281
IX	-572.922	-572.081	0.841
X	-508.201	-508.281	0.081
XI	-416.233	-416.074	0.159
XII	-300.718	-300.787	0.069

Tabla 7.12: Predicción de la energía que realiza el modelo sobre las moléculas del conjunto CP usando las matrices de Coulomb como métrica (datos en unidades Hartree).

Bibliografía

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[4] G. Landrum, "Rdkit." `https://www.rdkit.org`, 2010.

[5] L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, "DScribe: Library of descriptors for machine learning in materials science," *Computer Physics Communications*, vol. 247, p. 106949, 2020.

[6] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing, and Z. Wu, *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. `https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837`.

[7] G. Graziano, "Deep learning chemistry ab initio," nov 2020.

[8] J. Hermann, Z. Schätzle, and F. Noé, "Deep-neural-network solution of the electronic Schrödinger equation," *Nature Chemistry*, vol. 12, pp. 891–897, oct 2020.

[9] K. T. Schütt, M. Gastegger, A. Tkatchenko, K. R. Müller, and R. J. Maurer, "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," *Nature Communications*, vol. 10, pp. 1–10, dec 2019.

[10] Z. Wang, S. Ye, H. Wang, J. He, Q. Huang, and S. Chang, "Machine learning method for tight-binding Hamiltonian parameterization from ab-initio band structure," *npj Computational Materials*, vol. 7, pp. 1–10, dec 2021.

[11] J. T. Margraf and K. Reuter, "Pure non-local machine-learned density functional theory for electron correlation," *Nature Communications*, vol. 12, pp. 1–7, dec 2021.

[12] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay, "Analyzing Learned Molecular Representations for Property Prediction," *Journal of Chemical Information and Modeling*, vol. 59, no. 8, pp. 3370–3388, 2019.

[13] L. A. Miccio and G. A. Schwartz, "From chemical structure to quantitative polymer properties prediction through convolutional neural networks," *Polymer*, vol. 193, no. March, p. 122341, 2020.

[14] S. Boobier, D. R. Hose, A. J. Blacker, and B. N. Nguyen, "Machine learning with physicochemical relationships: solubility prediction in organic solvents and water," *Nature Communications*, vol. 11, pp. 1–10, dec 2020.

[15] M. Tsubaki and T. Mizoguchi, "Fast and Accurate Molecular Property Prediction: Learning Atomic Interactions and Potentials with Neural Networks," *Journal of Physical Chemistry Letters*, vol. 9, no. 19, pp. 5733–5741, 2018.

[16] S. Kiyohara, H. Oda, T. Miyata, and T. Mizoguchi, "Prediction of interface structures and energies via virtual screening," *Science Advances*, vol. 2, no. 11, pp. 1–8, 2016.

[17] S. Stocker, G. Csányi, K. Reuter, and J. T. Margraf, "Machine learning in chemical reaction space," *Nature Communications*, vol. 11, pp. 1–11, dec 2020.

[18] J. E. Hein, "Machine learning made easy for optimizing chemical reactions," *Nature*, vol. 590, pp. 40–41, feb 2021.

[19] K. T. Schütt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko, and K. R. Müller, "SchNet - A deep learning architecture for molecules and materials," *Journal of Chemical Physics*, vol. 148, p. 241722, jun 2018.

[20] D. A. O'Sullivan and W. Lepkowski, "Chemical science," *Chemical and Engineering News*, vol. 68, no. 20, pp. 42–61, 1990.

[21] J. W. Barnett, C. R. Bilchak, Y. Wang, B. C. Benicewicz, L. A. Murdock, T. Bereau, and S. K. Kumar, "Designing exceptional gas-separation polymer membranes using machine learning," *Science Advances*, vol. 6, p. eaaz4301, may 2020.

[22] Y. C. Lo, S. E. Rensi, W. Torng, and R. B. Altman, "Machine learning in chemoinformatics and drug discovery," aug 2018.

[23] O. T. Unke and M. Meuwly, "PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges," *Journal of Chemical Theory and Computation*, vol. 15, no. 6, pp. 3678–3693, 2019.

[24] M. Rupp, A. Tkatchenko, K. R. Müller, and O. A. Von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical Review Letters*, vol. 108, no. 5, pp. 1–5, 2012.

[25] A. J. Lockett, "No free lunch theorems," *Natural Computing Series*, vol. 1, no. 1, pp. 287–322, 2020.

[26] H. Huo and M. Rupp, "Unified Representation of Molecules and Crystals for Machine Learning," no. i, 2017.

[27] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Physical Review B - Condensed Matter and Materials Physics*, vol. 87, p. 184115, may 2013.

[28] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," *The Journal of Chemical Physics*, vol. 134, p. 074106, feb 2011.

[29] L. Himanen, M. O. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, "DScribe: Library of descriptors for machine learning in materials science," *Computer Physics Communications*, vol. 247, p. 106949, 2020.

[30] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, pp. 742–754, may 2010.

[31] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open Babel," *Journal of Cheminformatics*, vol. 3, no. 33, pp. 1–14, 2011.

[32] R. C. Glen, A. Bender, C. H. Arnby, L. Carlsson, S. Boyer, and J. Smith, "Circular fingerprints: Flexible molecular descriptors with applications from physical chemistry to ADME," *IDrugs*, vol. 9, pp. 199–204, mar 2006.

[33] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of MDL keys for use in drug discovery," *Journal of Chemical Information and Computer Sciences*, vol. 42, pp. 1273–1280, nov 2002.

[34] N. M. O'Boyle and R. A. Sayle, "Comparing structural fingerprints using a literature-based similarity benchmark," *Journal of Cheminformatics*, vol. 8, p. 36, jul 2016.

[35] K. P. Murphy, *Machine Learning, A Probabilistic Perspective*. Cambridge,, Massachusetts: MIT Press, 2012.

[36] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*. Cambridge, United Kingdom: Cambridge University Press, 2000.

[37] M. A. A.K. Md. Ehsanes Saleh and B. G. Kibria, *Theory of Ridge Regression Estimation with Applications*. United States of America: John Wiley & Sons, 2019.

[38] T. H. Gareth James, Daniela Witten and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2013.

[39] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, "Comparing molecules and solids across structural and alchemical space," *Physical Chemistry Chemical Physics*, vol. 18, no. 20, pp. 13754–13769, 2016.

[40] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, and E. E. Bolton, "PubChem in 2021: New data content and improved web interfaces," *Nucleic Acids Research*, vol. 49, no. D1, pp. D1388–D1395, 2021.

[41] D. Weininger, "SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[42] E. J. Bjerrum, "Smiles enumeration as data augmentation for neural network modeling of molecules," *arXiv*, no. Figure 1, 2017.

[43] S. Heller, "InChI – the worldwide chemical structure standard," *Journal of Cheminformatics*, vol. 6, no. S1, pp. 1–9, 2014.

[44] P. Tosco, N. Stiefl, and G. Landrum, "Bringing the MMFF force field to the RDKit: Implementation and validation," *Journal of Cheminformatics*, vol. 6, no. 1, pp. 4–7, 2014.

[45] H. S. Yu, X. He, and D. G. Truhlar, "MN15-L: A New Local Exchange-Correlation Functional for Kohn-Sham Density Functional Theory with Broad Accuracy for Atoms, Molecules, and Solids," *Journal of Chemical Theory and Computation*, vol. 12, no. 3, pp. 1280–1293, 2016.

[46] K. D. Dobbs and W. J. Hehre, "Molecular orbital theory of the properties of inorganic and organometallic compounds 5. Extended basis sets for first-row transition metals," *Journal of Computational Chemistry*, vol. 8, no. 6, pp. 861–879, 1987.

[47] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, M. Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. J. Radoux, A. Segura-Cabrera, A. Hersey, and A. R. Leach, "ChEMBL: Towards direct deposition of bioassay data," *Nucleic Acids Research*, vol. 47, no. D1, pp. D930–D940, 2019.

[48] M. Davies, M. Nowotka, G. Papadatos, N. Dedman, A. Gaulton, F. Atkinson, L. Bellis, and J. P. Overington, "ChEMBL web services: Streamlining access to drug discovery data and utilities," *Nucleic Acids Research*, vol. 43, no. W1, pp. W612–W620, 2015.

[49] G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, A. Lilienfeld, and K.-R. Müller, "Learning Invariant Representations of Molecules for Atomization Energy Prediction," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.