

# PROYECTO FINAL

## PREDICCIÓN DE LA CONSTANTE DE ACOPLAMIENTO ESCALAR

Abtration



NÉSTOR BATISTA DÍAZ

# ÍNDICE

<b>1. Introducción</b>	<b>3</b>
<b>2. Breve sumario de productos o resultados obtenidos.</b>	<b>4</b>
<b>3. Contexto y justificación del Trabajo</b>	<b>5</b>
3.1. Objetivos del Trabajo.	6
3.2. Enfoque y metodología.	7
3.3. Planificación del Trabajo.	7
3.4. Recursos de sistemas utilizados.	8
3.4.1. Recursos de hardware:	8
3.4.2. Recursos de software:	10
3.4.3. Control de versiones y despliegue:	10
<b>4. Creación del set de datos</b>	<b>11</b>
4.1. Estudio de datos necesarios.	11
4.2. Fuente de datos.	12
4.3. Recolección de los datos.	12
4.4. Visualización de datos.	12
4.4.1. Distribución espacial de los átomos	13
4.4.2. Densidades de las características comparada con el target	14
4.4.2.1. Densidad del Target	14
4.4.2.2. Densidades de las características más importantes	14
4.4.3. Matriz de correlación	15
<b>5. Optimización, normalización y calidad del set de datos.</b>	<b>16</b>
5.1. Técnicas de optimización y normalización	16
5.2. Aseguramiento de la calidad e integridad de los datos	17
5.3. Selección de características	18
5.4. Desafíos relacionados con la calidad de los datos	19
<b>6. Desarrollo del modelo predictivo</b>	<b>19</b>
6.1. Justificación del modelo seleccionado.	19
6.2. Descripción del modelo.	22
<b>7. Entrenamiento y evaluación del modelo predictivo.</b>	<b>23</b>
7.1. Configuración del proceso de entrenamiento	23
7.2. Métricas de evaluación	24
7.3. Resultados obtenidos	24
<b>8. Informe de rendimiento y métricas en aula virtualizada y otros entornos.</b>	<b>26</b>
8.1. Evaluación del rendimiento en diferentes entornos	26
8.1.1. Entornos evaluados	26
8.2. Métricas utilizadas	28
8.3. Resultados y hallazgos clave	29
8.4. Conclusiones	29
<b>9. Puesta en producción (API y aplicación cliente).</b>	<b>30</b>
9.1. Implementación de la API y la aplicación cliente	30
9.2. Tecnologías utilizadas para el desarrollo y despliegue	30
9.3. Flujo de trabajo	31
9.4. Desafíos en el despliegue	31
<b>10. Conclusiones y mejoras a realizar, modelos alternativos.</b>	<b>33</b>
10.1. Conclusiones	33
10.2. Mejoras a realizar	33
10.3. Modelos alternativos	34
<b>11. Bibliografía.</b>	<b>35</b>

# 1. Introducción

El avance en la comprensión de las interacciones moleculares es un desafío crucial en diversas áreas de la ciencia, como la química, la biología y la farmacología. En este proyecto final de curso, nos hemos embarcado en una competencia de Kaggle que aborda este desafío a nivel atómico, buscando predecir interacciones magnéticas entre átomos dentro de una molécula.

Las tecnologías de imagen como la Resonancia Magnética (MRI) y la Resonancia Magnética Nuclear (NMR) nos permiten visualizar y comprender la composición molecular de los tejidos. La NMR, en particular, utiliza principios físicos para desentrañar la estructura y dinámica de proteínas y otras moléculas complejas. En laboratorios de todo el mundo, los experimentos de NMR son fundamentales para avanzar en el conocimiento de la estructura molecular en áreas como la ciencia ambiental, la farmacología y la ciencia de materiales.

La competencia, organizada por el programa CChemistry and Mathematics in Phase Space (CHAMPS) en colaboración con la Universidad de Bristol, la Universidad de Cardiff, el Imperial College y la Universidad de Leeds, desafía a los participantes a desarrollar un algoritmo capaz de predecir la interacción magnética entre dos átomos de una molécula, también conocida como la constante de acoplamiento escalar.

El objetivo es diseñar un método rápido y fiable que permita a los químicos medicinales obtener información estructural de manera más eficiente y económica, facilitando la comprensión de cómo la estructura tridimensional de una molécula afecta sus propiedades y comportamiento. Las soluciones propuestas tendrán el potencial de acelerar el diseño de nuevas moléculas con aplicaciones específicas, como el desarrollo de mejores medicamentos para combatir enfermedades.

Nuestro proyecto se centra en desarrollar y perfeccionar un modelo predictivo que contribuya a este esfuerzo científico global, con la esperanza de colaborar con instituciones

de renombre y avanzar en la investigación académica a través de publicaciones conjuntas. Este desafío no solo representa una oportunidad para aplicar nuestros conocimientos de ciencia de datos y química computacional, sino también para impactar positivamente en el avance de la ciencia molecular.

## 2. Breve sumario de productos o resultados obtenidos.

A lo largo del desarrollo de este trabajo, se han logrado varios productos y resultados significativos. En primer lugar, se obtuvo un modelo de predicción con un rendimiento notable, alcanzando un error absoluto medio (MAE) de 1,59. Este modelo fue diseñado para predecir la constante de acoplamiento escalar a partir de los datos proporcionados en un formulario.

El set de datos utilizado para entrenar y evaluar el modelo proviene de una competición en Kaggle, especializada en este tipo de problemas. Este conjunto de datos es extenso, con más de 4,5 millones de filas, lo que justifica la necesidad de un enfoque robusto y eficiente para el modelado predictivo.

Para el desarrollo del modelo, se utilizaron los algoritmos de H2O, ya que esta plataforma es conocida por recomendar modelos óptimos para datasets de gran tamaño. La elección de H2O se debió a su capacidad para manejar y procesar grandes volúmenes de datos eficientemente, lo cual era crucial para este proyecto.

Los resultados obtenidos con el modelo fueron fructíferos, destacando un MAE bastante bajo, lo que indica una alta precisión en las predicciones. Además, se identificó un amplio margen de mejora para futuras iteraciones del modelo.

Adicionalmente, se implementó una API utilizando Flask, que permitió la creación de una aplicación web sencilla. Esta aplicación incluye un formulario a través del cual los usuarios

pueden ingresar datos y obtener predicciones sobre la constante de acoplamiento escalar, facilitando así el acceso y la usabilidad del modelo predictivo en aplicaciones prácticas.

### 3. Contexto y justificación del Trabajo

El presente trabajo se enmarca en el ámbito de la química, con el objetivo de desarrollar un método rápido y fiable para predecir las interacciones moleculares a través de la constante de acoplamiento escalar. La importancia de este proyecto radica en su potencial para proporcionar a los químicos herramientas más eficientes y económicas para adquirir conocimientos estructurales sobre las moléculas. Esto permitirá a los científicos comprender mejor cómo la estructura tridimensional de una molécula afecta a sus propiedades y comportamiento.

Actualmente, los químicos utilizan máquinas de Resonancia Magnética Nuclear (RMN) para calcular las constantes de acoplamiento escalar. Estas máquinas son extremadamente costosas, con precios que oscilan entre los 500.000 y 1 millón de euros, además de incurrir en costos adicionales por uso que van de 100 a más de 200 euros por sesión. Además, aunque los métodos avanzados de mecánica cuántica pueden calcular estas constantes con precisión, estos procesos son sumamente costosos y lentos, requiriendo días o incluso semanas por cada molécula, lo cual limita su aplicabilidad en los flujos de trabajo cotidianos.

La necesidad de este trabajo se justifica por la posibilidad de reducir significativamente los costos y el tiempo requeridos para obtener estas medidas. Al desarrollar un modelo predictivo eficaz, se podrán realizar estos cálculos de manera más accesible y rápida, democratizando el acceso a estos conocimientos y potenciando la investigación y desarrollo en la química.

Este trabajo no busca resolver un problema específico más allá de la optimización de costos y tiempo en la obtención de constantes de acoplamiento escalar, ofreciendo una alternativa viable a los métodos tradicionales y costosos actualmente en uso.

### 3.1. Objetivos del Trabajo.

El objetivo principal de este trabajo es desarrollar un modelo predictivo rápido y fiable para calcular la constante de acoplamiento escalar a partir de datos moleculares, proporcionando una alternativa más económica y eficiente a los métodos tradicionales utilizados en la química. Este modelo permitirá a los químicos obtener conocimientos estructurales de manera más accesible, reduciendo tanto los costos como el tiempo necesario para estos cálculos.

Objetivos específicos del trabajo incluyen:

- **Desarrollo del modelo predictivo:** Crear un modelo que pueda manejar grandes volúmenes de datos, específicamente un dataset con más de 4,5 millones de filas.
- **Creación del set de datos:** Recolectar y procesar datos relevantes de una competición en Kaggle para asegurar que el modelo esté basado en información robusta y representativa.
- **Evaluación y optimización del modelo:** Medir el rendimiento del modelo utilizando métricas como el error absoluto medio (MAE) y optimizarlo para obtener los mejores resultados posibles.
- **Implementación práctica:** Desarrollar una API utilizando Flask y una aplicación web con un formulario que permita a los usuarios ingresar datos y obtener predicciones sobre la constante de acoplamiento escalar.
- **Reducción de costos y tiempo:** Demostrar que el modelo predictivo puede sustituir el uso de máquinas de Resonancia Magnética Nuclear (RMN) y cálculos avanzados de mecánica cuántica, reduciendo significativamente los costos y el tiempo necesarios para obtener estas medidas.

Al final del trabajo, se espera haber desarrollado una herramienta útil y accesible para los químicos, facilitando la investigación y el desarrollo en el ámbito de la química mediante la democratización del acceso a cálculos estructurales precisos.

### 3.2. Enfoque y metodología.

Para abordar el problema planteado, se adoptó un enfoque pragmático centrado en la comprensión y análisis de los datos disponibles, así como en la selección del modelo de regresión más óptimo para manejar un gran volumen de datos. Este enfoque permitió identificar rápidamente las mejores prácticas para desarrollar un modelo predictivo eficaz y eficiente.

Las metodologías específicas utilizadas para desarrollar el modelo predictivo y el resto del trabajo se basaron en la estructura de trabajos anteriores realizados durante el curso. Esto incluyó una revisión sistemática de las técnicas y procesos aprendidos, aplicándolos de manera adaptada al problema actual.

### 3.3. Planificación del Trabajo.

La planificación del trabajo siguió una estructura meticulosa para asegurar que todas las fases del proyecto se ejecutarán de manera eficiente y dentro del tiempo previsto. El proyecto se completó en un total de 10 días, con una distribución de tiempos optimizada para cubrir todas las etapas críticas del desarrollo.

- **Entender los datos** (2 días): Los primeros dos días se dedicaron a una profunda comprensión de los datos proporcionados, identificando las variables relevantes y el formato de los datos. Este paso fue crucial para establecer una base sólida para el resto del proyecto.
- **Crear un dataset conjunto** (3 días): Durante los siguientes dos días, se consolidaron los datos de diferentes fuentes en un único dataset cohesivo, asegurando que estuvieran limpios y listos para el análisis. Este proceso incluyó la eliminación de datos redundantes y la corrección de posibles inconsistencias.

- **Analizar los datos y su correlación** (2 días): En esta fase, se realizó un análisis exhaustivo de los datos para identificar patrones y relaciones entre las variables. Se utilizaron técnicas estadísticas y visualizaciones para obtener una comprensión detallada de la estructura de los datos.
- **Elegir el modelo para entrenar** (1 día): Basándose en el análisis previo, se seleccionó el modelo de regresión más adecuado para el dataset. Esta fase incluyó la evaluación de varios modelos para determinar cuál proporcionaba la mayor precisión y eficiencia.
- **Entrenamiento y evaluación del modelo** (1 día): Una vez seleccionado el modelo, se entrenó y evaluó utilizando técnicas de validación cruzada para asegurar su robustez y precisión. Esta fase también incluyó ajustes y optimizaciones para mejorar el rendimiento del modelo.
- **Crear una aplicación para usar el modelo predictivo** (1 día): Finalmente, se desarrolló una aplicación web utilizando Flask, que incluye un formulario para que los usuarios puedan ingresar datos y obtener predicciones sobre la constante de acoplamiento escalar. Esta fase incluyó el diseño y la implementación de la interfaz de usuario y la integración del modelo predictivo en la aplicación.

## 3.4. Recursos de sistemas utilizados.

Durante el desarrollo del proyecto, se utilizaron diversos recursos de hardware y software para asegurar un flujo de trabajo eficiente y efectivo.

### 3.4.1. Recursos de hardware:

El principal recurso de hardware utilizado fue mi PC personal, cuyas especificaciones se detallan a continuación mediante capturas. Este equipo proporcionó la capacidad de procesamiento necesaria para manejar el extenso conjunto de datos y realizar los cálculos requeridos.

**CPU** | Mainboard | Memory | SPD | Graphics | Bench | About

**Processor**

Name	AMD Ryzen 5 5600X		
Code Name	Vermeer	Max TDP	65.0 W
Package	Socket AM4 (1331)		
Technology	7 nm	Core Voltage	1.200 V



**Specification** AMD Ryzen 5 5600X 6-Core Processor

Family	F	Model	1	Stepping	0
Ext. Family	19	Ext. Model	21	Revision	VMR-B0

**Instructions** MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AES, AVX, AVX2, FMA3, SHA

**Clocks (Core #0)**

Core Speed	4577.34 MHz
Multiplier	x 46.5 (5.5 - 46.5)
Bus Speed	98.44 MHz
Rated FSB	

**Cache**

L1 Data	6 x 32 KBytes	8-way
L1 Inst.	6 x 32 KBytes	8-way
Level 2	6 x 512 KBytes	8-way
Level 3	32 MBytes	16-way

**Selection** Socket #1 | Cores: 6 | Threads: 12

**CPU** | Mainboard | **Memory** | SPD | Graphics | Bench | About

**General**

Type	DDR4	Channel #	2 x 64-bit
Size	32 GBytes	DC Mode	

**Uncore Frequency** 1047.7 MHz

**Timings**

DRAM Frequency	1050.4 MHz
FSB:DRAM	3:32
CAS# Latency (CL)	15.0 clocks
RAS# to CAS# Delay (tRCD)	15 clocks
RAS# Precharge (tRP)	15 clocks
Cycle Time (tRAS)	36 clocks
Bank Cycle Time (tRC)	51 clocks
Command Rate (CR)	2T
DRAM Idle Timer	
Total CAS# (tRDRAM)	
Row To Column (tRCD)	

**CPU** | Mainboard | Memory | **SPD** | Graphics | Bench | About

**Memory Slot Selection**

Slot #1	DDR4
Max Bandwidth	DDR4-3200 (1600 MHz)
Module Manuf.	Corsair
DRAM Manuf.	Micron Technology
Part Number	CMK16GX4M2E3200C16
Serial Number	

**Timings Table**

	JEDEC #16	JEDEC #17	JEDEC #18	XMP-3200
Frequency	1066 MHz	1066 MHz	1066 MHz	1600 MHz
CAS# Latency	22.0	23.0	24.0	16.0
RAS# to CAS#	15	15	15	20
RAS# Precharge	15	15	15	20
tRAS	36	36	36	38
tRC	50	50	50	58
Command Rate				
Voltage	1.20 V	1.20 V	1.20 V	1.350 V

**CPU** | Mainboard | Memory | SPD | **Graphics** | Bench | About

**Display Device Selection**

NVIDIA GeForce RTX 3060	Perf Level	Current
-------------------------	------------	---------

**GPU**

Name	NVIDIA GeForce RTX 3060		
Board Manuf.	Micro-Star International Co., Ltd. (MSI)		
Code Name	GA106-302	Revision	A1
Technology	8 nm	TDP	170.0 W



**Clocks**

GFX Core	210.0 MHz
Shader / SoC	
Memory	405.0 MHz

**Memory**

Size	12 GBytes
Type	GDDR6
Vendor	Samsung
Bus Width	192 bits

### 3.4.2. Recursos de software:

Para el desarrollo del proyecto, se emplearon varias herramientas y plataformas clave:

- **VSCode:** El editor de código Visual Studio Code fue utilizado como el entorno principal de desarrollo. Su integración con diversas extensiones facilitó el manejo del proyecto y la escritura del código.
- **Conda:** El entorno de Conda fue crucial para gestionar las dependencias y los entornos de trabajo. A través de Conda, se configuró un entorno específico para ejecutar el notebook de Jupyter.
- **Jupyter Notebook:** Utilizado para el desarrollo y la ejecución del análisis de datos y el entrenamiento del modelo predictivo.

### 3.4.3. Control de versiones y despliegue:

- **Git y GitHub:** Git fue utilizado como el sistema de control de versiones para rastrear los cambios en el código y colaborar en el proyecto. Los repositorios fueron alojados en GitHub, proporcionando un acceso centralizado y seguro al código fuente.
- **Heroku:** La aplicación web desarrollada se desplegó en Heroku, una plataforma en la nube que permite alojar aplicaciones web de manera sencilla. Heroku facilitó el despliegue y la gestión de la aplicación, asegurando que estuviera accesible para los usuarios.

# 4. Creación del set de datos

## 4.1. Estudio de datos necesarios.

Para abordar el problema de predecir la constante escalar de acoplamiento entre pares de átomos en moléculas, se requirió un conjunto de datos detallado que incluía diversas características de las moléculas y sus átomos. Los datos necesarios para este proyecto incluyeron información específica sobre los pares de átomos, sus índices, las constantes de acoplamiento y las estructuras moleculares. Los archivos utilizados fueron:

- **train.csv**: Contiene los datos de entrenamiento con los nombres de las moléculas, índices de átomos y las constantes de acoplamiento escalar.
- **test.csv**: Similar a train.csv pero sin la variable objetivo, usado para las pruebas del modelo.
- **structures.csv**: Información sobre la estructura molecular, incluyendo elementos atómicos y sus coordenadas cartesianas.
- **dipole\_moments.csv**: Momentos dipolares eléctricos moleculares.
- **magnetic\_shielding\_tensors.csv**: Tensores de apantallamiento magnético.
- **mulliken\_charges.csv**: Cargas mulliken de los átomos.
- **potential\_energy.csv**: Energía potencial de las moléculas.
- **scalar\_coupling\_contributions.csv**: Contribuciones específicas a la constante de acoplamiento escalar.

## 4.2. Fuente de datos.

El conjunto de datos utilizado proviene de una competición de Kaggle enfocada en predecir constantes de acoplamiento escalar en moléculas. Estos datos están organizados en varios archivos CSV, cada uno conteniendo información específica y complementaria sobre las moléculas y sus propiedades.

## 4.3. Recolección de los datos.

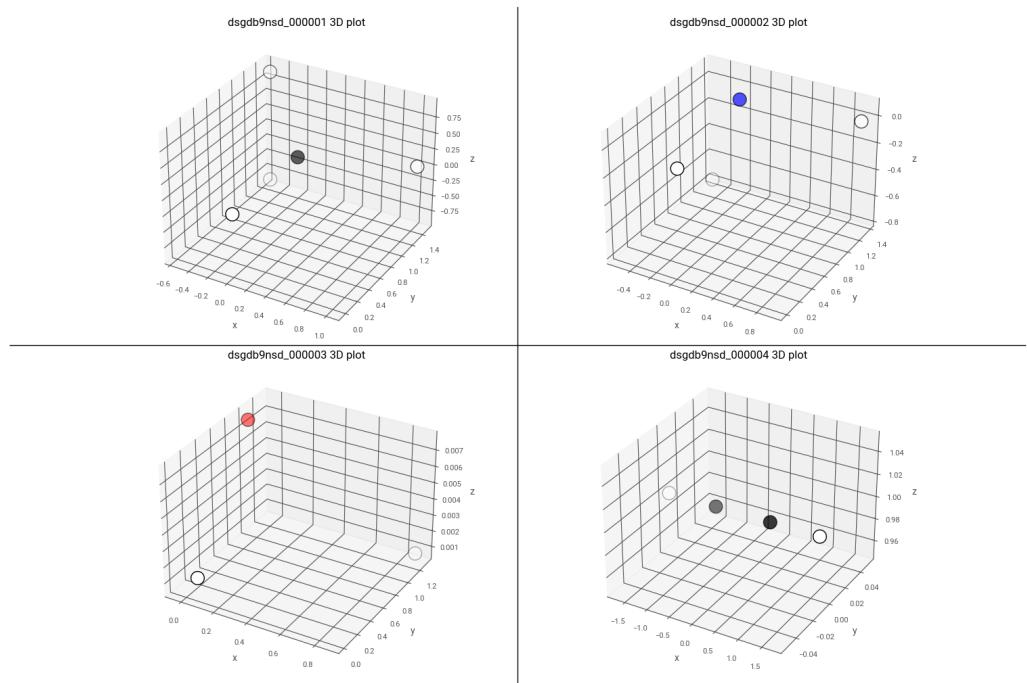
Los datos fueron descargados directamente desde la plataforma Kaggle. Los archivos relevantes fueron seleccionados y estudiados para comprender su estructura y contenido. Los archivos más importantes para el entrenamiento del modelo fueron:

- **train.csv**
- **structures.csv**
- **magnetic\_shielding\_tensors.csv**
- **mulliken\_charges.csv**
- **scalar\_coupling\_contributions.csv**

## 4.4. Visualización de datos.

Para visualizar y entender mejor los datos, se realizaron diversas técnicas de análisis y visualización. Esto incluyó la generación de gráficos y diagramas para observar las distribuciones de las constantes de acoplamiento escalar, correlaciones entre las variables y la estructura tridimensional de las moléculas. Estas visualizaciones ayudaron a identificar patrones y relaciones clave que serían útiles para el desarrollo del modelo predictivo.

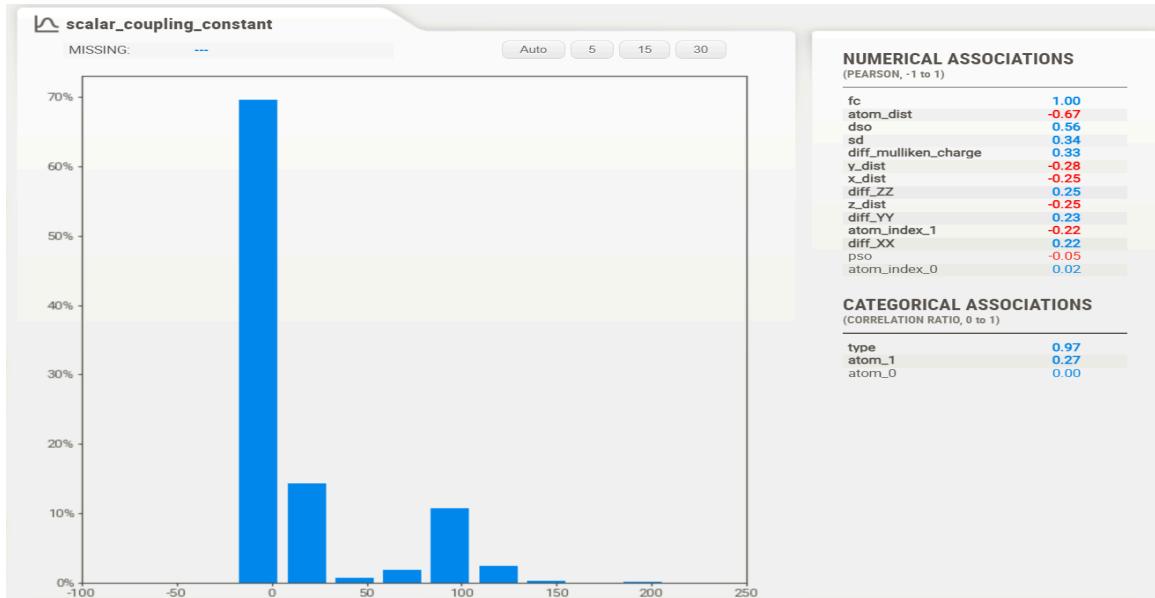
#### 4.4.1. Distribución espacial de los átomos



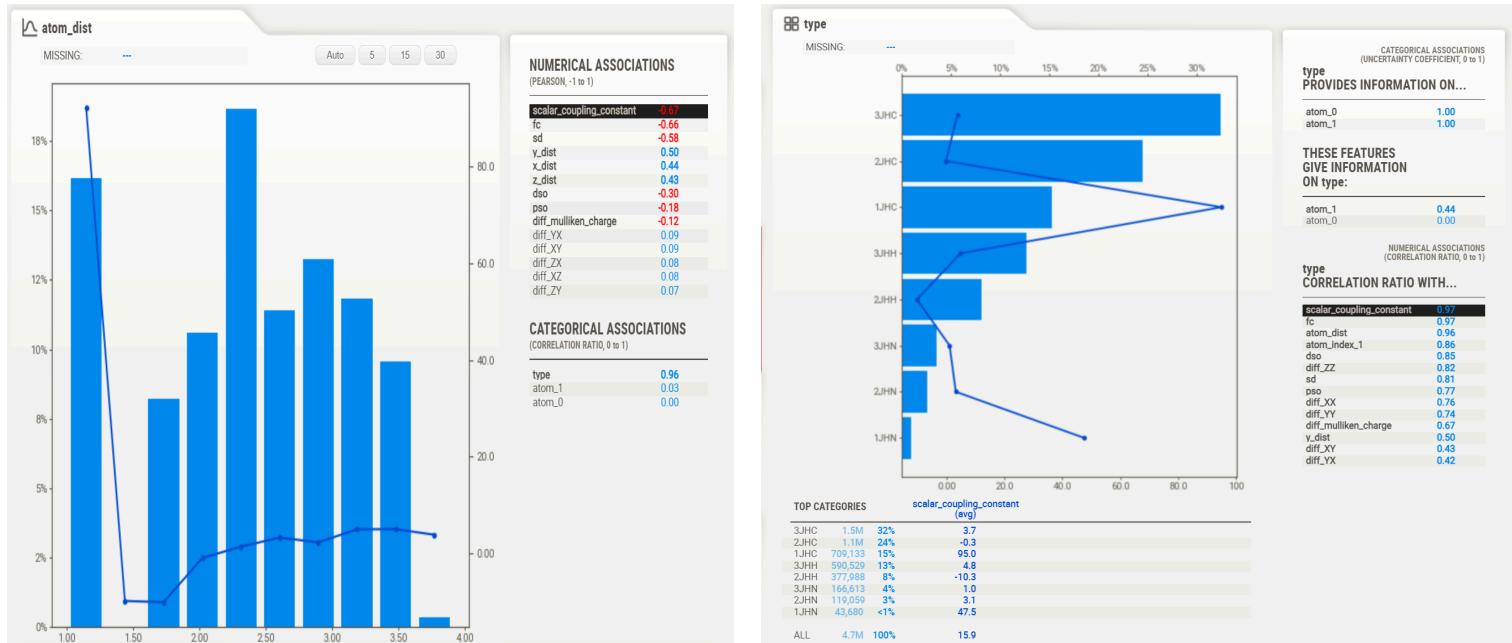
Como podemos observar, los átomos están bastante bien representados, por tanto de esto podemos concluir que los datos espaciales serán cruciales para la predicción de la constante de acoplamiento escalar ya que este valor tiene mucho que ver con la estructura molecular.

## 4.4.2. Densidades de las características comparada con el target

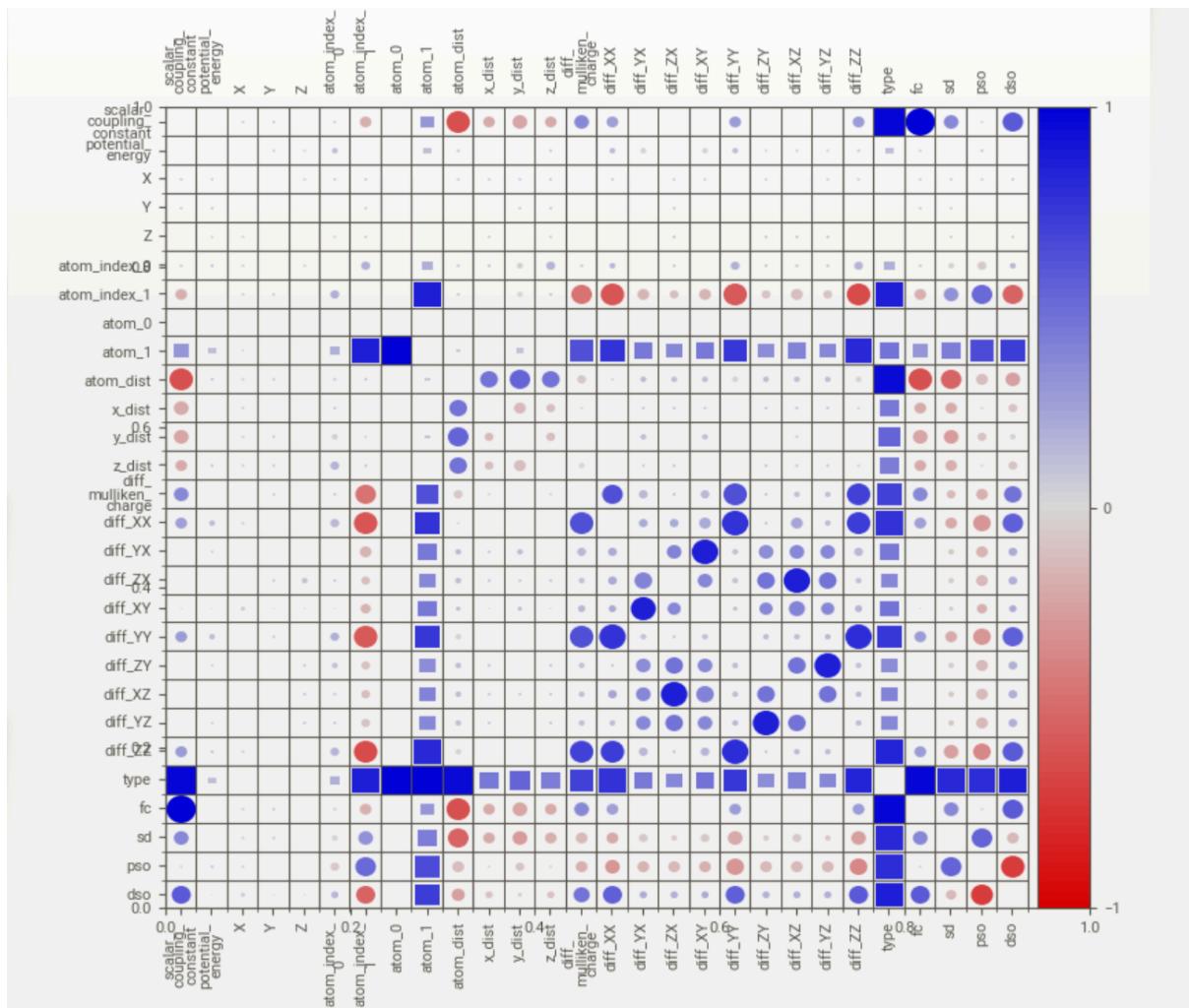
### 4.4.2.1. Densidad del Target



### 4.4.2.2. Densidades de las características más importantes



#### 4.4.3. Matriz de correlación



- Los cuadrados son asociaciones categóricas (coeficiente de incertidumbre y relación de correlación) de 0 a 1. El coeficiente de incertidumbre es asimétrico, (es decir, los valores de la ETIQUETA FILA indican en qué medida PROPORCIONAN INFORMACIÓN a cada ETIQUETA SUPERIOR).
- Los círculos son las correlaciones numéricicas simétricas (Pearson) de -1 a 1. La diagonal trivial se ha dejado en blanco intencionadamente para mayor claridad.

En la matriz podemos comprobar que los valores moleculares no son representativos para el target. Por otro lado, el tipo de enlace y las diferencias entre los tensores si que son relevantes.

En resumen, la preparación del conjunto de datos fue un proceso crucial que implicó la recolección, limpieza y unificación de datos provenientes de múltiples archivos. Estos datos proporcionaron la base necesaria para entrenar y evaluar el modelo predictivo con precisión y eficiencia.

## 5. Optimización, normalización y calidad del set de datos.

### 5.1. Técnicas de optimización y normalización

Para garantizar que los datos estuvieran en una forma adecuada para el modelo predictivo, se aplicaron varias técnicas de optimización y normalización:

- **Label Encoder:** Esta técnica se utilizó para convertir las variables categóricas en valores numéricos, lo que permite al modelo interpretarlas correctamente. Este proceso es esencial para trabajar con datos categóricos en algoritmos de aprendizaje automático.

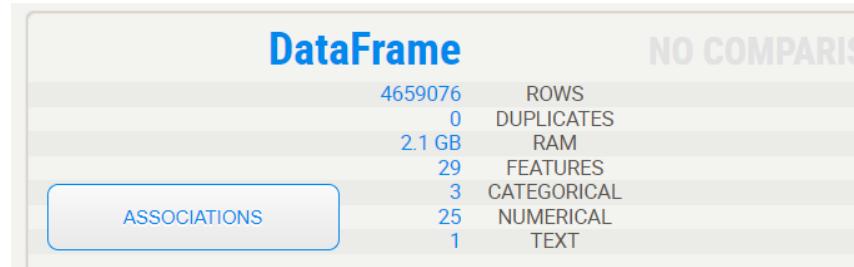
```
1 # Detectar columnas categóricas
2 categorical_cols = df.select_dtypes(include=['object']).columns
3
4 # Diccionario para almacenar los LabelEncoders
5 label_encoders = {}
6
7 for col in categorical_cols:
8     le = LabelEncoder()
9     df[col] = le.fit_transform(df[col])
10    label_encoders[col] = le
11
12 # Guardar los LabelEncoders en un archivo
13 with open('web/label_encoders.pkl', 'wb') as file:
14     pickle.dump(label_encoders, file)
```

- **StandardScaler:** Se aplicó la estandarización a los datos para que todas las características tuvieran una media de cero y una desviación estándar de uno. Esto es importante para asegurar que todas las variables contribuyan de manera equitativa al modelo y mejorar la eficiencia del algoritmo de aprendizaje.

```
1 targets_columns =['scalar_coupling_constant']
2
3 features = df.drop(columns=targets_columns)
4 target = df[targets_columns]
5
6 scaler = StandardScaler()
7 scaled_features = scaler.fit_transform(features)
8
9 # Convertir de nuevo a DataFrame
10 scaled_df = pd.DataFrame(scaled_features, columns=features.columns)
11 scaled_df[targets_columns] = target
```

## 5.2. Aseguramiento de la calidad e integridad de los datos

La calidad y la integridad de los datos fueron verificadas a través de un análisis exhaustivo. Durante este proceso se observó que no había valores nulos en el conjunto de datos, eliminando la necesidad de manejar datos faltantes. Además, dado el tipo de problema y la naturaleza de los datos, no fue necesario buscar ni tratar outliers, ya que todos los valores eran relevantes y precisos para las predicciones.



### 5.3. Selección de características

Para la selección de características se utilizaron tres modelos distintos para compararlos entre ellos.

- **Ridge Regression (Regresión Ridge)**: También conocida como regresión de crestas, es una técnica de regularización que añade una penalización L2 basada en la suma de los cuadrados de los coeficientes de los predictores al minimizar la función de pérdida.
- **Lasso Regression (Regresión Lasso)**: Lasso significa "Least Absolute Shrinkage and Selection Operator" y es una técnica de regularización que añade una penalización L1 basada en la suma de los valores absolutos de los coeficientes de los predictores.
- **ElasticNet**: Combina las penalizaciones L1 (de Lasso) y L2 (de Ridge) en una única función de pérdida.

	Característica	Ridge	Lasso	ElasticNet
0	atom_dist	-148.306841	-17.215655	-9.720183
12	diff_YZ	-0.601598	0.000000	0.000000
11	diff_XZ	-0.409115	0.000000	-0.000000
4	diff_mulliken_charge	-0.149561	6.940657	4.407578
8	diff_XY	0.020226	0.000000	0.000000
6	diff_YX	0.703702	0.000000	0.000000
9	diff_YY	0.751089	0.000000	0.661943
7	diff_ZX	0.795553	0.000000	0.000000
10	diff_ZY	0.889557	0.000000	0.000000
5	diff_XX	1.048185	0.185793	0.699842
13	diff_ZZ	1.414815	0.295017	0.974185
14	type	4.362495	-5.018139	-7.192785
3	z_dist	83.803589	0.058649	-0.933077
1	x_dist	85.537350	0.051807	0.957214
2	y_dist	93.116311	0.000000	-1.160693

Tras varias pruebas con estas características, el modelo entrena mejor con todas ellas y no supone mucha diferencia en el tiempo de entrenamiento.

#### 5.4. Desafíos relacionados con la calidad de los datos

Afortunadamente, los datos utilizados eran de alta calidad y no presentaron desafíos significativos en términos de integridad o precisión. Esto facilitó el proceso de preparación de los datos y permitió enfocar los esfuerzos en la construcción y optimización del modelo predictivo.

## 6. Desarrollo del modelo predictivo

### 6.1. Justificación del modelo seleccionado.

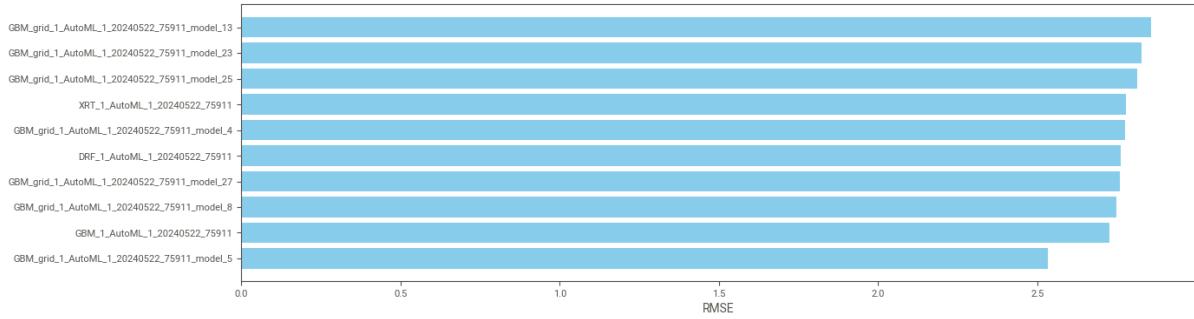
Para abordar el problema de predecir la constante escalar de acoplamiento entre pares de átomos, se utilizó la herramienta H2O AutoML. La elección de H2O AutoML fue motivada principalmente por la gran dimensión del conjunto de datos, que contenía más de 4.5 millones de filas. H2O AutoML es una poderosa herramienta que automatiza el proceso de selección y ajuste de modelos de aprendizaje automático, lo que resulta extremadamente útil cuando se trabaja con grandes volúmenes de datos.

El proceso de H2O AutoML implica la ejecución de múltiples algoritmos de aprendizaje automático y su comparación para seleccionar el mejor modelo basado en métricas de

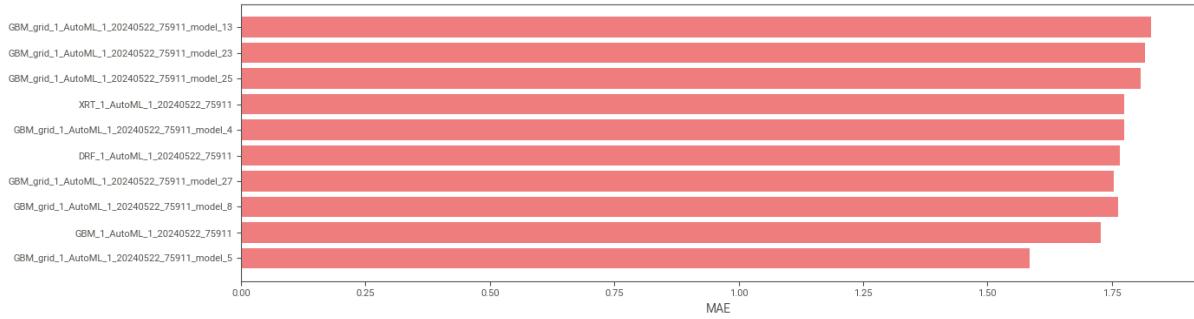
rendimiento. En este caso, se compararon los primeros 10 modelos generados por H2O AutoML y se seleccionó el modelo que ofreció el mejor rendimiento en términos de error absoluto medio (MAE).

	Model	RMSE	MAE
0	GBM_grid_1_AutoML_1_20240522_75911_model_5	2.533652	1.585266
1	GBM_1_AutoML_1_20240522_75911	2.724599	1.727424
2	GBM_grid_1_AutoML_1_20240522_75911_model_8	2.747204	1.762974
3	GBM_grid_1_AutoML_1_20240522_75911_model_27	2.757648	1.753213
4	DRF_1_AutoML_1_20240522_75911	2.761754	1.766038
5	GBM_grid_1_AutoML_1_20240522_75911_model_4	2.774924	1.774663
6	XRT_1_AutoML_1_20240522_75911	2.776546	1.775152
7	GBM_grid_1_AutoML_1_20240522_75911_model_25	2.811998	1.807807
8	GBM_grid_1_AutoML_1_20240522_75911_model_23	2.825702	1.816055
9	GBM_grid_1_AutoML_1_20240522_75911_model_13	2.855322	1.828252

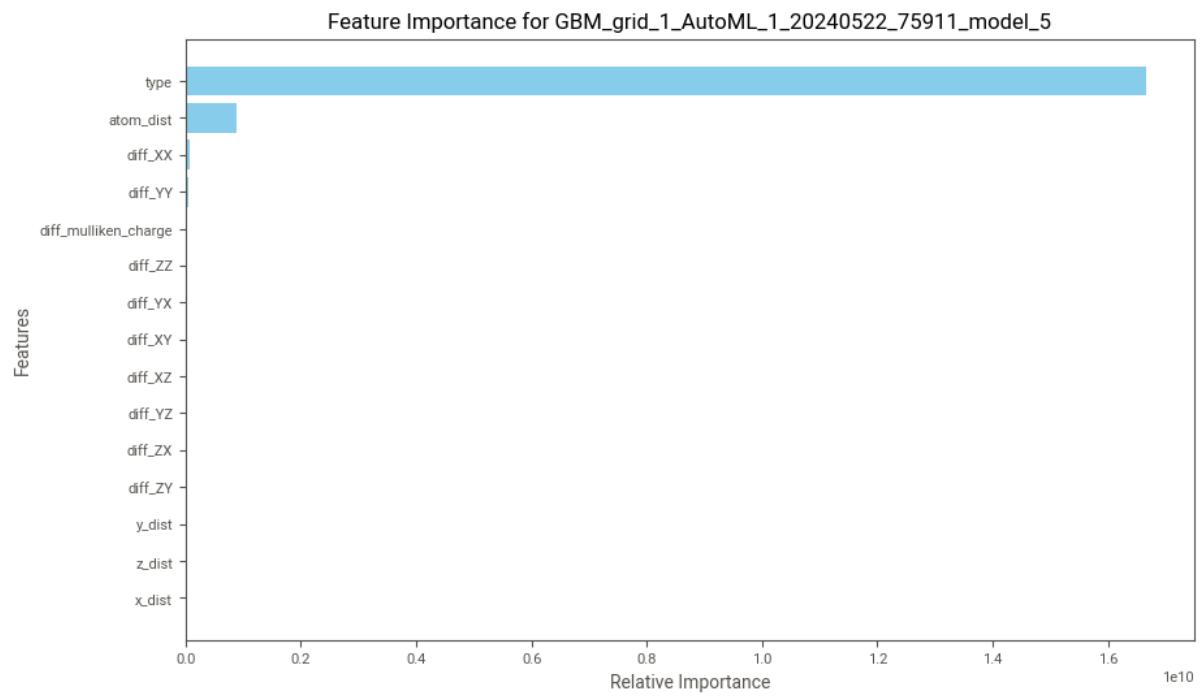
Comparación de RMSE para los 10 mejores modelos

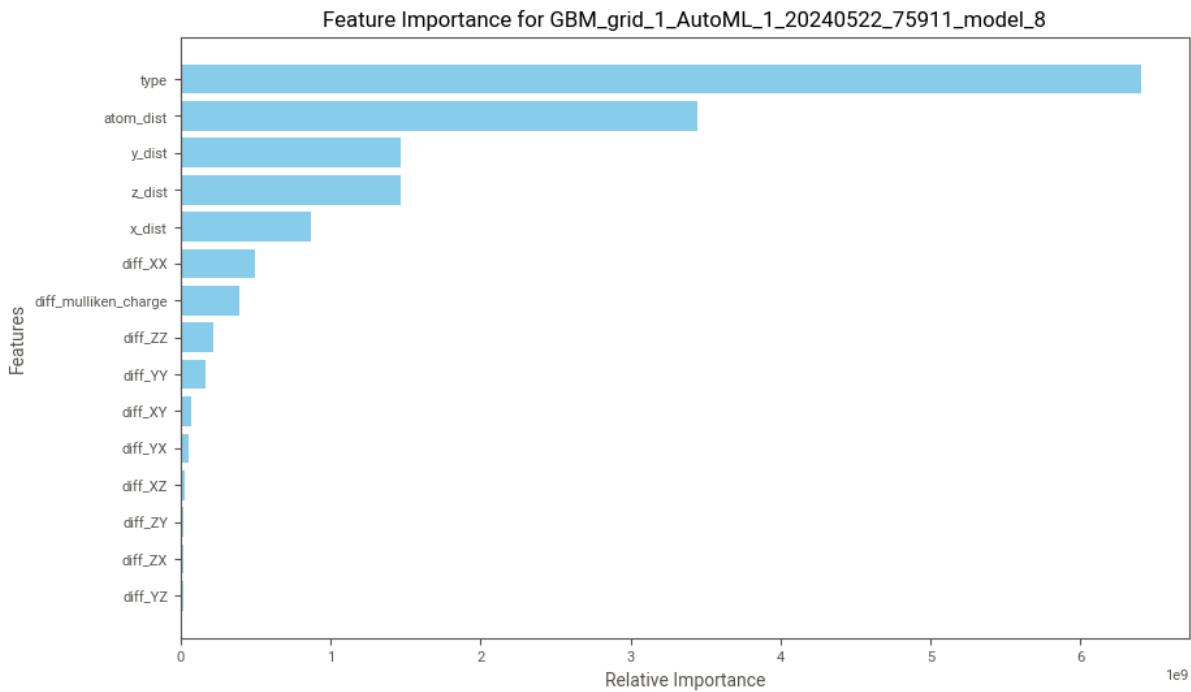


Comparación de MAE para los 10 mejores modelos



También se compararon la importancia de las características en cada uno de los modelos:





## 6.2. Descripción del modelo.

El modelo seleccionado a través de H2O AutoML es un modelo AutoML, el cual funciona de la siguiente manera:

- **Entrenamiento y Validación Automáticos:** H2O AutoML realiza un entrenamiento y validación automáticos de varios modelos utilizando técnicas como validación cruzada y partición de datos de entrenamiento y prueba.
- **Algoritmos Múltiples:** Integra múltiples algoritmos de aprendizaje automático, incluyendo regresión lineal, árboles de decisión, gradient boosting machines (GBM), redes neuronales profundas, entre otros.
- **Stacking y Ensemble Learning:** AutoML combina varios modelos individuales mediante técnicas de ensamblaje (stacking), creando modelos de ensamblaje que generalmente mejoran el rendimiento al combinar las predicciones de múltiples modelos.
- **Optimización de Hiperparámetros:** AutoML ajusta automáticamente los hiperparámetros de los modelos para optimizar el rendimiento.

El modelo final elegido fue el que mostró el mejor desempeño basado en las métricas de evaluación proporcionadas por H2O AutoML.

## 7. Entrenamiento y evaluación del modelo predictivo.

El proceso de entrenamiento y evaluación del modelo predictivo se realizó utilizando H2O AutoML, que automatiza la configuración y optimización del modelo.

### 7.1. Configuración del proceso de entrenamiento

H2O AutoML fue configurado para manejar el gran volumen de datos del proyecto. Esta herramienta ejecutó múltiples algoritmos de aprendizaje automático, incluyendo regresión linear, gradient boosting machines (GBM), y redes neuronales profundas, entre otros. AutoML también implementó validación cruzada y partición de datos para asegurar la robustez del modelo.

```

1 # Inicializa el clúster de H2O
2 h2o.init(min_mem_size="6G", max_mem_size="12G")
3
4 # Convierte el DataFrame de pandas a un H2OFrame
5 data = h2o.H2OFrame(scaled_df)
6
7 # Define la columna de objetivo y las características
8 y = "scalar_coupling_constant"
9 x = data.columns
10 x.remove(y)
11
12 # Divide el dataset en entrenamiento, prueba y validación
13 train, val, test = data.split_frame(ratios=[.6, .2])
14
15 # Inicializa y entrena el modelo de AutoML
16 aml = H2OAutoML(max_runtime_secs=3600, seed=1, nfolds=0)
17 aml.train(x=x, y=y, training_frame=train, validation_frame=val)
18
19 # Visualiza el líder del AutoML
20 lb = aml.leaderboard
21 print(lb)

```

## 7.2. Métricas de evaluación

Para evaluar el rendimiento de los modelos, se utilizó principalmente el Error Absoluto Medio (MAE). Esta métrica es crucial para problemas de regresión, ya que mide la media de los errores absolutos entre las predicciones del modelo y los valores reales.

## 7.3. Resultados obtenidos

Durante la fase de evaluación, el modelo que ofreció el mejor rendimiento mostró un MAE de 1.59. Este resultado fue obtenido tras comparar los primeros 10 modelos generados por H2O AutoML. La herramienta ajustó automáticamente los hiperparámetros y utilizó técnicas de ensamblaje para mejorar la precisión y robustez del modelo.

model_id		rmse	mse	mae	rmsle	mean_residual_de
GBM_grid_1_AutoML_1_20240522_75911_model_5	2.50884	6.2943	1.58055	nan	6	
GBM_1_AutoML_1_20240522_75911	2.6983	7.28081	1.72246	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_8	2.72077	7.4026	1.75766	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_27	2.72889	7.44684	1.7467	nan	7	
DRF_1_AutoML_1_20240522_75911	2.73379	7.4736	1.7606	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_4	2.74833	7.55334	1.76866	nan	7	
XRT_1_AutoML_1_20240522_75911	2.75045	7.56498	1.76995	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_25	2.78553	7.75918	1.80171	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_23	2.79853	7.83178	1.81105	nan	7	
GBM_grid_1_AutoML_1_20240522_75911_model_13	2.82126	7.95953	1.81947	nan	7	

## *Modelos entrenados*



## *Predicciones*

En resumen, H2O AutoML facilitó un proceso eficiente y automatizado para entrenar y evaluar múltiples modelos, seleccionando el que mejor rendimiento tuvo basado en métricas de evaluación rigurosas. Este enfoque permitió obtener un modelo predictivo altamente optimizado y listo para ser desplegado.

# 8. Informe de rendimiento y métricas en aula virtualizada y otros entornos.

## 8.1. Evaluación del rendimiento en diferentes entornos

Para evaluar el rendimiento del modelo predictivo, se realizaron pruebas en tres entornos distintos: mi PC personal, el ordenador de clase y un entorno virtualizado. Estas pruebas permitieron comparar la eficiencia y la efectividad del modelo en diferentes configuraciones de hardware y software.

### 8.1.1. Entornos evaluados

#### ➤ PC personal:

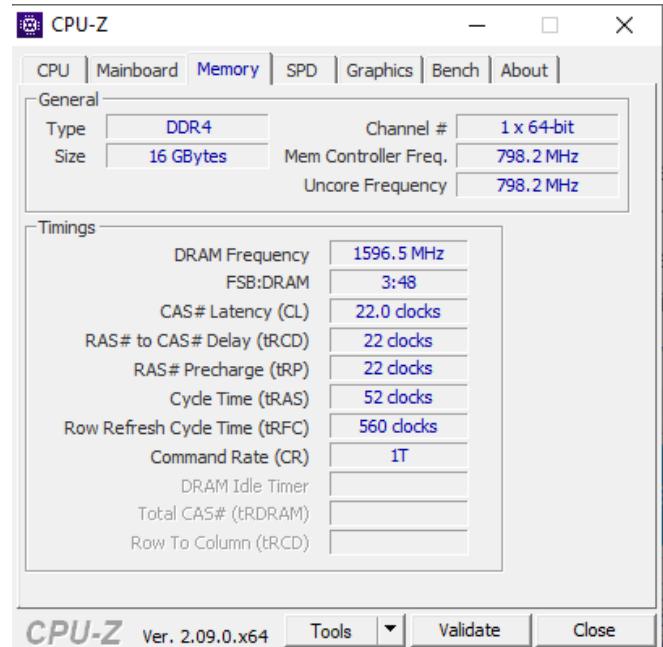
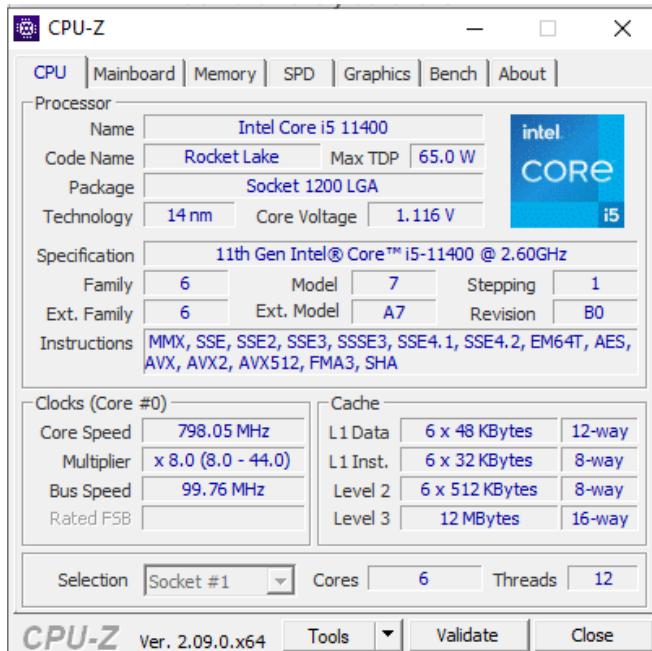
##### ○ Especificaciones:

CPU	Mainboard	Memory	SPD	Graphics	Bench	About
<b>Processor</b>						
Name	AMD Ryzen 5 5600X					
Code Name	Vermeer	Max TDP	65.0 W			
Package	Socket AM4 (1331)					
Technology	7 nm	Core Voltage	1.200 V			
Specification	AMD Ryzen 5 5600X 6-Core Processor					
Family	F	Model	1	Stepping	0	
Ext. Family	19	Ext. Model	21	Revision	VMR-B0	
Instructions	MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AES, AVX, AVX2, FMA3, SHA					
<b>Clocks (Core #0)</b>						
Core Speed	4577.34 MHz	Cache				
Multiplier	x 46.5 (5.5 - 46.5)	L1 Data	6 x 32 KBytes	8-way		
Bus Speed	98.44 MHz	L1 Inst.	6 x 32 KBytes	8-way		
Rated FSB		Level 2	6 x 512 KBytes	8-way		
		Level 3	32 MBytes	16-way		
Selection	Socket #1	Cores	6	Threads	12	
<b>Memory</b>						
Type	DDR4	Channel #	2 x 64-bit			
Size	32 GBytes	DC Mode				
		Uncore Frequency	1047.7 MHz			
<b>Timings</b>						
DRAM Frequency	1050.4 MHz					
FSB:DRAM	3:32					
CAS# Latency (CL)	15.0 clocks					
RAS# to CAS# Delay (tRCD)	15 clocks					
RAS# Precharge (tRP)	15 clocks					
Cycle Time (tRAS)	36 clocks					
Bank Cycle Time (tRC)	51 clocks					
Command Rate (CR)	2T					
DRAM Idle Timer						
Total CAS# (tRDRAM)						
Row To Column (tRCD)						

- Tiempo de ejecución: 01:14:02

➤ Ordenador de clase:

- Especificaciones:



- Tiempo de ejecución: 01:15:37

➤ **Entorno virtualizado (Velorcios Cloud):**

- Especificaciones:

The screenshot shows two windows of the CPU-Z application. The left window displays processor specifications: Name (Intel Xeon DP), Code Name (Dempsey), Package (Socket 771 LGA), Technology (65 nm), and Instructions (MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, AES, AVX, AVX2, AVX512, FMA3). It also shows clock speeds (Core Speed: 1004.78 MHz, Multiplier: x 1.0, Bus Speed: 1005.39 MHz, Rated FSB: 4019.11 MHz) and cache details (L1 Data: 4 x 32 KBytes, 8-way; L1 Inst.: 4 x 32 KBytes, 8-way; Level 2: 2 MBytes, 8-way; Level 3: [empty]). The right window displays memory specifications: Type (FPM), Size (32 GBytes), Channel #, DC Mode, and Uncore Frequency. Timings are listed as DRAM Frequency (1006.1 MHz), Read Burst Rate (DRBT) (x444), Write Burst Rate (DWBT) (x444), RAS# to CAS# Delay (tRCD) (0 clocks), RAS# Precharge (tRP) (3 clocks), Cycle Time (tRAS), Bank Cycle Time (tRC), Command Rate (CR), DRAM Idle Timer, Total CAS# (tRDRAM), and Row To Column (tRCD).

- Tiempo de ejecución: 01:43:21

## 8.2. Métricas utilizadas

Las principales métricas utilizadas para evaluar el rendimiento en estos entornos fueron:

- **Tiempo de Ejecución:** Se midió el tiempo total que tomó ejecutar el cuaderno completo en cada uno de los entornos. Esta métrica es crucial para entender la eficiencia del modelo en diferentes configuraciones de hardware.
- **Uso de Recursos:** Se monitorea el uso de CPU y memoria durante la ejecución del cuaderno para evaluar la carga de trabajo en cada entorno.
- **Estabilidad del Sistema:** Se observó la estabilidad del sistema durante la ejecución, anotando cualquier fallo o interrupción que ocurriera.

### 8.3. Resultados y hallazgos clave

#### ➤ **PC personal:**

- Uso de Recursos: El sistema utilizó el 100% de la CPU y hasta el 80% de la RAM durante el entrenamiento.
- Estabilidad: El sistema se mantuvo estable durante toda la ejecución sin interrupciones.

#### ➤ **Ordenador de clase:**

- Uso de Recursos: El sistema también utilizó el 100% de la CPU y hasta el 80% de la RAM durante el entrenamiento.
- Estabilidad: Se observó una ligera carga adicional en la CPU, pero el sistema se mantuvo estable.

#### ➤ **Entorno virtualizado (Velorcios Cloud):**

- Uso de Recursos: El entorno virtualizado manejó la carga de trabajo con el 100% de uso de la CPU y hasta el 80% de la RAM.
- Estabilidad: Aunque el tiempo de ejecución fue superior, el entorno virtualizado se mantuvo estable y no presentó interrupciones.

### 8.4. Conclusiones

Los resultados muestran que el rendimiento del modelo predictivo varía según el entorno en el que se ejecuta. Mi PC personal, con sus especificaciones más avanzadas, logró el menor tiempo de ejecución y uso eficiente de recursos. El ordenador de clase tuvo un rendimiento similar, con una ligera diferencia en el tiempo de ejecución. El entorno virtualizado presentó un tiempo de ejecución más alto, pero manejó la carga de trabajo de manera eficiente, lo que lo hace viable para usos en entornos donde la virtualización es necesaria.

Estos hallazgos subrayan la importancia de considerar las especificaciones del hardware y la configuración del entorno al implementar modelos predictivos en diferentes contextos.

Aunque todos los sistemas alcanzaron el 100% de uso de CPU y el 80% de uso de RAM, la estabilidad se mantuvo en todos los casos, garantizando la confiabilidad del modelo en diferentes configuraciones.

## 9. Puesta en producción (API y aplicación cliente).

La puesta en producción del modelo predictivo involucró la implementación de una API y una aplicación cliente. Para lograr esto, se utilizaron diversas tecnologías y herramientas.

### 9.1. Implementación de la API y la aplicación cliente

La API fue implementada utilizando Flask, un microframework de Python que facilita la creación de aplicaciones web. La interfaz de usuario se desarrolló con un paquete básico de HTML, CSS y JavaScript, proporcionando una plataforma sencilla pero efectiva para que los usuarios interactúen con el modelo predictivo.

### 9.2. Tecnologías utilizadas para el desarrollo y despliegue

- **Flask:** Utilizado para crear la API que maneja las solicitudes del formulario web, procesando los datos de entrada y devolviendo las predicciones.
- **HTML, CSS y JavaScript:** Estos lenguajes fueron utilizados para construir la interfaz de usuario, creando un formulario web donde los usuarios pueden ingresar los datos necesarios para la predicción.
- **Heroku:** Plataforma de despliegue utilizada para alojar tanto la API como la aplicación cliente. Heroku permite la fácil implementación de aplicaciones

web y APIs, facilitando el acceso a los usuarios finales. Heroku se conecta directamente al repositorio de GitHub, lo que facilita el despliegue continuo y la integración de nuevas versiones de la aplicación.

### 9.3. Flujo de trabajo

El flujo de trabajo desde la entrada de datos en el formulario hasta la obtención de la predicción es el siguiente:

- **Ingreso de Datos:** El usuario ingresa los datos necesarios a través del formulario web.
- **Envío de Datos:** Los datos son enviados a la API de Flask.
- **Procesamiento:** La API de Flask recibe los datos, los procesa para asegurarse de que estén en el formato adecuado para el modelo predictivo, y los pasa al modelo.
- **Predicción:** El modelo predictivo procesa los datos y genera una predicción para la constante de acoplamiento escalar.
- **Devolución de Resultados:** La API devuelve un archivo CSV que contiene tanto los datos introducidos por el usuario como la predicción generada. Este archivo es descargado por el usuario, proporcionándole un registro claro y utilizable de la predicción junto con los datos iniciales.

### 9.4. Desafíos en el despliegue

Inicialmente, se intentó desplegar la aplicación en otros servicios como Render. Sin embargo, surgieron problemas debido a la incompatibilidad con la versión de Java necesaria para ejecutar el modelo H2O. Estos problemas se resolvieron al migrar el despliegue a Heroku, que permitió instalar y ejecutar correctamente la versión de Java

requerida. Además, Heroku se conecta al repositorio de GitHub, lo que facilita la integración y el despliegue continuo. Cada vez que se realiza un push en el repositorio, Heroku automáticamente actualiza la aplicación en producción, asegurando que las últimas versiones y mejoras estén siempre disponibles.



### [Web Predicción de la constante de acoplamiento escalar](#)

En resumen, la implementación y despliegue de la API y la aplicación cliente se realizaron utilizando tecnologías robustas y adecuadas para el proyecto, superando desafíos iniciales y asegurando una puesta en producción exitosa.

# 10. Conclusiones y mejoras a realizar, modelos alternativos.

## 10.1. Conclusiones

El desarrollo de un modelo predictivo para calcular la constante escalar de acoplamiento ha demostrado ser una alternativa viable y económica frente a los métodos tradicionales como la Resonancia Magnética Nuclear (RMN) y los cálculos de mecánica cuántica. Utilizando H2O AutoML, se logró seleccionar y optimizar un modelo que mostró un rendimiento sólido con un error absoluto medio (MAE) de 1,59. La implementación de este modelo en una aplicación web, desplegada en Heroku, permite a los usuarios ingresar datos y obtener predicciones rápidamente, lo cual representa una significativa reducción en costos y tiempo.

## 10.2. Mejoras a realizar

A pesar de los resultados positivos, hay varias áreas en las que se podrían realizar mejoras para incrementar aún más la precisión y eficiencia del modelo:

Aumento del volumen de datos de entrenamiento: Incluir más datos de entrenamiento podría mejorar la capacidad del modelo para generalizar y hacer predicciones más precisas.

- **Feature Engineering avanzado:** Investigar y desarrollar características adicionales a partir de los datos existentes podría ayudar a mejorar el rendimiento del modelo.
- **Optimización continua del modelo:** Explorar técnicas de optimización de hiperparámetros y ajustes finos de los modelos existentes podría resultar en mejoras adicionales en la precisión.

- **Implementación de técnicas de ensamblaje más avanzadas:** Combinar múltiples modelos utilizando técnicas más sofisticadas de ensamblaje podría mejorar el rendimiento general del sistema.

### 10.3. Modelos alternativos

Si bien H2O AutoML fue la herramienta elegida para este proyecto debido a su capacidad de manejar grandes volúmenes de datos y automatizar la selección y optimización de modelos, existen otros enfoques y modelos que podrían ser explorados en futuras iteraciones:

- **Redes Neuronales Profundas (Deep Learning):** Modelos más complejos como las redes neuronales profundas podrían capturar relaciones no lineales y complejidades en los datos que otros modelos no pueden.
- **Máquinas de Soporte Vectorial (SVM):** Las SVM son otra opción que podría ofrecer buenos resultados, especialmente con conjuntos de datos bien estructurados y normalizados.
- **Modelos basados en Árboles de Decisión:** Métodos como Random Forests o Gradient Boosting Machines (GBM) han demostrado ser efectivos en muchos problemas de regresión y clasificación y podrían ser explorados más a fondo.
- **Modelos de Regresión Bayesianos:** Estos modelos podrían proporcionar una estimación más robusta y manejable de la incertidumbre en las predicciones.

En resumen, el proyecto ha cumplido sus objetivos, pero hay oportunidades para mejorar y explorar modelos alternativos. Implementar estas mejoras y nuevas metodologías fortalecerá la capacidad del modelo para realizar predicciones precisas y útiles en química.

# 11. Bibliografía.

- ❖ [Competición](#)
- ❖ [Cuaderno de guía](#)
- ❖ [Heroku](#)
- ❖ [Github](#)
- ❖ [Web Producción](#)
- ❖ [ChatGPT](#)
- ❖ [LibreTexs Espanol](#)