Entrega 2

Introducción a la inteligencia artificial

Andres David Medina Herrera, Miguel Arias Londoño

cc1003562902, cc1017272204

Facultad de ingeniería, Universidad de Antioquia Sede Medellín

andres.medina2@udea.edu.co - miguel.ariasl@udea.edu.co

22/10/2023

La constante de acoplamiento escalar es un parámetro en espectroscopia que indica la interacción entre núcleos atómicos o partículas subatómicas. En teoría de campos, física de partículas, física cuántica y demás ciencias exactas relacionadas con lo nuclear, es usado para realizar predicciones y cálculos, ya que guarda estrecha relación con la distancia entre núcleos atómicos, ángulos de enlace, configuración estereoquímica, etc.}

Actualmente existen técnicas que permiten calcular tal constante de acoplamiento, pero debido a la complejidad y costos de estos procedimientos, resultan considerablemente limitantes en los flujos de trabajo del día a día. Partiendo de ello, el presente trabajo busca obtener predicciones de la constante de acoplamiento a partir de algoritmos de inteligencia artificial. Para ello se usaran datos provenientes de kaggle, de la competición "*Predicting Molecular Properties".* (https://www.kaggle.com/competitions/champs-scalar-coupling/overview). Con los datasets presentes en la competición se espera tener la información suficiente para crear un modelo capaz de predecir la constante de acoplamiento escalar entre los diferentes pares de átomos presentes en las diferentes moléculas proporcionadas en los datos.

Descripción y estructura de los datasets:

La competición da los datos en un zip, el cual contenía los siguientes archivos:

- potential_energy.csv
- magnetic_shielding_tensors.csv
- structures.csv
- scalar_coupling_contributions.csv
- test.csv
- dipole_moments.csv
- train.csv

- sample_submission.csv
- mulliken_charges.csv

Cada uno de los datasets contiene información importante de las diferentes moléculas, tal información se encuentra dispersa en cada uno de los CSV, por lo cual se deberá entender y estudiar cada archivo, para disponer de la data adecuadamente.

potential_energy.csv:

Este data frame contiene información sobre la energía potencial presente en la molécula.

	molecule_name	potential_energy
0	dsgdb9nsd_000001	-40.523680
1	dsgdb9nsd_000002	-56.560246
2	dsgdb9nsd_000003	-76.426077
3	dsgdb9nsd_000004	-77.335268
4	dsgdb9nsd_000005	-93.428488
5	dsgdb9nsd_000006	-114.510216

Figura 1. Dataframe potencial_energy.csv

Como se observa en la figura 1, este posee dos filas, una que identifica la molécula y la respectiva energía potencial de esta. Posee en total 130789 filas, por lo cual tendremos ese mismo número de moléculas para análisis.

magnetic_shielding_tensors.csv:

Este data frame contiene información sobre los tensores de blindaje magnéticos, los cuales físicamente hablando, en términos simples son como un “escudo” que rodea el núcleo atómico y protege a los electrones circundantes de los efectos del campo magnético externo, guardando cierta relación con la constante de acoplamiento.

	molecule_name	atom_index	XX	YX	ZX	XY	YY	ZY	XZ	YZ	ZZ
0	dsgdb9nsd_000001	0	195.3147	0.0000	-0.0001	0.0000	195.3171	0.0007	-0.0001	0.0007	195.3169
1	dsgdb9nsd_000001	1	31.3410	-1.2317	4.0544	-1.2317	28.9546	-1.7173	4.0546	-1.7173	34.0861
2	dsgdb9nsd_000001	2	31.5814	1.2173	-4.1474	1.2173	28.9036	-1.6036	-4.1476	-1.6036	33.8967
3	dsgdb9nsd_000001	3	31.5172	4.1086	1.2723	4.1088	33.9068	1.6950	1.2724	1.6951	28.9579
4	dsgdb9nsd_000001	4	31.4029	-4.0942	-1.1793	-4.0944	34.0776	1.6259	-1.1795	1.6260	28.9013
5	dsgdb9nsd_000002	0	275.6345	0.0003	0.0133	0.0003	275.6364	-0.0003	0.0161	-0.0004	237.4966

Figura 2. magnetic_shielding_tensor.csv

Este data frame contiene 2358875 filas, ya que contiene información de cada átomo presente en cada una de las moléculas.

structures.csv:

El dataframe de estructuras posee información de la molécula, como el tipo de átomos presente en esta y la ubicaciones de los mismos, al tiempo que da un índice a cada uno de los átomos respecto a la molécula.

	molecule_name	atom_index	atom	x	y	z
0	dsgdb9nsd_000001	0	C	-0.012698	1.085804	0.008001
1	dsgdb9nsd_000001	1	H	0.002150	-0.006031	0.001976
2	dsgdb9nsd_000001	2	H	1.011731	1.463751	0.000277
3	dsgdb9nsd_000001	3	H	-0.540815	1.447527	-0.876644
4	dsgdb9nsd_000001	4	H	-0.523814	1.437933	0.906397
5	dsgdb9nsd_000002	0	N	-0.040426	1.024108	0.062564
6	dsgdb9nsd_000002	1	H	0.017257	0.012545	-0.027377

Figura 3. structure.csv

Al poseer información de cada unos de los átomos de cada una de las moléculas, es de esperarse que este data frame tenga el mismo número de filas que el anterior, 2358875.

scalar_coupling_contributions.csv:

Este data frame me da información sobre las contribuciones de acoplamiento escalar, estas contribuciones se relacionan con la interacción entre los núcleos atómicos en una misma molécula, lo que proporciona información valiosa sobre la estructura y la conectividad de los átomos en la molécula, siendo entonces parámetros que no se pueden ignorar para el modelo. Entre las filas se tiene: sd, el cual se refiere al acoplamiento escalar; fc, acoplamiento escalar de contacto fermi; pso, acoplamiento escalar paramagnético spin-orbit; y el dso, acoplamiento escalar diamagnético spin-orbit.

	molecule_name	atom_index_0	atom_index_1	type	fc	sd	pso	dso
0	dsgdb9nsd_000001	1	0	1JHC	83.02240	0.254579	1.258620	0.272010
1	dsgdb9nsd_000001	1	2	2JHH	-11.03470	0.352978	2.858390	-3.433600
2	dsgdb9nsd_000001	1	3	2JHH	-11.03250	0.352944	2.858520	-3.433870
3	dsgdb9nsd_000001	1	4	2JHH	-11.03190	0.352934	2.858550	-3.433930
4	dsgdb9nsd_000001	2	0	1JHC	83.02220	0.254585	1.258610	0.272013
5	dsgdb9nsd_000001	2	3	2JHH	-11.03170	0.352932	2.858560	-3.433950
6	dsgdb9nsd_000001	2	4	2JHH	-11.03240	0.352943	2.858530	-3.433870
7	dsgdb9nsd_000001	3	0	1JHC	83.02410	0.254634	1.258560	0.272012
8	dsgdb9nsd_000001	3	4	2JHH	-11.03190	0.352943	2.858560	-3.433930
9	dsgdb9nsd_000001	4	0	1JHC	83.02430	0.254628	1.258560	0.272012
10	dsgdb9nsd_000002	1	0	1JHN	30.61160	0.059952	1.949350	0.067923

Figura 4.scalar_coupling_contributions.csv

Como se observa en la figura 4 se permutan cada uno de los átomos en la molécula con los otros, es decir que habrá tantas filas por molécula como el número de permutaciones posibles entre sus átomos, siendo en total 4659075 filas.

dipole_moments:

Este data frame posee información de la molécula y del momento dipolar de la misma descompuesto en las coordenadas cartesianas. Representa la distribución de cargas eléctricas en una molécula y la diferencia entre las cargas positivas y negativas en la misma.

	molecule_name	X	Y	Z
0	dsgdb9nsd_000001	0.0000	0.0000	0.0000
1	dsgdb9nsd_000002	-0.0002	0.0000	1.6256
2	dsgdb9nsd_000003	0.0000	0.0000	-1.8511
3	dsgdb9nsd_000004	0.0000	0.0000	0.0000
4	dsgdb9nsd_000005	0.0000	0.0000	-2.8937
5	dsgdb9nsd_000006	-2.1089	0.0000	0.0000
6	dsgdb9nsd_000007	0.0000	0.0000	0.0000

Figura 5. dipole moments.csv

Como se observa en la figura 5, el momento dipolar es dado por molécula, por ende el data frame deberá tener 130789 filas.

mulliken_charges.csv:

Este parámetro entrega información sobre el valor de distribución de carga eléctrica de una molécula, teniendo estrecha relación con el momento dipolar.

	molecule_name	atom_index	mulliken_charge
0	dsgdb9nsd_000001	0	-0.535689
1	dsgdb9nsd_000001	1	0.133921
2	dsgdb9nsd_000001	2	0.133922
3	dsgdb9nsd_000001	3	0.133923
4	dsgdb9nsd_000001	4	0.133923
5	dsgdb9nsd_000002	0	-0.707143
6	dsgdb9nsd_000002	1	0.235712
7	dsgdb9nsd_000002	2	0.235712

Figura 6. mulliken_charges.csv

La figura 6 muestra las primeras 6 filas del data frame, donde se observa el valor de la variables mencionada por cada átomo de cada molécula, es decir teniendo en total 2358875 filas, el mismo número que structures y magnetic shielding, dataframes que dan información por átomos.

train.csv y test.csv:

Train y test contiene información de las moléculas, todas las posibles permutaciones de los átomos de la misma y la variables a predecir, constante escalar de acoplamiento (train).

	id	molecule_name	atom_index_0	atom_index_1	type	scalar_coupling_constant
0	0	dsgdb9nsd_000001	1	0	1JHC	84.80760
1	1	dsgdb9nsd_000001	1	2	2JHH	-11.25700
2	2	dsgdb9nsd_000001	1	3	2JHH	-11.25480
3	3	dsgdb9nsd_000001	1	4	2JHH	-11.25430
4	4	dsgdb9nsd_000001	2	0	1JHC	84.80740
5	5	dsgdb9nsd_000001	2	3	2JHH	-11.25410
6	6	dsgdb9nsd_000001	2	4	2JHH	-11.25480
7	7	dsgdb9nsd_000001	3	0	1JHC	84.80930

Figura 7. train.csv

Test tiene la misma organización de datos que train, exceptuando evidentemente la columna de la constante escalar de acoplamiento.

Iteraciones de desarrollo:

Una vez visualizados cada uno de los data frame que se tenían, analizadas sus columnas, filas, significado físico de las variables que daban, análisis de la importancia de las mismas, etc., se procedió con la selección y limpieza de los datos que se usarán para el entrenamiento del modelo.

Se optó por usar todas las variables que se tienen para el proceso de predicción, para lo cual primero se deben tener los datos organizados en un solo dataframe y no de manera dispersa:

Los dataframe de potencial_energy y dipole_moments contiene información por molécula, es decir que su número de filas será el mismo, así entonces, se creó una nueva tabla con la información de las dos.

	molecule_name	dipole_X	dipole_Y	dipole_Z	potential_energy
0	dsgdb9nsd_000001	0.0000	0.0	0.0000	-40.523680
1	dsgdb9nsd_000002	-0.0002	0.0	1.6256	-56.560246
2	dsgdb9nsd_000003	0.0000	0.0	-1.8511	-76.426077
3	dsgdb9nsd_000004	0.0000	0.0	0.0000	-77.335268
4	dsgdb9nsd_000005	0.0000	0.0	-2.8937	-93.428488

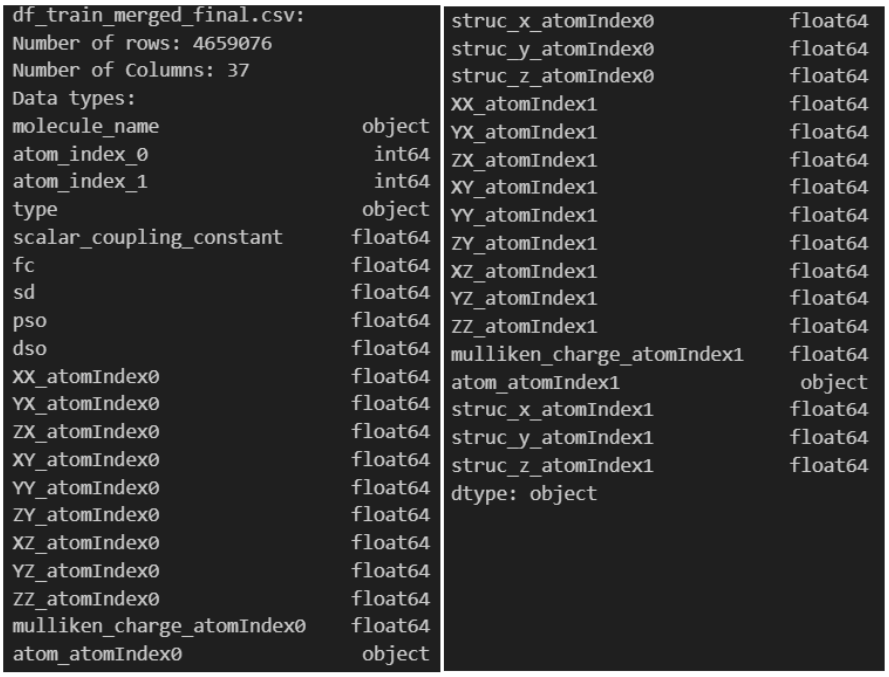
(130789, 5)

Figura 8. Energía potencial y momento dipolar por molécula.

Por otro lado los dataframe magnetic_shielding_tensors.csv, mulliken_charges.csv y structures.csv poseen información de cada uno de los átomos de las diferentes moléculas, por lo cual la data de estos puede integrarse, haciendo coincidir en cada una los nombres de las moléculas, y los indices atomicos. Quedando un nuevo data frame con cada una de las columnas de los anteriores, sin repetir evidentemente las columnas referentes a los índices químicos y a la molécula perteneciente.

También se tiene que los dataframe train y scalar_coupling_contributions poseen una organización de los datos similar, como se mencionó antes, estos poseen información de las posibles permutaciones entre átomos de una misma molécula, por ende se procede igual que en los dos casos anteriores. Resultando en un nuevo data frame con el mismo número de filas que los dos que se tenían, pero con las columnas de importancia de ambos.

Debido a que tenemos información relevante acerca de cada átomo de y nuestro dataset de entrenamiento posee 2 índices atómicos,estos son los que definen la configuración de átomos presentes en esa molécula,por ende debemos tener datos relacionados de nuestros dataframes magnetic_shielding_tensors.csv, mulliken_charges.csv y structures.csv para cada 1,por ende, se les agrega el sufijo de index_0 o index_1 para separar la información de cada átomo presente



```
df_train_merged_final.csv:
Number of rows: 4659076
Number of Columns: 37
Data types:
molecule_name                object
atom_index_0                  int64
atom_index_1                  int64
type                         object
scalar_coupling_constant    float64
fc                          float64
sd                          float64
pso                         float64
dso                         float64
XX_atomIndex0               float64
YX_atomIndex0               float64
ZX_atomIndex0               float64
XY_atomIndex0               float64
YY_atomIndex0               float64
ZY_atomIndex0               float64
XZ_atomIndex0               float64
YZ_atomIndex0               float64
ZZ_atomIndex0               float64
mulliken_charge_atomIndex0  float64
atom_atomIndex0              object
struc_x_atomIndex0          float64
struc_y_atomIndex0          float64
struc_z_atomIndex0          float64
XX_atomIndex1               float64
YX_atomIndex1               float64
ZX_atomIndex1               float64
XY_atomIndex1               float64
YY_atomIndex1               float64
ZY_atomIndex1               float64
XZ_atomIndex1               float64
YZ_atomIndex1               float64
ZZ_atomIndex1               float64
mulliken_charge_atomIndex1  float64
atom_atomIndex1              object
struc_x_atomIndex1          float64
struc_y_atomIndex1          float64
struc_z_atomIndex1          float64
dtype: object
```

Figura 9. Dataframe de entrenamiento con toda la información procesada

Se tienen más de 4 millones de datos,con 37 columnas de las cuales 4 de ellas son variables categóricas por lo cual el dataset cumple con las especificaciones solicitadas en el proyecto

```
# Select 6 random columns
cols = df_train_merged_final.sample(n=6, axis=1).columns

# Replace 5% of the data in those columns with NaN
for col in cols:
    df_train_merged_final.loc[df_train_merged_final.sample(frac=0.05).index, col] = np.nan

```

Figura 10. Código para eliminar el 5% de los datos en 6 columnas

De esta manera se cumplen todos los requisitos necesarios para continuar con el proyecto.