Emisiones globales de CO2

[**1.** **Base de datos** 2](#_Toc185179322)

[**2.** **Emisiones globales de CO2** 2](#_Toc185179323)

[**3.** **Preguntas** 3](#_Toc185179324)

[**4.** **Análisis de datos** 3](#_Toc185179325)

[**A.** **¿Qué país emite la mayor cantidad de CO2 por año?** 3](#_Toc185179326)

[**B.** **¿Como es la evolución de la emisión de CO2 por persona por año?** 6](#_Toc185179327)

[**C.** **¿Cuál es la relación entre el PBI per cápita y las emisiones de CO2?** 7](#_Toc185179328)

[**D.** **¿Qué países han logrado reducir sus emisiones de CO2 en relación con su crecimiento económico?** 8](#_Toc185179329)

[**5.** **Entrega Final** 9](#_Toc185179330)

[**A.** **Objetivo** 9](#_Toc185179331)

[**B.** **Preparación del dataset** 9](#_Toc185179332)

[**C.** **Modelo de árbol de decisión** 12](#_Toc185179333)

[**D.** **K-Nearest Neighbors (KNN)** 14](#_Toc185179334)

1. **Base de datos**

https://mavenanalytics.io/data-playground?order=date\_added%2Cdesc&tags=Weather

1. **Emisiones globales de CO2**

La base de datos contiene información detallada sobre emisiones de dióxido de carbono (CO₂), gases de efecto invernadero (GEI), consumo energético y PIB a nivel mundial, desglosada por país y año, desde 1750 hasta 2021. Esta información se divide en columnas que reflejan aspectos específicos, como las emisiones de CO₂ derivadas de fuentes como el cemento, carbón, gas, petróleo y el cambio en el uso de la tierra, así como métricas como emisiones per cápita y por unidad de energía y PIB. También se incluyen datos sobre la proporción de cada país en el total global y el crecimiento anual de las emisiones, tanto absoluto como porcentual.

Diversos hitos históricos impulsaron el aumento de las emisiones de CO₂:

1. Segunda Revolución Industrial (1870-1914): Las innovaciones en electricidad, motores de combustión interna y maquinaria alimentada por carbón y petróleo incrementaron significativamente las emisiones. La expansión del acero y el transporte motorizado intensificaron la industrialización y el uso de combustibles fósiles.
2. Expansión del Automóvil (1900-1940): La popularización del automóvil y el auge de la industria del petróleo impulsaron el consumo de combustibles fósiles. Las líneas de ensamblaje masivas, como la de Ford, facilitaron la producción y distribución de vehículos.
3. Desarrollo de la Industria Química y de Plásticos (1940-1950): Durante y después de la Segunda Guerra Mundial, la producción de plásticos y productos químicos aumentó, con una gran demanda de combustibles fósiles.
4. Boom Económico y Auge del Petróleo (1950-1973): La economía mundial, especialmente en EE.UU. y Europa, experimentó un crecimiento sin precedentes en la posguerra, elevando las emisiones debido al consumo energético y el transporte, impulsados por el petróleo.
5. Crisis del Petróleo (1973 y 1979): Estas crisis impulsaron la búsqueda de fuentes alternativas de energía, aunque algunos países adoptaron opciones más contaminantes, como el carbón.
6. Industrialización de Países en Desarrollo (1980-2000): La rápida industrialización de China, India y otros países dependió en gran medida de los combustibles fósiles, contribuyendo sustancialmente al aumento de las emisiones globales.
7. Crecimiento de China y Globalización (2000-2010): China se consolidó como una potencia manufacturera mundial, aumentando las emisiones por el uso de carbón y el crecimiento de comercio y transporte internacionales.
8. Fracturación Hidráulica y Producción de Shale Oil (2010 en adelante): Con la tecnología de fracking, países como EE.UU. aumentaron la producción de petróleo y gas. Aunque el gas natural emite menos CO₂ que el carbón, el incremento en el uso de combustibles fósiles continuó impulsando las emisiones.
9. Economía Digital e Industria Tecnológica (2010 en adelante): El auge de los centros de datos y la minería de criptomonedas ha incrementado la demanda energética, aunque algunas compañías buscan fuentes renovables.
10. Pandemia de COVID-19 (2020): Las restricciones globales derivadas de la pandemia provocaron una caída temporal en las emisiones debido a la menor actividad económica y de transporte. Sin embargo, las emisiones repuntaron rápidamente en 2021 con la reactivación económica.

Este análisis se enfoca en emisiones de CO₂ originadas por fuentes específicas como el cemento, carbón y gas, además de otros gases de efecto invernadero como el metano y el óxido nitroso. La base de datos será clave para evaluar patrones de emisiones, identificar factores impulsores y examinar la relación entre crecimiento económico y consumo de energía en diversas etapas históricas

1. **Preguntas**
2. ¿Qué país emite la mayor cantidad de CO2 por año?
3. ¿Como es la evolución de la emisión de CO2 por persona por año?
4. ¿Cuál es la relación entre el PIB per cápita y las emisiones de CO2?
5. ¿Qué países han logrado reducir sus emisiones de CO2 en relación con su crecimiento económico?
6. **Análisis de datos**

## **¿Qué país emite la mayor cantidad de CO2 por año?**

Para observar esto sumare las emisiones de CO2 de las distintas fuentes.

new\_df['emisiones\_totales'] = new\_df[['cement\_co2', 'gas\_co2', 'coal\_co2', 'consumption\_co2', 'flaring\_co2', 'land\_use\_change\_co2', 'oil\_co2', 'other\_industry\_co2','trade\_co2']].sum(axis=1) # Calculo las emisiones totales de CO₂ por país y año

df\_totales = new\_df.groupby(['country', 'year'], as\_index=False)['emisiones\_totales'].sum() # Agrupo por pais y ano para obtener el total de emisiones

df\_max\_emisiones = df\_totales.loc[df\_totales.groupby('year')['emisiones\_totales'].idxmax()] # Agrupo por ano y obtengo el país con la mayor cantidad de emisiones

print(df\_max\_emisiones.tail(20))

Durante los últimos 20 años, China y EEUU lideran la emisión de CO2 por año.

Si quisiéramos ver la emisión de CO2 de China por ejemplo en detalle:

# China = Matplotlib

df\_china = new\_df[new\_df['country'] == 'China']

plt.figure(figsize=(10, 6))

plt.plot(df\_china['year'], df\_china['emisiones\_totales'], label='Emisiones Totales', color='black', linewidth=2)

plt.plot(df\_china['year'], df\_china['cement\_co2'], label='Cemento', linestyle='--', color='blue')

plt.plot(df\_china['year'], df\_china['gas\_co2'], label='Gas', linestyle='--', color='orange')

plt.plot(df\_china['year'], df\_china['coal\_co2'], label='Carbon', linestyle='--', color='green')

plt.plot(df\_china['year'], df\_china['consumption\_co2'], label='Consumo', linestyle='--', color='pink')

plt.plot(df\_china['year'], df\_china['flaring\_co2'], label='Quema', linestyle='--', color='red')

plt.plot(df\_china['year'], df\_china['land\_use\_change\_co2'], label='Uso de la tierra', linestyle='--', color='violet')

plt.plot(df\_china['year'], df\_china['oil\_co2'], label='Petroleo', linestyle='--', color='grey')

plt.plot(df\_china['year'], df\_china['other\_industry\_co2'], label='Otras industrias', linestyle='--', color='yellow')

plt.xlabel('Año')

plt.ylabel('Emisiones de CO₂')

plt.title('Emisiones de CO₂ en China por Origen y Totales')

plt.legend()

plt.grid(True)

plt.show()

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Repito el análisis para EEUU, utilizando Seaborn:

#United States + Seaborn

df\_usa = new\_df[new\_df['country'] == 'United States']

plt.figure(figsize=(10, 6))

sns.lineplot(x='year', y='emisiones\_totales', data=df\_usa, label='Emisiones Totales', color='black', linewidth=2)

sns.lineplot(x='year', y='cement\_co2', data=df\_usa, label='Cemento', linestyle='--', color='blue')

sns.lineplot(x='year', y='gas\_co2', data=df\_usa, label='Gas', linestyle='--', color='orange')

sns.lineplot(x='year', y='coal\_co2', data=df\_usa, label='Carbon', linestyle='--', color='green')

plt.xlabel('Año')

plt.ylabel('Emisiones de CO₂')

plt.title('Emisiones de CO₂ en Estados Unidos por Origen (Cemento, Gas, Carbon) y Totales')

plt.legend()

plt.grid(True)

plt.show()

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **¿Como es la evolución de la emisión de CO2 por persona por año?**

1. df\_anio = new\_df.groupby('year', as\_index=False).agg({'population': 'sum', 'emisiones\_totales': 'sum'})
2. # Calcular las emisiones totales per cápita
3. df\_anio['emisiones\_per\_capita'] = df\_anio['emisiones\_totales'] / df\_anio['population']
4. plt.figure(figsize=(10, 6))
5. plt.plot(df\_anio['year'], df\_anio['emisiones\_per\_capita'], color='blue', marker='o', linestyle='-')
6. plt.xlabel('Año')
7. plt.ylabel('Emisiones de CO₂ per cápita')
8. plt.title('Evolución de las Emisiones Totales de CO₂ per Cápita por Año')
9. plt.grid(True)
10. plt.show()

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **¿Cuál es la relación entre el PBI per cápita y las emisiones de CO2?**

Luego del cálculo del PBI per capita, analizo si existe correlación entre ambas variables:

new\_df['PIB\_per\_capita'] = new\_df['gdp'] / new\_df['population']

correlacion = new\_df[['PIB\_per\_capita', 'emisiones\_totales']].corr()

print("Correlación entre PIB per cápita y emisiones totales de CO₂:")

print(correlacion)

El resultado es el siguiente:

PIB\_per\_capita emisiones\_totales

PIB\_per\_capita 1.000000 0.201632

emisiones\_totales 0.201632 1.000000

0.2016 indica que, en promedio, un aumento en el PIB per cápita está asociado con un ligero incremento en las emisiones de CO₂, pero esta relación es débil, lo que sugiere que otros factores también influyen significativamente en las emisiones.

Por lo tanto, el PBI no es un factor determinante, y sería útil considerar otras variables para explicar mejor las variaciones en las emisiones de CO₂ a nivel mundial.

## **¿Qué países han logrado reducir sus emisiones de CO2 en relación con su crecimiento económico?**

Considerando que China y EEUU fueron los países con mayor emisión de CO2 en los últimos años, analizare a ambos por separado.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Para el caso de EEUU, observamos que la curva de la intensidad de carbono desciende a lo largo de los años, esto indica que el país ha logrado reducir sus emisiones en relación con su PIB, mostrando una mejora en eficiencia energética. Para el caso particular de China, se detecta un comienzo en el descenso de intensidad de carbono alrededor de 1990.

1. **Entrega Final**

## **Objetivo**

Analizar si el nivel de industrialización del país se puede estimar en base a las emisiones de CO2.

## **Preparación del dataset**

Para reducir la dimensionalidad del dataset fui ejecutando los siguientes pasos:

1. Realice un filtro de países únicamente (para no utilizar continentes):

new\_df = df[df['iso\_code'].notna()]

Información del dataset:

Texto

Descripción generada automáticamente

El dataset presenta una gran cantidad de columnas que no aplican en este caso y, además, con gran cantidad de nulos.

1. Para considerar únicamente las columnas de emisiones de CO2, realice un filtro de las mismas:

# Elimino columnas con alta cantidad de nulos.

columnas\_usar = ['country',

                 'year',

                 'population',

                 'cement\_co2',

                 'gas\_co2',

                 'coal\_co2',

                 'consumption\_co2',

                 'flaring\_co2',

                 'land\_use\_change\_co2',

                 'oil\_co2',

                 'other\_industry\_co2']

df\_filtrado = new\_df[columnas\_usar]

print((df\_filtrado.isna().sum()/df\_filtrado.shape[0])\*100) # % de valores nulos por columna

Luego, analice la cantidad de nulos:

Texto

Descripción generada automáticamente

1. Reemplace valores nulos con el promedio de la columna del país correspondiente:

# Reemplazo valores nulos con el promedio de la columna del país correspondiente

for col in ['cement\_co2',

                 'gas\_co2',

                 'coal\_co2',

                 'consumption\_co2',

                 'flaring\_co2',

                 'land\_use\_change\_co2',

                 'oil\_co2',

                 'other\_industry\_co2']:

    # Calculo el promedio por país para la columna actual

    promedio\_pais = df\_filtrado.groupby('country')[col].transform('mean')

    # Reemplazo los valores nulos con el promedio correspondiente

    df\_filtrado.loc[:, col] = df\_filtrado[col].fillna(promedio\_pais)

print((df\_filtrado.isna().sum()/df\_filtrado.shape[0])\*100) # % de valores nulos por columna

#df\_filtrado.dropna(inplace=True) # No me sirve, ya que pierdo mas del 50% de los datos.

De esta manera logro obtener lo siguiente:

Texto

Descripción generada automáticamente

Realice la prueba para borrar los nulos, pero de esta forma perdía la mayoría de los datos. Con lo cual seguí modificando el dataset.

1. Para columnas con % menores al 10% de nulos los reemplace con 0 y, por otra parte, elimine del análisis las columnas consumption\_co2 y other\_industry\_co2.

# Para columnas con % menores a 10% de nulos los reemplazo por 0

df\_filtrado.loc[:, 'cement\_co2'] = df\_filtrado['cement\_co2'].fillna(0)

df\_filtrado.loc[:, 'gas\_co2'] = df\_filtrado['gas\_co2'].fillna(0)

df\_filtrado.loc[:, 'coal\_co2'] = df\_filtrado['coal\_co2'].fillna(0)

df\_filtrado.loc[:, 'flaring\_co2'] = df\_filtrado['flaring\_co2'].fillna(0)

df\_filtrado.loc[:, 'land\_use\_change\_co2'] = df\_filtrado['land\_use\_change\_co2'].fillna(0)

df\_filtrado.loc[:, 'oil\_co2'] = df\_filtrado['oil\_co2'].fillna(0)

print((df\_filtrado.isna().sum()/df\_filtrado.shape[0])\*100) # % de valores nulos por columna

# Elimino las columnas consumption\_co2 y other\_industry\_co2

columnas\_usar = ['country',

                 'year',

                 'population',

                 'cement\_co2',

                 'gas\_co2',

                 'coal\_co2',

                 'flaring\_co2',

                 'land\_use\_change\_co2',

                 'oil\_co2']

final\_df = df\_filtrado[columnas\_usar]

print((final\_df.isna().sum()/final\_df.shape[0])\*100) # % de valores nulos por columna

De esta manera, las variables a utilizar en el modelo no contienen nulos:

Texto

Descripción generada automáticamente

1. Finalmente, cree la variable **indus\_level**, intentando obtener una distribución de Baja industrialización menor al 80%:

# Necesito crear una variable categorica: Nivel de Industrialización

# Clasifico los países según el nivel de emisiones de CO₂ por sectores industriales en comparación con sus emisiones totales.

final\_df['indus\_level'] = final\_df['emisiones\_totales'].apply(

    lambda x: 'Alta industrialización' if x > 200 else 'Media industrialización' if x > 50 else 'Baja industrialización'

)

# Verificar los resultados

category\_counts = final\_df['indus\_level'].value\_counts()

print(category\_counts)

Texto

Descripción generada automáticamente

De esta manera, Baja industrialización cubre el 78% de los datos:

Texto

Descripción generada automáticamente

## **Modelo de árbol de decisión**

1. Comencé con la preparación de datos para el modelo. Para ello, definí en X a todas las variables menos la que quiero predecir, la cual se ubica en y.

X = final\_df[['cement\_co2', 'gas\_co2', 'coal\_co2', 'flaring\_co2', 'land\_use\_change\_co2', 'oil\_co2','emisiones\_totales']]

y = final\_df[['indus\_level']]

y.value\_counts(normalize=True)\*100

1. Realice la división del dataset en conjuntos de entrenamiento (70%) y prueba (30%).

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, stratify=y)

|  |  |  |
| --- | --- | --- |
|  | Filas | Columnas |
| X\_train | 26051 | 7 |
| X\_test | 11166 | 7 |
| y\_train | 26051 | 1 |
| y\_test | 11166 | 1 |

Se estratifican los datos para garantizar que las proporciones de las clases en y se mantengan en ambos conjuntos.

1. Codifiqué la variable objetivo, es decir, convertí las categorías en números:

rules = {'Alta industrialización': 0, 'Media industrialización':1, 'Baja industrialización':2}

y\_train['indus\_level'] = y\_train['indus\_level'].map(rules)

y\_test['indus\_level'] = y\_test['indus\_level'].map(rules)

1. Realizo el entrenamiento del modelo de árbol de decisión.

# ENTRENAMIENTO DEL MODELO -----------------------------------------

clf = DecisionTreeClassifier(max\_depth=10, random\_state=42).fit(X\_train,y\_train)

train\_pred = clf.predict(X\_train)

test\_pred = clf.predict(X\_test)

1. Evaluación del modelo:

Reporte de entrenamiento:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Reporte de pruebas:

Gráfico, Tabla, Calendario

Descripción generada automáticamente

1. Conclusiones

* En ambos casos, los modelos logran un accuracy y métricas perfectas. De hecho, si comparamos con el % del azar (78%) vemos que es superior, con lo cual el modelo es desempeñándose correctamente.
* La calidad de las predicciones depende en gran medida de la calidad del preprocesamiento (relleno de valores nulos, selección de variables, etc.).
* El uso de una única métrica (emisiones totales) para definir niveles de industrialización puede simplificar excesivamente la realidad. Seria bueno considerar otras variables para poder contar con una buena estimación del modelo.

## **K-Nearest Neighbors (KNN)**

1. Comencé con la normalización de los datos. MinMaxScaler transforma las características a un rango entre 0 y 1. Se ajusta (fit) al conjunto de entrenamiento y luego transforma (transform) ambos conjuntos.

scaler = MinMaxScaler() # Inicializo

# Creo conjuntos de datos

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.fit\_transform(X\_test)

1. Se seleccionan 5 vecinos más cercanos (n\_neighbors=5) para clasificar un punto según la mayoría de sus clases.

knn = KNeighborsClassifier(n\_neighbors=5)

knn.fit(X\_train\_scaled, y\_train)

1. Luego, se realizan las predicciones:

train\_pred = knn.predict(X\_train\_scaled)

test\_pred = knn.predict(X\_test\_scaled)

1. Evaluación del rendimiento:

print(classification\_report(y\_train, train\_pred))

print(classification\_report(y\_test, test\_pred))

Entrenamiento:

Una calculadora negra con letras blancas

Descripción generada automáticamente con confianza media

Prueba:

Imagen que contiene celular, teléfono, sostener, calle

Descripción generada automáticamente

1. Conclusiones:

* El modelo tiene un desempeño casi perfecto en el conjunto de entrenamiento con un 99% de accuracy. Pero, el accuracy disminuye al 70%, mostrando que el modelo en el conjunto de prueba no generaliza bien.
* El accuracy general es 70%. Aunque no es un mal valor, hay un claro desbalance en el desempeño entre las clases en ambas instancias.