GHG

June 19, 2025

[48]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
years = range(2010, 2017)
df = pd.read_excel(r"C:\Users\Sai␣
 ↪Pranavi\OneDrive\Pictures\Documents\edunnnet\SupplyChainEmissionFactorsforUSIndustriesCommo
 ↪xlsx", sheet_name=f'{years[0]}_Detail_Commodity')
df.columns = df.columns.str.strip().str.replace(" ", "_").str.
 ↪replace(r"[^\w\s]", "", regex=True)
df = df.loc[:, ~df.columns.str.contains("^Unnamed")]
num_cols = df.select_dtypes(include='number').columns
cat_cols = df.select_dtypes(include='object').columns
df[num_cols] = df[num_cols].fillna(df[num_cols].median())
df[cat_cols] = df[cat_cols].fillna("Unknown")
label_encoders = {}
for col in cat_cols:
    le = LabelEncoder()
    df[col + "_encoded"] = le.fit_transform(df[col])
    label_encoders[col] = le
scaler = StandardScaler()
scale_targets = [
    "Supply_Chain_Emission_Factors_without_Margins",
    "Margins_of_Supply_Chain_Emission_Factors",
    "Supply_Chain_Emission_Factors_with_Margins"
]
for col in scale_targets:
    if col in df.columns:
        df[col + "_scaled"] = scaler.fit_transform(df[[col]])
df.head()
```

[48]:

	Commodity_Code	Commodity_Name	\
0	1111A0	Fresh soybeans, canola, flaxseeds, and other o…	
1	1111A0	Fresh soybeans, canola, flaxseeds, and other o…	
2	1111A0	Fresh soybeans, canola, flaxseeds, and other o…	
3	1111A0	Fresh soybeans, canola, flaxseeds, and other o…	
4	1111B0	Fresh wheat, corn, rice, and other grains	

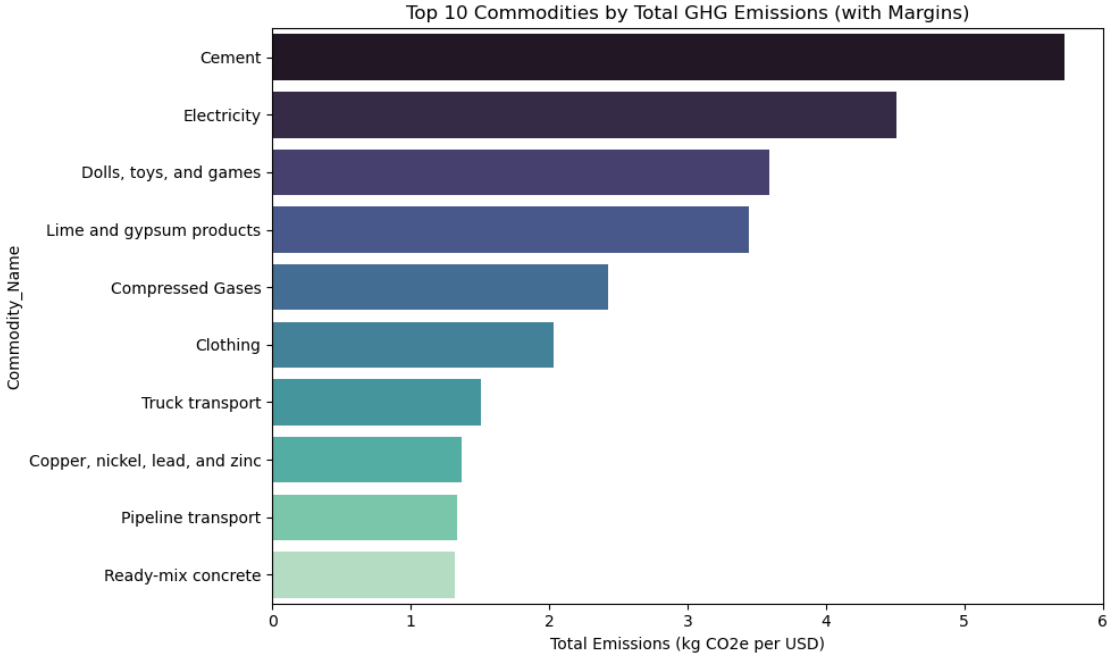
Substance	Unit	\

```
0  carbon dioxide       kg/2018 USD, purchaser price
1         methane       kg/2018 USD, purchaser price
2   nitrous oxide       kg/2018 USD, purchaser price
3      other GHGs  kg CO2e/2018 USD, purchaser price
4  carbon dioxide       kg/2018 USD, purchaser price

   Supply_Chain_Emission_Factors_without_Margins  \
0                                          0.398
1                                          0.001
2                                          0.002
3                                          0.002
4                                          0.659

   Margins_of_Supply_Chain_Emission_Factors  \
0                                     0.073
1                                     0.001
2                                     0.000
3                                     0.000
4                                     0.081

   Supply_Chain_Emission_Factors_with_Margins  \
0                                       0.470
1                                       0.002
2                                       0.002
3                                       0.002
4                                       0.740

   DQ_ReliabilityScore_of_Factors_without_Margins  \
0                                               4
1                                               4
2                                               4
3                                               3
4                                               4

   DQ_TemporalCorrelation_of_Factors_without_Margins  \
0                                                  3
1                                                  3
2                                                  3
3                                                  3
4                                                  3

   DQ_GeographicalCorrelation_of_Factors_without_Margins  \
0                                                  1
1                                                  1
2                                                  1
3                                                  1
4                                                  1
```

```
   DQ_TechnologicalCorrelation_of_Factors_without_Margins  \
0                                                  4
1                                                  1
2                                                  4
3                                                  3
4                                                  4

   DQ_DataCollection_of_Factors_without_Margins  Commodity_Code_encoded  \
0                                             1                       0
1                                             1                       0
2                                             1                       0
3                                             1                       0
4                                             1                       1

   Commodity_Name_encoded  Substance_encoded  Unit_encoded  \
0                     120                  0             1
1                     120                  1             1
2                     120                  2             1
3                     120                  3             0
4                     122                  0             1

   Supply_Chain_Emission_Factors_without_Margins_scaled  \
0                                           1.230209
1                                          -0.313531
2                                          -0.309643
3                                          -0.309643
4                                           2.245112

   Margins_of_Supply_Chain_Emission_Factors_scaled  \
0                                         0.451214
1                                        -0.158225
2                                        -0.166689
3                                        -0.166689
4                                         0.518929

   Supply_Chain_Emission_Factors_with_Margins_scaled
0                                           1.237471
1                                          -0.333490
2                                          -0.333490
3                                          -0.333490
4                                           2.143795
```

[49]:
```
import matplotlib.pyplot as plt
import seaborn as sns
if "Commodity_Name" in df.columns and␣
 ↪"Supply_Chain_Emission_Factors_with_Margins" in df.columns:
```

```
    top_emitters = (
        df.
↪groupby("Commodity_Name")["Supply_Chain_Emission_Factors_with_Margins"]
        .sum()
        .sort_values(ascending=False)
        .head(10)
    )
    plt.figure(figsize=(10, 6))
    sns.barplot(x=top_emitters.values, y=top_emitters.index, palette="mako")
    plt.title("Top 10 Commodities by Total GHG Emissions (with Margins)")
    plt.xlabel("Total Emissions (kg CO2e per USD)")
    plt.tight_layout()
    plt.show()
else:
    print("Required columns not found in the DataFrame.")
```



[ ]: