Week 1

Step 1: Import Required Libraries

In [25]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import joblib
```

Step 2: Load Dataset

In [26]:

```
excel_file = r"/af60b10b8dad38110304 (1).xlsx"  # Replace with actual path
years = range(2010, 2017)
```

In [27]:

```
years[2]
```

Out[27]:

```
2012
```

In [28]:

```
df_1 = pd.read_excel('/content/af60b10b8dad38110304 (1).xlsx', sheet_name=f'{y
df_1.head()
```

Out[28]:

	Commodity Code	Commodity Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins
0	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	carbon dioxide	kg/2018 USD, purchaser price	0.398	0.073	0.470
1	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	methane	kg/2018 USD, purchaser price	0.001	0.001	0.002
2	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	nitrous oxide	kg/2018 USD, purchaser price	0.002	0.000	0.002
3	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.002	0.000	0.002
4	1111B0	Fresh wheat, corn, rice, and other grains	carbon dioxide	kg/2018 USD, purchaser price	0.659	0.081	0.740

In [29]:

```
df_2 = pd.read_excel('/content/af60b10b8dad38110304 (1).xlsx', sheet_name=f'{y
df_2.head()
```

Out[29]:

	Industry Code	Industry Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins	Unna
0	1111A0	Oilseed farming	carbon dioxide	kg/2018 USD, purchaser price	0.414	0.073	0.487	
1	1111A0	Oilseed farming	methane	kg/2018 USD, purchaser price	0.001	0.001	0.002	
2	1111A0	Oilseed farming	nitrous oxide	kg/2018 USD, purchaser price	0.002	0.000	0.002	
3	1111A0	Oilseed farming	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.002	0.000	0.002	
4	1111B0	Grain farming	carbon dioxide	kg/2018 USD, purchaser price	0.680	0.082	0.762	

In [30]:

```
all_data = []

for year in years:
    try:
        df_com = pd.read_excel('/content/af60b10b8dad38110304 (1).xlsx', sheet
        df_ind = pd.read_excel('/content/af60b10b8dad38110304 (1).xlsx', sheet

        df_com['Source'] = 'Commodity'
        df_ind['Source'] = 'Industry'
        df_com['Year'] = df_ind['Year'] = year

        df_com.columns = df_com.columns.str.strip()
        df_ind.columns = df_ind.columns.str.strip()

        df_com.rename(columns={
            'Commodity Code': 'Code',
            'Commodity Name': 'Name'
        }, inplace=True)

        df_ind.rename(columns={
            'Industry Code': 'Code',
            'Industry Name': 'Name'
        }, inplace=True)
        all_data.append(pd.concat([df_com, df_ind], ignore_index=True))

    except Exception as e:
        print(f"Error processing year {year}: {e}")
```

In [31]:

```
all_data[3]
```

Out[31]:

	Code	Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins
0	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	carbon dioxide	kg/2018 USD, purchaser price	0.373	0.072	0.444
1	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	methane	kg/2018 USD, purchaser price	0.001	0.001	0.002
2	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	nitrous oxide	kg/2018 USD, purchaser price	0.002	0.000	0.002
3	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.002	0.000	0.002
4	1111B0	Fresh wheat, corn, rice, and other grains	carbon dioxide	kg/2018 USD, purchaser price	0.722	0.079	0.801
...	...	...	...	...	...	...	...
3151	813B00	Civic, social, professional, and similar organ...	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.008	0.000	0.008
3152	814000	Private households	carbon dioxide	kg/2018 USD, purchaser price	0.000	0.000	0.000
3153	814000	Private households	methane	kg/2018 USD, purchaser price	0.000	0.000	0.000
3154	814000	Private households	nitrous oxide	kg/2018 USD,	0.000	0.000	0.000

	Code	Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins
				purchaser price			
3155	814000	Private households	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.000	0.000	0.000

3156 rows × 15 columns

In [32]:

```
len(all_data)
```

Out[32]:

```
7
```

In [33]:

```
df = pd.concat(all_data, ignore_index=True)
df.head(10)
```

Out[33]:

	Code	Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins	Unna
0	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	carbon dioxide	kg/2018 USD, purchaser price	0.398	0.073	0.470	
1	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	methane	kg/2018 USD, purchaser price	0.001	0.001	0.002	
2	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	nitrous oxide	kg/2018 USD, purchaser price	0.002	0.000	0.002	
3	1111A0	Fresh soybeans, canola, flaxseeds, and other o...	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.002	0.000	0.002	
4	1111B0	Fresh wheat, corn, rice, and other grains	carbon dioxide	kg/2018 USD, purchaser price	0.659	0.081	0.740	
5	1111B0	Fresh wheat, corn, rice, and other grains	methane	kg/2018 USD, purchaser price	0.008	0.001	0.009	
6	1111B0	Fresh wheat, corn, rice, and other grains	nitrous oxide	kg/2018 USD, purchaser price	0.004	0.000	0.004	
7	1111B0	Fresh wheat, corn, rice, and other grains	other GHGs	kg CO2e/ 2018 USD, purchaser price	0.004	0.000	0.004	
8	111200	Fresh	carbon	kg/2018	0.183	0.132	0.315	

	Code	Name	Substance	Unit	Supply Chain Emission Factors without Margins	Margins of Supply Chain Emission Factors	Supply Chain Emission Factors with Margins	Unna
		vegetables, melons, and potatoes	dioxide	USD, purchaser price				
9	111200	Fresh vegetables, melons, and potatoes	methane	kg/2018 USD, purchaser price	0.001	0.001	0.002	

In [35]:

```
len(df)
```

Out[35]:

```
22092
```

Step 3: Data Preprocessing

In [36]:

```
df.columns # Checking columns
df.drop(columns=['Unnamed: 7'], inplace=True)
```

In [37]:

```
df.isnull().sum()
```

Out[37]:

	0
Code	0
Name	0
Substance	0
Unit	0
Supply Chain Emission Factors without Margins	0
Margins of Supply Chain Emission Factors	0
Supply Chain Emission Factors with Margins	0
DQ ReliabilityScore of Factors without Margins	0
DQ TemporalCorrelation of Factors without Margins	0
DQ GeographicalCorrelation of Factors without Margins	0
DQ TechnologicalCorrelation of Factors without Margins	0
DQ DataCollection of Factors without Margins	0
Source	0
Year	0

dtype: int64

In [38]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22092 entries, 0 to 22091
Data columns (total 14 columns):
 #   Column                                                  Non-Null Count  Dt
ype
---  ------                                                  --------------
-----
 0   Code                                                    22092 non-null  ob
ject
 1   Name                                                    22092 non-null  ob
ject
 2   Substance                                               22092 non-null  ob
ject
 3   Unit                                                    22092 non-null  ob
ject
 4   Supply Chain Emission Factors without Margins           22092 non-null  fl
oat64
 5   Margins of Supply Chain Emission Factors                22092 non-null  fl
oat64
 6   Supply Chain Emission Factors with Margins              22092 non-null  fl
oat64
 7   DQ ReliabilityScore of Factors without Margins          22092 non-null  in
t64
 8   DQ TemporalCorrelation of Factors without Margins       22092 non-null  in
t64
 9   DQ GeographicalCorrelation of Factors without Margins   22092 non-null  in
t64
 10  DQ TechnologicalCorrelation of Factors without Margins  22092 non-null  in
t64
 11  DQ DataCollection of Factors without Margins            22092 non-null  in
t64
 12  Source                                                  22092 non-null  ob
ject
 13  Year                                                    22092 non-null  in
t64
dtypes: float64(3), int64(6), object(5)
memory usage: 2.4+ MB
```