In [99]:
```
# Import Libraries

import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
import xgboost as xgb
from xgboost import XGBRegressor
```

In [100]:
```
# Import the dataset

ghg = pd.read_excel("atmosphericemissionsghg.xlsx", index_col = 0, header = 3,
                    sheet_name = 1, usecols = "C:AH", skiprows = [25])
```

In [101]:
```
#Load the data

ghg
```

Out[101]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2011	
Agriculture, forestry and fishing	55231.9	54984.4	54543.9	53737.9	54940.7	54770.8	55696.1	54825.4	54808.8	54703.0	...	46788.9	46
Mining and quarrying	48693.8	48707.1	49331.3	49142.4	43788.7	45511.6	45808.9	44970.7	43786.3	40140.2	...	23694.3	21
Manufacturing	180563.3	181614.5	174343.2	168704.3	171407.7	169367.2	170747.2	169529.7	160536.1	144288.0	...	93409.9	89
Electricity, gas, steam and air conditioning supply	216597.7	213315.8	201452.1	183993.2	179741.9	177199.8	177054.6	163758.4	169982.1	162476.6	...	163204.4	177
Water supply; sewerage, waste management	67701.1	69200.7	69402.1	69615.1	69851.2	70768.0	71303.9	70177.9	71206.3	66836.6	...	30447.3	29

Cleaning the Dataset

In [102]:
```
#Select the needed rows

ghg = ghg.iloc[:22]
```

In [103]:
```
#Confirm that the selection was done properly

ghg
```

Out[103]:

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	...	2011	2012	2013	2
Agriculture, forestry and fishing	55231.9	54984.4	54543.9	53737.9	54940.7	54770.8	55696.1	54825.4	54808.8	54703.0	...	46788.9	46521.4	46318.8	482
Mining and quarrying	48693.8	48707.1	49331.3	49142.4	43788.7	45511.6	45808.9	44970.7	43786.3	40140.2	...	23694.3	21941.5	20645.0	205
Manufacturing	180563.3	181614.5	174343.2	168704.3	171407.7	169367.2	170747.2	169529.7	160536.1	144288.0	...	93409.9	89820.1	92374.4	918
Electricity, gas, steam and air conditioning supply	216597.7	213315.8	201452.1	183993.2	179741.9	177199.8	177054.6	163758.4	169982.1	162476.6	...	163204.4	177660.1	168696.9	1475
Water supply; sewerage, waste management and remediation	67701.1	69200.7	69402.1	69615.1	69851.2	70768.0	71303.9	70177.9	71206.3	66836.6	...	30447.3	29329.2	26867.1	251

In [104]:
```
# decribe the data for more information

ghg.describe()
```

Out[104]:

	1991	1992	1993	1994	1995	1996	1997	1998	1999	200(
count	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	22.00000(
mean	76631.827273	74855.759091	73224.054545	72350.000000	71966.431818	74369.209091	72938.309091	73382.727273	70482.872727	70770.00454
std	182267.883123	177710.673396	173503.327192	171440.887734	170332.422360	175958.483413	172334.422468	173331.632007	166373.114290	167322.44529
min	45.600000	46.600000	47.700000	50.200000	51.800000	50.500000	51.000000	49.000000	50.500000	60.70000(
25%	1996.950000	1852.475000	1889.200000	1875.950000	1886.050000	1935.400000	1768.200000	1819.300000	1829.875000	1841.60000(
50%	7738.900000	7960.550000	7861.300000	8025.400000	8039.900000	8240.050000	7718.100000	7498.950000	7360.000000	6993.75000(
75%	63643.075000	64335.825000	65156.350000	66123.575000	66768.700000	67401.950000	66339.775000	67106.925000	63803.200000	61565.20000(
max	842950.100000	823413.400000	805464.700000	795850.000000	791630.700000	818061.300000	802321.400000	807210.100000	775311.600000	778470.00000(

8 rows × 30 columns

In [105]:
```
#Transpose the data set for easy usage

ghg = ghg.T
ghg
```

Out[105]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply; sewerage, waste management and remediation activities	Construction	Wholesale and retail trade; repair of motor vehicles and motorcycles	Transport and storage	Accommodation and food services	Information and communication	...	Professi scie tech activ
1990	55231.9	48693.8	180563.3	216597.7	67701.1	9441.6	11724.4	65997.8	2968.7	1103.6	...	24
1991	54984.4	48707.1	181614.5	213315.8	69200.7	9409.6	12113.6	66529.3	3401.9	1181.4	...	27
1992	54543.9	49331.3	174343.2	201452.1	69402.1	9505.4	11933.0	67599.8	3124.6	1149.6	...	2(
1993	53737.9	49142.4	168704.3	183993.2	69615.1	9622.9	12179.4	68962.5	3329.8	1199.8	...	25
1994	54940.7	43788.7	171407.7	179741.9	69851.2	10047.5	12562.8	69959.0	3317.5	1240.6	...	25
1995	54770.8	45511.6	169367.2	177199.8	70768.0	10156.6	12745.3	72797.0	3422.0	1278.6		2(

In [106]:
```
# decribe the transposed data for more information

ghg.describe()
```

Out[106]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply; sewerage, waste management and remediation activities	Construction	Wholesale and retail trade; repair of motor vehicles and motorcycles	Transport and storage	Accommodation and food services	Information and communication	...	Professio scient techn activi
count	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	...	3
unique	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	31.0	...	3
top	55231.9	48693.8	180563.3	216597.7	67701.1	9441.6	11724.4	65997.8	2968.7	1103.6	...	249
freq	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	

4 rows × 22 columns

In [107]:
```
# Search for missing values

ghg.isnull().sum().sum()
```

Out[107]: 0

In [108]:
```
#rename the overly long_named columns

ghg.rename(columns={'Activities of households as employers; undifferentiated goods and services-producing activities of householc
                  "Water supply; sewerage, waste management and remediation activities":"Water supply",
                    "Wholesale and retail trade; repair of motor vehicles and motorcycles":"Wholesale and retail trade",
                   "Public administration and defence; compulsory social security":"Public administration and defence"}
           , inplace=True)
```

Exploring the Data

In [109]:
```
#Check the dimension of the data

ghg.shape
```

Out[109]:
```
(31, 22)
```

In [110]:
```
 #Confirm the data types

ghg.dtypes
```

Out[110]:
```
Agriculture, forestry and fishing                      object
Mining and quarrying                                   object
Manufacturing                                          object
Electricity, gas, steam and air conditioning supply    object
Water supply                                           object
Construction                                           object
Wholesale and retail trade                             object
Transport and storage                                  object
Accommodation and food services                        object
Information and communication                          object
Financial and insurance activities                     object
Real estate activities                                 object
Professional, scientific and technical activities      object
Administrative and support service activities          object
Public administration and defence                      object
Education                                              object
Human health and social work activities                object
Arts, entertainment and recreation                     object
Other service activities                               object
```

In [111]:
```
#Convert the years to date time

pd.to_datetime(ghg.index,format='%Y')
```

Out[111]:
```
DatetimeIndex(['1990-01-01', '1991-01-01', '1992-01-01', '1993-01-01',
               '1994-01-01', '1995-01-01', '1996-01-01', '1997-01-01',
               '1998-01-01', '1999-01-01', '2000-01-01', '2001-01-01',
               '2002-01-01', '2003-01-01', '2004-01-01', '2005-01-01',
               '2006-01-01', '2007-01-01', '2008-01-01', '2009-01-01',
               '2010-01-01', '2011-01-01', '2012-01-01', '2013-01-01',
               '2014-01-01', '2015-01-01', '2016-01-01', '2017-01-01',
               '2018-01-01', '2019-01-01', '2020-01-01'],
              dtype='datetime64[ns]', freq=None)
```

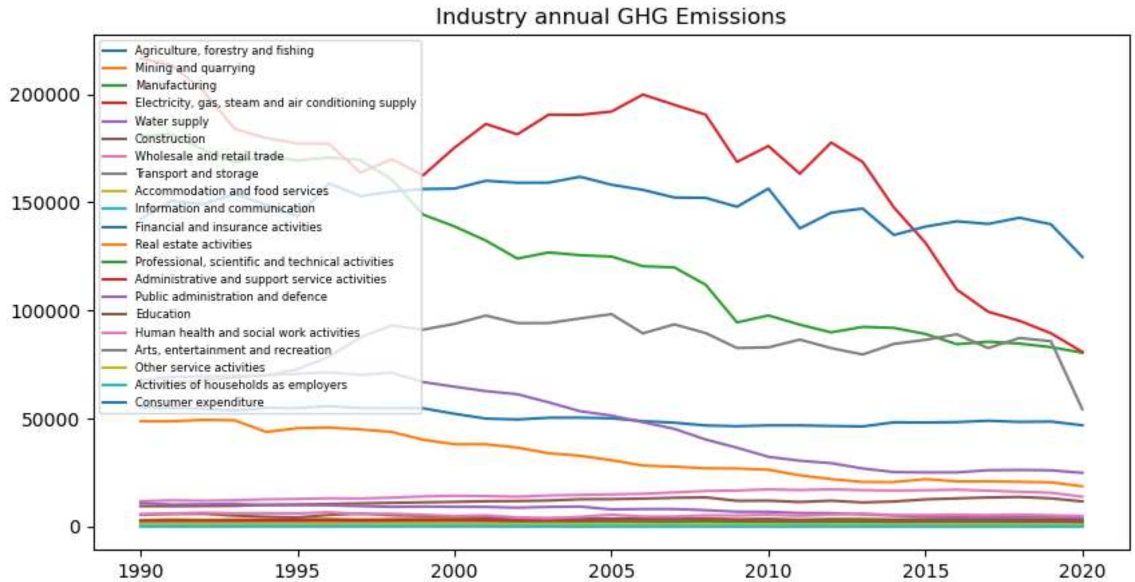
In [112]:
```
#Select only data for the industry to create a graph

industry = ghg.iloc[:, :-1]
industry
```

Out[112]:

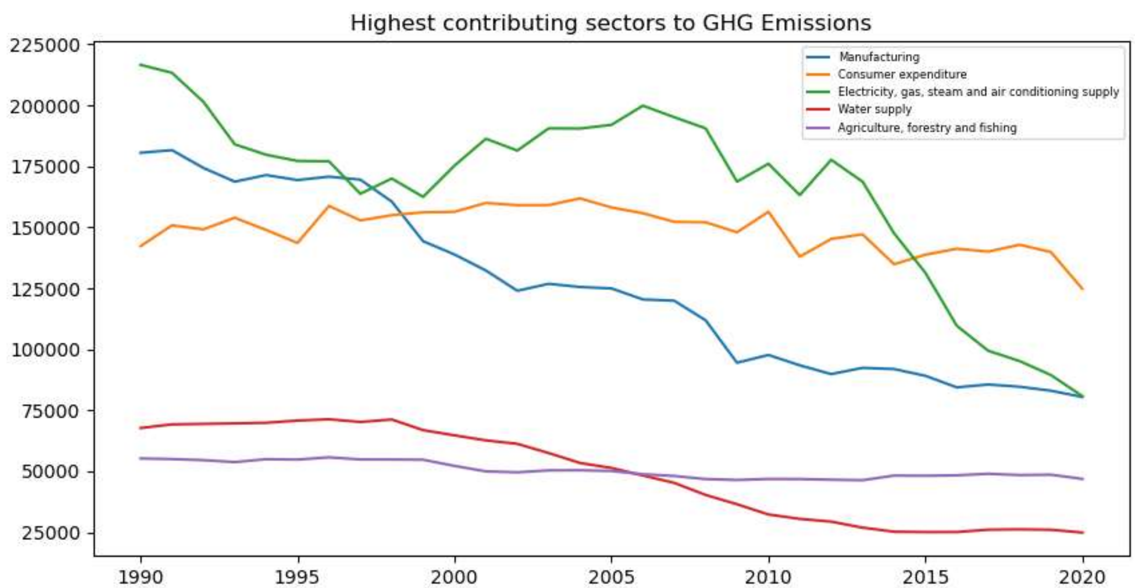
	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...	Real estate activities	Prof
1990	55231.9	48693.8	180563.3	216597.7	67701.1	9441.6	11724.4	65997.8	2968.7	1103.6	...	639.8	
1991	54984.4	48707.1	181614.5	213315.8	69200.7	9409.6	12113.6	66529.3	3401.9	1181.4	...	716.7	
1992	54543.9	49331.3	174343.2	201452.1	69402.1	9505.4	11933.0	67599.8	3124.6	1149.6	...	687.7	
1993	53737.9	49142.4	168704.3	183993.2	69615.1	9622.9	12179.4	68962.5	3329.8	1199.8	...	726.8	
1994	54940.7	43788.7	171407.7	179741.9	69851.2	10047.5	12562.8	69959.0	3317.5	1240.6	...	745.7	
1995	54770.8	45511.6	169367.2	177199.8	70768.0	10156.6	12745.3	72797.0	3422.0	1278.6	...	776.3	
1996	55696.1	45808.9	170747.2	177054.6	71303.9	10430.2	13070.4	78479.7	3607.3	1287.4	...	808.4	
1997	54825.4	44970.7	169529.7	163758.4	70177.9	10681.6	12966.8	87546.2	3220.1	1224.4	...	757.0	

Descriptive Analysis

In [113]:
```
#graph excluding total GHG emissions
industry.plot(figsize =(10,5), title = 'Industry annual GHG Emissions');
plt.legend(loc=2, prop={'size': 6})
plt.show()
```



In [114]:
```
#Plot to see the highest contributor to GHG emission for recent years

top_industry = industry[["Manufacturing",
                                      "Consumer expenditure",'Electricity, gas, steam and air conditioning supply',
                          "Water supply", "Agriculture, forestry and fishing"]].copy()

top_industry.plot(figsize =(10,5), title = 'Highest contributing sectors to GHG Emissions');
plt.legend(loc=1, prop={'size': 6})
plt.show()
```



In [115]:
```python
#Check the mean of all sectors

industry.mean(axis = 0).sort_values(ascending=False)
```

Out[115]:
```
Electricity, gas, steam and air conditioning supply    166285.235484
Consumer expenditure                                   149123.248387
Manufacturing                                          125239.225806
Transport and storage                                   83626.438710
Agriculture, forestry and fishing                       50471.325806
Water supply                                            48848.290323
Mining and quarrying                                    32786.170968
Wholesale and retail trade                              14752.964516
Construction                                            11634.522581
Public administration and defence                        7737.774194
Human health and social work activities                  5382.509677
Education                                                3902.387097
Accommodation and food services                          3445.919355
Administrative and support service activities            3073.480645
Professional, scientific and technical activities        2253.454839
Arts, entertainment and recreation                       1307.222581
Other service activities                                 1092.703226
Information and communication                            1075.222581
Real estate activities                                    860.977419
Financial and insurance activities                        308.983871
Activities of households as employers                      40.009677
dtype: float64
```

In [116]:
```python
#Check the standard deviation of all sectors

industry.std(axis = 0).sort_values(ascending=False)
```

Out[116]:
```
Electricity, gas, steam and air conditioning supply    36340.474088
Manufacturing                                          35065.150836
Water supply                                           18765.350080
Transport and storage                                  10907.974268
Mining and quarrying                                   10729.212472
Consumer expenditure                                    8831.072486
Agriculture, forestry and fishing                       3332.167983
Public administration and defence                       2182.689545
Wholesale and retail trade                              1811.612945
Construction                                            1278.783360
Education                                               1103.939929
Human health and social work activities                  641.622700
Professional, scientific and technical activities        328.578126
Arts, entertainment and recreation                       286.676005
Accommodation and food services                          224.622267
Information and communication                            166.968603
Administrative and support service activities            156.824657
Other service activities                                 130.993429
Real estate activities                                    97.076474
Financial and insurance activities                        37.111715
Activities of households as employers                      9.546321
dtype: float64
```

In [117]:
```python
#calculate the percentage change for all the sectors

pct_change = ghg.pct_change()
pct_change
```

Out[117]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...	Professional, scientific and technical activities
1990	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1991	-0.004481	0.000273	0.005822	-0.015152	0.022150	-0.003389	0.033196	0.008053	0.145922	0.070497	...	0.083150
1992	-0.008011	0.012815	-0.040037	-0.055616	0.002910	0.010181	-0.014909	0.016091	-0.081513	-0.026917	...	-0.034837
1993	-0.014777	-0.003829	-0.032344	-0.086665	0.003069	0.012361	0.020649	0.020158	0.065672	0.043667	...	-0.039080
1994	0.022383	-0.108943	0.016024	-0.023106	0.003392	0.044124	0.031479	0.014450	-0.003694	0.034006	...	0.022027
1995	-0.003092	0.039346	-0.011904	-0.014143	0.013125	0.010858	0.014527	0.040567	0.031500	0.030630	...	0.030829
1996	0.016894	0.006532	0.008148	-0.000819	0.007573	0.026938	0.025507	0.078062	0.054150	0.006883	...	0.008847
1997	-0.015633	-0.018298	-0.007130	-0.075097	-0.015792	0.024103	-0.007926	0.115527	-0.107338	-0.048936	...	-0.068283

In [118]:
```
#Get the overage average of all the sector percentage change

avg_pct_change = pct_change.mean(axis=0).sort_values(ascending=False)
avg_pct_change
```

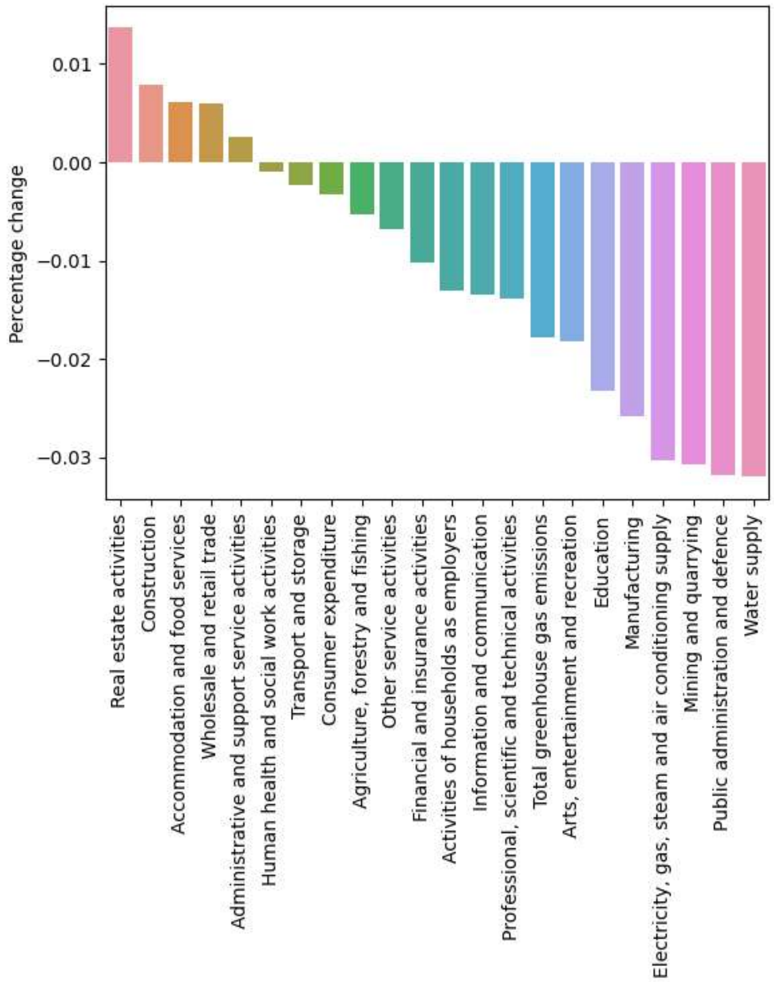
Out[118]:
```
Real estate activities                                  0.013655
Construction                                            0.007932
Accommodation and food services                         0.006132
Wholesale and retail trade                              0.005988
Administrative and support service activities           0.002522
Human health and social work activities                -0.000915
Transport and storage                                  -0.002385
Consumer expenditure                                   -0.003299
Agriculture, forestry and fishing                      -0.005310
Other service activities                               -0.006801
Financial and insurance activities                     -0.010192
Activities of households as employers                  -0.013050
Information and communication                          -0.013422
Professional, scientific and technical activities      -0.013818
Total greenhouse gas emissions                         -0.017810
Arts, entertainment and recreation                     -0.018184
Education                                              -0.023214
Manufacturing                                          -0.025791
Electricity, gas, steam and air conditioning supply    -0.030301
```

In [119]:
```
#convert to a dataframe

avg_pct_change = pd.DataFrame(avg_pct_change)
avg_pct_change.columns=["Percentage change"]
avg_pct_change
```

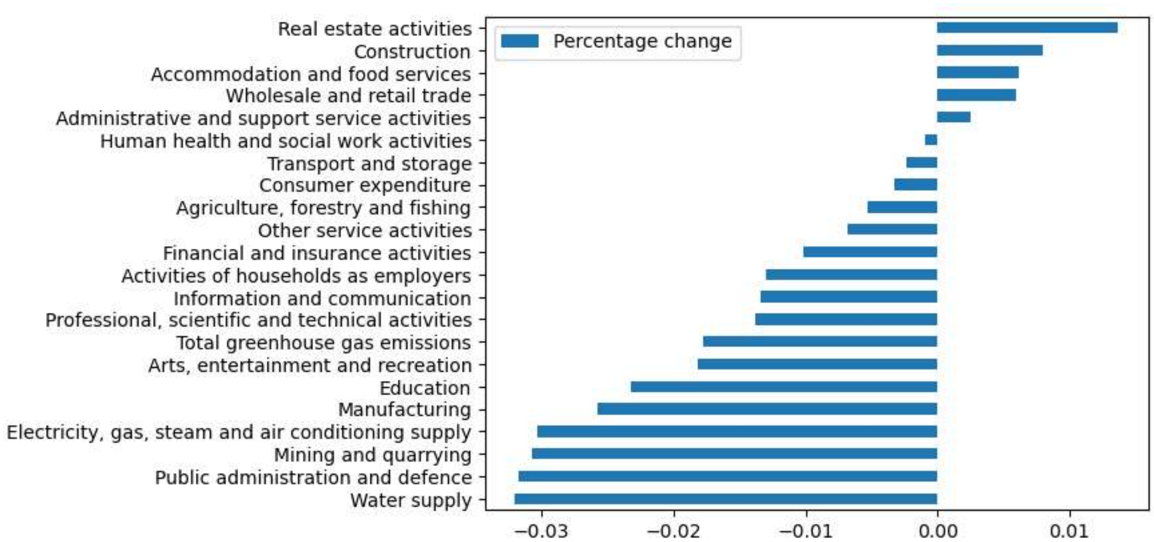
Out[119]:

	Percentage change
Real estate activities	0.013655
Construction	0.007932
Accommodation and food services	0.006132
Wholesale and retail trade	0.005988
Administrative and support service activities	0.002522
Human health and social work activities	-0.000915
Transport and storage	-0.002385
Consumer expenditure	-0.003299
Agriculture, forestry and fishing	-0.005310
Other service activities	-0.006801
Financial and insurance activities	-0.010192

In [120]:
```
#Plot the average percentage change using seaborn

sns.barplot(data = avg_pct_change, x= avg_pct_change.index, y= "Percentage change");
plt.xticks(rotation=90)
plt.show()
```



In [121]:
```
#Plot the average percentage change using matplotlib


avg_pct_change.sort_values("Percentage change").plot(kind="barh");
```

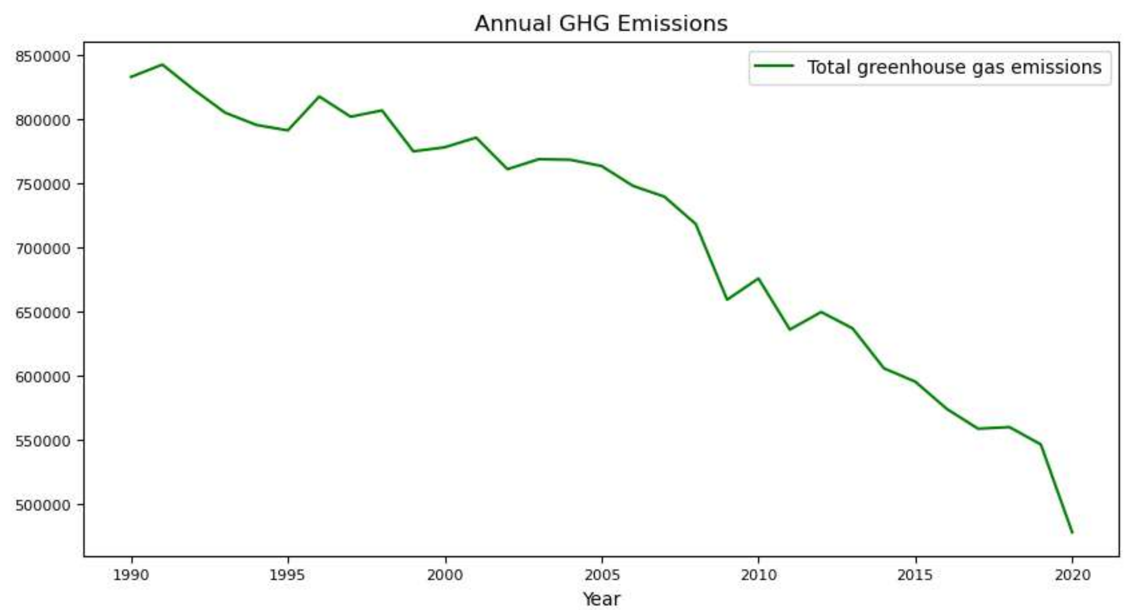


In [122]:
```
#create a dataframe with the years and total GHG emissions

df = pd.DataFrame(ghg["Total greenhouse gas emissions"], ghg.index)
df.index.name = "Year"
df
```

Out[122]:

Year	Total greenhouse gas emissions
1990	833354.6
1991	842950.1
1992	823413.4
1993	805464.7
1994	795850.0
1995	791630.7
1996	818061.3
1997	802321.4
1998	807210.1
1999	775311.6

In [123]:
```
#create a plot with the years and total GHG emissions

df.plot(figsize =(10,5),
        title = 'Annual GHG Emissions',
        fontsize = 8, color= "green");
```



In [124]:
```
#Percentage change of total emissions over the years

df.pct_change().sort_values("Total greenhouse gas emissions", ascending = True)
```

Out[124]:

Year	Total greenhouse gas emissions
2020	-0.125225
2009	-0.082107
2011	-0.058879
2014	-0.048859
1999	-0.039517
2016	-0.035505
2002	-0.031307
2008	-0.028850
2017	-0.026938
2019	-0.024122

In [125]:
```
#Total GHG change from 1990 to 2020

a = df.iloc[0]
b= df.iloc[30]
((a-b) / a) * 100.0
```

Out[125]:
```
Total greenhouse gas emissions    42.604325
dtype: object
```

In [ ]:

Test/Train Split

In [126]:
```
#Split dataset into train and test

train = ghg.loc[ghg.index <= 2018]
test = ghg.loc[ghg.index > 2018]
```

In [127]:
```
#confirm the split was properly done

train
```

Out[127]:

Year	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...	Professional, scientific and technical activities
1990	55231.9	48693.8	180563.3	216597.7	67701.1	9441.6	11724.4	65997.8	2968.7	1103.6	...	2499.1
1991	54984.4	48707.1	181614.5	213315.8	69200.7	9409.6	12113.6	66529.3	3401.9	1181.4	...	2706.9
1992	54543.9	49331.3	174343.2	201452.1	69402.1	9505.4	11933.0	67599.8	3124.6	1149.6	...	2612.6
1993	53737.9	49142.4	168704.3	183993.2	69615.1	9622.9	12179.4	68962.5	3329.8	1199.8	...	2510.5
1994	54940.7	43788.7	171407.7	179741.9	69851.2	10047.5	12562.8	69959.0	3317.5	1240.6	...	2565.8
1995	54770.8	45511.6	169367.2	177199.8	70768.0	10156.6	12745.3	72797.0	3422.0	1278.6	...	2644.9
1996	55696.1	45808.9	170747.2	177054.6	71303.9	10430.2	13070.4	78479.7	3607.3	1287.4	...	2668.3

In [128]:
```
#visiualise the test data

test
```

Out[128]:

Year	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...	Professional, scientific and technical activities	Adm a
2019	48574.6	20436.8	82996.7	89421.3	26009.3	13036.3	15642.8	85748.4	3663.3	819.2	...	1823.3	
2020	46843.2	18620.3	80438.6	80770.4	24857.2	11614.6	13800.9	54263.8	3172.5	703.7	...	1544.2	

2 rows × 22 columns

In [129]:
```
#Convert to numbers

train = train.to_numpy()
test = test.to_numpy()
```

In [130]:
```
#Setting our input and output arrays

X = train[:, :-1]
y = train[:, -1]

X_test= test[:, :-1]
y_test= test[:, -1]
```

In [131]:
```
#view

X
```

Out[131]:
```
array([[55231.9, 48693.8, 180563.3, 216597.7, 67701.1, 9441.6, 11724.4,
        65997.8, 2968.7, 1103.6, 333, 639.8, 2499.1, 2681.5, 10854.7,
        5480.6, 5743.7, 1641.3, 1120.9, 45.3, 142290.9],
       [54984.4, 48707.1, 181614.5, 213315.8, 69200.7, 9409.6, 12113.6,
        66529.3, 3401.9, 1181.4, 364.9, 716.7, 2706.9, 2824.1, 10174.6,
        5809.0, 6068.2, 1760.3, 1220.1, 45.6, 150801.4],
       [54543.9, 49331.3, 174343.2, 201452.1, 69402.1, 9505.4, 11933.0,
        67599.8, 3124.6, 1149.6, 340.3, 687.7, 2612.6, 2781.0, 10303.2,
        5921.0, 6415.7, 1599.1, 1160.7, 46.6, 149160.4],
       [53737.9, 49142.4, 168704.3, 183993.2, 69615.1, 9622.9, 12179.4,
        68962.5, 3329.8, 1199.8, 356.9, 726.8, 2510.5, 2872.7, 10512.6,
        5030.7, 6099.7, 1682.1, 1217.4, 47.7, 153920.1],
       [54940.7, 43788.7, 171407.7, 179741.9, 69851.2, 10047.5, 12562.8,
        69959.0, 3317.5, 1240.6, 353.7, 745.7, 2565.8, 2965.2, 9876.2,
        4503.7, 6174.6, 1646.0, 1218.2, 50.2, 148893.1],
       [54770.8, 45511.6, 169367.2, 177199.8, 70768.0, 10156.6, 12745.3,
        72797.0, 3422.0, 1278.6, 359.9, 776.3, 2644.9, 3022.0, 9989.2,
        4205.3, 6090.6, 1633.1, 1243.9, 51.8, 143596.9],
       [55696.1, 45808.9, 170747.2, 177054.6, 71303.9, 10430.2, 13070.4,
```

In [132]:
```
#view

y
```

Out[132]:
```
array([833354.6, 842950.1, 823413.4, 805464.7, 795850.0, 791630.7,
       818061.3, 802321.4, 807210.1, 775311.6, 778470.0, 786023.2,
       761414.9, 769198.0, 768757.0, 763847.3, 748397.6, 740014.3,
       718664.6, 659657.4, 676190.7, 636377.6, 650028.2, 637271.5,
       606135.1, 595689.2, 574539.0, 559062.3, 560295.1], dtype=object)
```

In [133]:
```
#create and train the model

model = xgb.XGBRegressor( n_estimators = 100, learning_rate=0.25)
model.fit(X,y)
```

Out[133]:
```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, feature_types=None, gamma=0, gpu_id=-1,
             grow_policy='depthwise', importance_type=None,
             interaction_constraints='', learning_rate=0.25, max_bin=256,
             max_cat_threshold=64, max_cat_to_onehot=4, max_delta_step=0,
             max_depth=6, max_leaves=0, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, ...)
```

In [134]:
```
#make predictions using the model

predictions = model.predict(X_test)
predictions[0:]
```

Out[134]:
```
array([560706.1 , 564884.44], dtype=float32)
```

In [135]:
```
#view the target result

y_test[0:]
```

Out[135]:
```
array([546779.9, 478309.5], dtype=object)
```

In [136]:
```
#accuracy of the model on the trained data

predictions_train = model.predict(X)
r2_score(y, predictions_train)
```

Out[136]:
```
0.9999999999977015
```

In [137]:
```
#mean squared error of the trained data

mean_squared_error(y, predictions_train)
```

Out[137]:
```
0.017844827583818376
```

In [138]:
```
#mean absolute error of the trained data

mean_absolute_error(y, predictions_train)
```

Out[138]: 0.09482758620288223

In [139]:
```
#accuracy of the model on the test data

print (model.score(X_test, y_test))

r2_score( y_test, predictions)
```

-2.2802212522739875

Out[139]: -2.2802212522739875

In [140]:
```
#mean squared error of the trained data

mean_squared_error( y_test, predictions)
```

Out[140]: 3844579772.9397655

In [141]:
```
#mean absolute error of the trained data

mean_absolute_error( y_test, predictions)
```

Out[141]: 50250.58124999999

In [142]:
```
#Get the next years to predict and convert into a dataframe

next_years = pd.date_range('2021', '2026', freq = 'Y')
future_years = pd.DatetimeIndex(next_years).year
future_df = pd.DataFrame(index = future_years)
future_df
```

Out[142]:

2021
2022
2023
2024
2025

In [143]:
```
#Add the next years to the original data

new_df = pd.concat([ghg,future_df])
```

In [144]:
```
#Fill every column in the new data frame with the 0.2% decrease of the previous year value

A = new_df.iloc[31]
B = new_df.iloc[30]
A.fillna(B*0.98, inplace= True)

A = new_df.iloc[32]
B = new_df.iloc[31]
A.fillna(B*0.98, inplace= True)

A = new_df.iloc[33]
B = new_df.iloc[32]
A.fillna(B*0.98, inplace= True)

A = new_df.iloc[34]
B = new_df.iloc[33]
A.fillna(B*0.98, inplace= True)

A = new_df.iloc[35]
B = new_df.iloc[34]
A.fillna(B*0.98, inplace= True)
```

In [145]:
```
#round the dataframe to 2 decimal places

new_df.astype(float).round(2)
```

Out[145]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...	Professional, scientific and technical activities
1990	55231.90	48693.80	180563.30	216597.70	67701.10	9441.60	11724.40	65997.80	2968.70	1103.60	...	2499.10
1991	54984.40	48707.10	181614.50	213315.80	69200.70	9409.60	12113.60	66529.30	3401.90	1181.40	...	2706.90
1992	54543.90	49331.30	174343.20	201452.10	69402.10	9505.40	11933.00	67599.80	3124.60	1149.60	...	2612.60
1993	53737.90	49142.40	168704.30	183993.20	69615.10	9622.90	12179.40	68962.50	3329.80	1199.80	...	2510.50
1994	54940.70	43788.70	171407.70	179741.90	69851.20	10047.50	12562.80	69959.00	3317.50	1240.60	...	2565.80
1995	54770.80	45511.60	169367.20	177199.80	70768.00	10156.60	12745.30	72797.00	3422.00	1278.60	...	2644.90
1996	55696.10	45808.90	170747.20	177054.60	71303.90	10430.20	13070.40	78479.70	3607.30	1287.40	...	2668.30
1997	54825.40	44970.70	169529.70	163758.40	70177.90	10681.60	12966.80	87546.20	3220.10	1224.40	...	2486.10

In [146]:
```
#select the new years data

test2 = new_df.loc[new_df.index >= 2021]
test2
```

Out[146]:

	Agriculture, forestry and fishing	Mining and quarrying	Manufacturing	Electricity, gas, steam and air conditioning supply	Water supply	Construction	Wholesale and retail trade	Transport and storage	Accommodation and food services	Information and communication	...
2021	45906.336	18247.894	78829.828	79154.992	24360.056	11382.308	13524.882	53178.524	3109.05	689.626	...
2022	44988.20928	17882.93612	77253.23144	77571.89216	23872.85488	11154.66184	13254.38436	52114.95352	3046.869	675.83348	...
2023	44088.445094	17525.277398	75708.166811	76020.454317	23395.397782	10931.568603	12989.296673	51072.65445	2985.93162	662.31681	...
2024	43206.676193	17174.77185	74194.003475	74500.04523	22927.489827	10712.937231	12729.510739	50051.201361	2926.212988	649.070474	...
2025	42342.542669	16831.276413	72710.123405	73010.044326	22468.94003	10498.678487	12474.920525	49050.177333	2867.688728	636.089065	...

5 rows × 22 columns

In [147]:
```
#Convert to numbers and set input and output data

test2= test2.to_numpy()
X_test2= test2[:, :-1]
y_test2= test2[:, -1]
```

In [148]:
```
#view the input data

X_test2
```

Out[148]:
```
array([[45906.335999999996, 18247.894, 78829.82800000001, 79154.992,
        24360.056, 11382.308, 13524.882, 53178.524000000005,
        3109.0499999999997, 689.626, 225.59599999999998,
        872.0039999999999, 1513.316, 2764.776, 3885.994, 2324.462,
        4784.556, 860.048, 834.7639999999999, 22.736, 122271.56199999999],
       [44988.209279999995, 17882.93612, 77253.23144, 77571.89216,
        23872.85488, 11154.66184, 13254.38436, 52114.95352,
        3046.8689999999997, 675.83348, 221.08407999999997,
        854.5639199999999, 1483.04968, 2709.4804799999997, 3808.27412,
        2277.97276, 4688.864879999999, 842.84704, 818.0687199999999,
        22.28128, 119826.13075999999],
       [44088.4450944, 17525.2773976, 75708.1668112, 76020.4543168,
        23395.3977824, 10931.568603200001, 12989.2966728, 51072.6544496,
        2985.93162, 662.3168104, 216.66239839999997, 837.4726416,
        1453.3886864, 2655.2908703999997, 3732.1086376, 2232.4133048,
        4595.0875823999995, 825.9900992, 801.7073455999998, 21.8356544,
        117429.60814479999],
       [43206.676192512, 17174.771849647997, 74194.00347497601,
        74500.04523046399, 22927.489826752, 10712.937231136,
        12729.510739344, 50051.201360608, 2926.2129876, 649.070474192,
        212.32915043199998, 820.723188768, 1424.320912672,
        2602.185052992, 3657.4664648479998, 2187.765038704,
        4503.185830752, 809.4702972160001, 785.6731986879998,
        21.398941311999998, 115081.01598190398],
       [42342.54266866176, 16831.276412655036, 72710.1234054765,
        73010.04432585472, 22468.94003021696, 10498.67848651328,
        12474.92052455712, 49050.17733339584, 2867.688727848,
        636.08906470816, 208.08256742335996, 804.30872499264,
        1395.83449441856, 2550.1413519321595, 3584.31713555104,
        2144.00973792992, 4413.12211413696, 793.28089127168,
        769.9597347142399, 20.970962485759998, 112779.3956622659]],
      dtype=object)
```

In [149]:
```
#view the output data

y_test2
```

Out[149]:
```
array([468743.31, 459368.4438, 450181.074924, 441177.45342552,
       432353.90435700957], dtype=object)
```

In [150]:
```
#run the new data into the model

GHG_forecast= model.predict(X_test2)
GHG_forecast
```

Out[150]:
```
array([564859.75, 564859.75, 564858.9 , 564858.9 , 564850.1 ],
      dtype=float32)
```

In [151]:
```
#create a data frame of the forecast

predictions_df = pd.DataFrame({'future_years': future_years, 'GHG_forecast': GHG_forecast},
                              columns=['future_years', 'GHG_forecast'])
predictions_df
```

Out[151]:

	future_years	GHG_forecast
0	2021	564859.750
1	2022	564859.750
2	2023	564858.875
3	2024	564858.875
4	2025	564850.125

In [152]:
```
#Make the years the index

predictions_df = predictions_df.set_index("future_years")
predictions_df
```

Out[152]:

	GHG_forecast
future_years	
2021	564859.750
2022	564859.750
2023	564858.875
2024	564858.875
2025	564850.125

In [153]:
```
#Plot a graph with the GHG Emissions Forecast result

predictions_df.plot( title = 'Annual GHG Emissions Forecast',
         fontsize = 6, color= "red");
```