Predicting Environmental Intensity by industry

For this notebook, we want to see whether there are different situations when applying time series model to predict environmental intensity by industry, and also combining the Environmental intensity growth rate to check whether there will be an improvement in regression performance.

First, let's import some libraries

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as msno
import warnings
import statsmodels.formula.api as smf
from sklearn import linear_model
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from math import sqrt
warnings.filterwarnings('ignore')
```

In [ ]:

```
df = pd.read_csv('Environmental_impact_cleaned.csv')
df.head(3)
```

Out[ ]:

	ISIN	Year	CompanyName	Country	Industry(Exiobase)	EnvironmentalIntensity(Sales)	EnvironmentalIntensity(OpInc)	TotalEnviror
0	DE0005545503	2016	1&1 DRILLISCH AG	Germany	Post and telecommunications (64)	-0.07%	-0.82%	
1	GB00B1YW4409	2010	3I GROUP PLC	United Kingdom	Financial intermediation, except insurance and...	-0.12%	-0.11%	
2	GB00B1YW4409	2011	3I GROUP PLC	United Kingdom	Financial intermediation, except insurance and...	-0.16%	-0.16%	

In [ ]:

```
df = df.loc[:,['CompanyName','Year','Industry(Exiobase)','Env_intensity','Environmental_Growth']]
df.head()
```

Out[ ]:

	CompanyName	Year	Industry(Exiobase)	Env_intensity	Environmental_Growth
0	1&1 DRILLISCH AG	2016	Post and telecommunications (64)	-0.0007	NaN
1	3I GROUP PLC	2010	Financial intermediation, except insurance and...	-0.0012	NaN
2	3I GROUP PLC	2011	Financial intermediation, except insurance and...	-0.0016	33.333333
3	3I GROUP PLC	2012	Financial intermediation, except insurance and...	-0.0015	-6.250000
4	3M COMPANY	2010	Activities of membership organisation n.e.c. (91)	-0.0790	NaN

Industries Regressions-- Past years Environment Intensity

First, we filter industries with more than 3 companies. We are doing this to be able to analyze and create insighful regressions.

In [ ]:

```
df_industry = df.groupby('Industry(Exiobase)').count()['CompanyName'].reset_index()
industries = df_industry[df_industry['CompanyName'] > 3]['Industry(Exiobase)']
df_industry_count4 = df[df['Industry(Exiobase)'].isin(industries)]
df_industry_count4.head()
```

Out[ ]:

	CompanyName	Year	Industry(Exiobase)	Env_intensity	Environmental_Growth
0	1&1 DRILLISCH AG	2016	Post and telecommunications (64)	-0.0007	NaN
1	3I GROUP PLC	2010	Financial intermediation, except insurance and...	-0.0012	NaN
2	3I GROUP PLC	2011	Financial intermediation, except insurance and...	-0.0016	33.333333
3	3I GROUP PLC	2012	Financial intermediation, except insurance and...	-0.0015	-6.250000
4	3M COMPANY	2010	Activities of membership organisation n.e.c. (91)	-0.0790	NaN

We created the following function to create a regression for each unique industry in the dataset. The regression features will be Environmental Intensity values from previous years.

In [ ]:

```
df_c = df_industry_count4.copy()
def calculateIndustriesRegressions(outcomeYear, pastYears, df_c):
    industry_regressions = {}
    for i in np.unique(df_c['Industry(Exiobase)']):
        for year in years:
            years.sort()
            data = df_c[(df_c['Industry(Exiobase)'] == i) & (df_c['Year'] == year)]
            data = data.loc[:,['CompanyName','Env_intensity']]
            data.rename(columns={'Env_intensity': f'Env_intensity_{year}'}, inplace=True)
            if(year == min(years)):
                data1 = pd.DataFrame(data)
            else:
                data2 = pd.merge(data1, data, on=["CompanyName"])
                data1 = data2.copy()
        if (data2.shape[0])>10:
            data3 = df_c[df_c['Year'] == outcomeYear]
            data3 = data3[['CompanyName','Env_intensity']]
            data3.rename(columns={'Env_intensity': f'Env_intensity_{outcomeYear}'}, inplace=True)
            data3 = pd.merge(data3, data2, on=["CompanyName"])

            filter_col = [col for col in data3 if ((col.startswith('Env_intensity') and not(col.endswith(f'{out
            outcome_col = [col for col in data3 if (col.startswith('Env_intensity') and col.endswith(f'{outcome
            X=data3[filter_col]
            y=data3[outcome_col]


            regr = linear_model.LinearRegression()
            X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
            regr.fit(X_train, y_train)
            y_train_pred = regr.predict(X_train)
            y_pred = regr.predict(X_test)
        else:
            continue
        industry_regressions[i] = {'OutcomeYear':outcomeYear, 'MSE_test': metrics.mean_squared_error(y_test, y_
                                   'R2_score':metrics.r2_score(y_test, y_pred)}
    data_items = industry_regressions.items()
    data_list = list(data_items)

    df = pd.DataFrame(data_list)
    df=pd.concat([df.drop(columns=df.columns[1], axis=1), df.iloc[:,1].apply(pd.Series)], axis=1)
    return df
```

Input data from 2016 to 2018 for independent variable

At first , we tried filter indutries with more than 0 company from 2016 to 2018. However, the result showed kinds of NAN for R Squared. Then, we tried filter more. Finally, we decided to filter 10 to make sure get R Squared value. We converted results into the dataframe and adjusted the dataframe format to get a clean dataset about regression results.

In [ ]:

```
years=[2016,2017,2018]
indf=calculateIndustriesRegressions(2019, years, df_c)
arr1=np.array(indf['Coefficient'].to_list())[:,0][:,0]
arr2=np.array(indf['Coefficient'].to_list())[:,0][:,1]
arr3=np.array(indf['Coefficient'].to_list())[:,0][:,2]
indf['Coefficient2016'] = arr1.tolist()
indf['Coefficient2017'] = arr2.tolist()
indf['Coefficient2018'] = arr3.tolist()
indf['Industries']=indf.iloc[:,0]
first_column = indf.pop('Industries')
indf.insert(0, 'Industries', first_column)
indf['Intercept']=np.array(indf['Intercept'].to_list())[:,0].tolist()
indf=indf.drop(indf.columns[1],axis=1)#remove repeated industries column
indf=indf.drop(columns=['Coefficient'])#remove coefficient column
indf
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2016	Coefficient2017	Coefficient2018
0	Activities auxiliary to financial intermediati...	2019	0.001666	0.040811	0.036721	-368.570788	17.976546	-36.236407	17.399672
1	Activities of membership organisation n.e.c. (91)	2019	0.000210	0.014482	0.028009	0.981829	-0.133093	0.940968	0.532472
2	Air transport (62)	2019	0.079894	0.282656	0.000122	0.243137	0.367418	-0.563799	1.192029
3	Chemicals nec	2019	0.005145	0.071727	0.005150	0.793452	0.372398	0.437839	0.173081
4	Computer and related activities (72)	2019	0.000003	0.001760	-0.000111	0.807159	0.903596	-0.338364	0.298088
5	Construction (45)	2019	0.000307	0.017532	-0.000133	0.937546	0.363022	-0.879220	1.339938
6	Extraction of crude petroleum and services rel...	2019	0.011216	0.105908	-0.018099	0.061144	0.020640	0.179937	0.608204
7	Financial intermediation, except insurance and...	2019	0.000043	0.006520	-0.000111	0.974641	-0.307046	0.922647	0.273847
8	Manufacture of beverages	2019	0.017007	0.130413	0.008430	0.930428	-0.117300	0.298025	1.101787
9	Manufacture of electrical machinery and appara...	2019	0.000402	0.020052	-0.000177	0.891901	0.320314	-0.212506	0.837339
10	Manufacture of fabricated metal products, exce...	2019	0.000020	0.004507	0.001019	0.969378	-0.207075	0.769293	0.456634
11	Manufacture of machinery and equipment n.e.c. ...	2019	0.000075	0.008643	0.002781	0.958841	0.382328	-0.201888	0.978355
12	Manufacture of medical, precision and optical ...	2019	0.001385	0.037214	-0.005836	0.694061	1.422123	-3.047546	2.260235
13	Manufacture of motor vehicles, trailers and se...	2019	0.000006	0.002397	-0.000956	0.706050	-1.127513	0.073982	2.054633
14	Manufacture of office machinery and computers ...	2019	0.000005	0.002217	0.001117	0.880571	-0.120562	-0.362403	1.666240
15	Manufacture of radio, television and communica...	2019	0.000036	0.005976	0.000662	0.899038	0.533981	-0.335609	0.779901
16	Mining of chemical and fertilizer minerals, pr...	2019	0.002808	0.052988	-0.014066	-5.955074	0.327795	-0.316628	0.847110
17	Mining of other non-ferrous metal ores and con...	2019	0.009774	0.098865	-0.071785	0.393724	0.510881	0.836161	-0.581353
18	Other land transport	2019	0.000854	0.029226	0.003607	-0.008668	-1.711894	-5.363758	8.666406
19	Other service activities (93)	2019	0.002809	0.052997	0.000708	0.919784	-0.012430	0.121612	0.705495
20	Paper	2019	0.012970	0.113885	-0.068752	0.140848	-0.151396	-0.313042	0.884858
21	Petroleum Refinery	2019	0.030471	0.174558	-0.091204	-0.170077	-1.246171	0.055407	2.007661
22	Post and telecommunications (64)	2019	0.000027	0.005179	0.000517	0.913822	0.068439	-0.433030	1.356070
23	Processing of Food products nec	2019	0.001302	0.036084	-0.000860	0.750276	-0.709269	1.276200	0.822100
24	Production of electricity nec	2019	0.047521	0.217993	-0.008272	0.804207	0.634163	-0.996679	1.199900
25	Quarrying of sand and clay	2019	0.019192	0.138534	-0.055621	0.861248	0.149951	-0.324322	1.065805
26	Real estate activities (70)	2019	0.649060	0.805643	-0.033733	-17.722403	-1.694673	-2.168827	4.263585
27	Recreational, cultural and sporting activities...	2019	0.000047	0.006886	-0.001670	0.769077	-0.043652	-0.608984	1.196476
28	Renting of machinery and equipment without ope...	2019	0.000003	0.001803	0.000032	0.914342	0.572242	-0.259025	0.580795
29	Research and development (73)	2019	0.000024	0.004931	-0.003852	0.975646	0.805671	-0.298986	0.064527
30	Retail trade, except of motor vehicles and mot...	2019	0.000006	0.002444	-0.002056	0.914233	0.007670	-0.490603	1.404277

Check with one industry's regression, 'Activities auxiliary to financial intermediation (67)', to have a look of the regression accuracy

In [ ]:

```
years = [2016, 2017, 2018,2019]
df_c = df_industry_count4.copy()
for year in years:
  data = df_c[df_c['Year'] == year]
  data = data.loc[:,['CompanyName','Env_intensity','Industry(Exiobase)']]
  data.rename(columns={'Env_intensity': f'Env_intensity_{year}'}, inplace=True)
  if(year == min(years)):
    data1 = pd.DataFrame(data)
  else:
    data2 = pd.merge(data1, data, on=["CompanyName",'Industry(Exiobase)'])
    data1 = data2.copy()

data = data2[data2['Industry(Exiobase)'] == 'Activities auxiliary to financial intermediation (67)']
X=data[['Env_intensity_2016','Env_intensity_2017','Env_intensity_2018']]
y=data['Env_intensity_2019']
regr = linear_model.LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
regr.fit(X_train, y_train)
y_train_pred = regr.predict(X_train)
y_pred = regr.predict(X_test)
print('MSE train: %.3f, test: %.3f' % (metrics.mean_squared_error(y_train, y_train_pred),
                metrics.mean_squared_error(y_test, y_pred)))
print('R2 score:', metrics.r2_score(y_test, y_pred))
print('Model intercept: ', regr.intercept_)
print('Model coefficients: ', regr.coef_)
```

```
MSE train: 0.002, test: 0.002
R2 score: -368.570788210546
Model intercept:  0.0367213032370731
Model coefficients:  [ 17.97654644 -36.23640739  17.3996719 ]
```


In [ ]:

```
years = [2016, 2017, 2018,2019]
df_c = df_industry_count4.copy()
for year in years:
  data = df_c[df_c['Year'] == year]
  data = data.loc[:,['CompanyName','Env_intensity','Industry(Exiobase)']]
  data.rename(columns={'Env_intensity': f'Env_intensity_{year}'}, inplace=True)
  if(year == min(years)):
    data1 = pd.DataFrame(data)
  else:
    data2 = pd.merge(data1, data, on=["CompanyName",'Industry(Exiobase)'])
    data1 = data2.copy()

data = data2[data2['Industry(Exiobase)'] == 'Activities auxiliary to financial intermediation (67)']
X=data[['Env_intensity_2016','Env_intensity_2017','Env_intensity_2018']]
y=data['Env_intensity_2019']
regr = linear_model.LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
regr.fit(X_train, y_train)
y_train_pred = regr.predict(X_train)
y_pred = regr.predict(X_test)
print('MSE train: %.3f, test: %.3f' % (metrics.mean_squared_error(y_train, y_train_pred),
                metrics.mean_squared_error(y_test, y_pred)))
print('R2 score:', metrics.r2_score(y_test, y_pred))
print('Model intercept: ', regr.intercept_)
print('Model coefficients: ', regr.coef_)
```

```
MSE train: 0.002, test: 0.002
R2 score: -368.570788210546
Model intercept:  0.0367213032370731
Model coefficients:  [ 17.97654644 -36.23640739  17.3996719 ]
```


The result is consistent with function result. Then, we have a look at the result table.

In [ ]:

```
indf['R2_score'].describe()
```

Out[ ]:

```
count     31.000000
mean     -12.010988
std       66.266901
min     -368.570788
25%        0.318431
50%        0.807159
75%        0.917063
max        0.981829
Name: R2_score, dtype: float64
```

From the distribution of regressions' R square, we can find that there is a huge difference in the performance for different industries

In [ ]:

```
indf=indf.set_index('Industries')
maxValuesObj = indf['R2_score'].max()
print('Maximum value in each column : ')
print(maxValuesObj)
maxValueIndexObj = indf['R2_score'].idxmax()
print("Max values of columns are at row index position :")
print(maxValueIndexObj)
minValuesObj = indf['R2_score'].min()
print('Minimum value in each column : ')
print(minValuesObj)
minValueIndexObj = indf['R2_score'].idxmin()
print("Min values of columns are at row index position :")
print(minValueIndexObj)
```

```
Maximum value in each column :
0.9818286119890922
Max values of columns are at row index position :
Activities of membership organisation n.e.c. (91)
Minimum value in each column :
-368.570788210546
Min values of columns are at row index position :
Activities auxiliary to financial intermediation (67)
```

For Activities of membership organisation n.e.c. (91), we have the highest R2_score(0.98), which indicates about 98% variance in the Environmental intensity for 2019 that the independent variables explain correctively.

If the fit is actually worse than just fitting a horizontal line then R-square is negative. Accordingly, the regression does not work for Activities auxiliary to financial intermediation (67).

In [ ]:

```
# five largest values of indutries for R square
indf5=indf.nlargest(5, ['R2_score'])
indf5
```

Out[ ]:

Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2016	Coefficient2017	Coefficient2018
Activities of membership organisation n.e.c. (91)	2019	0.000210	0.014482	0.028009	0.981829	-0.133093	0.940968	0.532472
Research and development (73)	2019	0.000024	0.004931	-0.003852	0.975646	0.805671	-0.298986	0.064527
Financial intermediation, except insurance and pension funding (65)	2019	0.000043	0.006520	-0.000111	0.974641	-0.307046	0.922647	0.273847
Manufacture of fabricated metal products, except machinery and equipment (28)	2019	0.000020	0.004507	0.001019	0.969378	-0.207075	0.769293	0.456634
Manufacture of machinery and equipment n.e.c. (29)	2019	0.000075	0.008643	0.002781	0.958841	0.382328	-0.201888	0.978355

The first five industries ordered by R square are Activities of membership organisation n.e.c. (91), Research and development (73), Financial intermediation, except insurance and pension funding (65), Manufacture of fabricated metal products, except machinery and equipment (28), and Manufacture of machinery and equipment n.e.c. (29).

Due to limited industries left when using years from 2016 to 2018 to forecast 2019 environmental intensity, we decide to have a look at years from 2017 and 2018 to forecast 2019 environmental intensity.

Input data from 2017 and 2018 for independent variable

In [ ]:

```
years=[2017,2018]
years.sort()
indf2=calculateIndustriesRegressions(2019, years, df_c)
arr1=np.array(indf2['Coefficient'].to_list())[:,0][:,0]
arr2=np.array(indf2['Coefficient'].to_list())[:,0][:,1]
indf2['Coefficient2017'] = arr1.tolist()
indf2['Coefficient2018'] = arr2.tolist()
indf2['Industries']=indf2.iloc[:,0]
first_column = indf2.pop('Industries')
indf2.insert(0, 'Industries', first_column)
indf2['Intercept']=np.array(indf2['Intercept'].to_list())[:,0].tolist()
indf2=indf2.drop(indf2.columns[1],axis=1)#remove repeated industries column
indf2=indf2.drop(columns=['Coefficient'])#remove coefficient column
indf2
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2017	Coefficient2018
0	Activities auxiliary to financial intermediati...	2019	0.000243	0.015601	0.022774	-53.299135	4.261583	-2.454473
1	Activities of membership organisation n.e.c. (91)	2019	0.000174	0.013206	0.028261	0.979978	0.865499	0.414375
2	Air transport (62)	2019	0.052530	0.229194	-0.002812	0.055293	-0.416125	1.417321
3	Chemicals nec	2019	0.000208	0.014438	0.001932	0.949105	-0.103712	1.103262
4	Computer and related activities (72)	2019	0.000876	0.029590	0.000253	-39.568634	0.503347	0.462031
5	Construction (45)	2019	0.000324	0.017998	0.000525	0.932321	-0.728185	1.704921
6	Extraction of crude petroleum and services rel...	2019	0.003437	0.058622	-0.035540	0.793735	0.123274	0.648872
7	Financial intermediation, except insurance and...	2019	0.000022	0.004709	-0.001817	0.973692	0.332456	0.500322
8	Manufacture of beverages	2019	0.005129	0.071619	0.009014	0.979018	0.068502	1.207932
9	Manufacture of electrical machinery and appara...	2019	0.000087	0.009308	0.000343	0.964291	0.458504	0.555336
10	Manufacture of fabricated metal products, exce...	2019	0.000007	0.002585	0.000973	0.991924	0.400094	0.604175
11	Manufacture of machinery and equipment n.e.c. ...	2019	0.000111	0.010532	0.001310	0.943499	0.401535	0.682682
12	Manufacture of medical, precision and optical ...	2019	0.000028	0.005252	-0.007983	0.694867	-0.287051	0.968224
13	Manufacture of motor vehicles, trailers and se...	2019	0.000003	0.001789	-0.001909	0.931291	-0.946215	1.739701
14	Manufacture of office machinery and computers ...	2019	0.001392	0.037308	0.001466	0.644262	-0.479598	1.650000
15	Manufacture of radio, television and communica...	2019	0.000007	0.002671	0.000078	0.985330	-0.182957	1.172883
16	Manufacture of rubber and plastic products (25)	2019	0.000023	0.004759	-0.000800	0.798472	-0.052314	1.060128
17	Mining of chemical and fertilizer minerals, pr...	2019	0.077498	0.278385	0.006814	-36.986142	-0.373049	1.431251
18	Mining of other non-ferrous metal ores and con...	2019	0.012135	0.110161	-0.102917	-1.012352	1.397051	-0.520040
19	Other land transport	2019	0.001215	0.034851	-0.001655	-0.434315	-6.457498	7.786464
20	Other service activities (93)	2019	0.000217	0.014732	-0.004491	0.965843	-0.060901	0.905729
21	Paper	2019	0.005293	0.072750	-0.041829	0.693188	-0.068479	0.701733
22	Petroleum Refinery	2019	0.003777	0.061460	-0.066768	0.854948	-1.151669	1.824741
23	Post and telecommunications (64)	2019	0.000033	0.005756	0.000561	0.885229	-0.323640	1.368631
24	Processing of Food products nec	2019	0.021462	0.146499	-0.000120	0.131518	0.144099	0.840529
25	Production of electricity nec	2019	0.035013	0.187118	0.002436	0.271946	-0.088990	1.022710
26	Quarrying of sand and clay	2019	0.017008	0.130415	-0.062171	0.850068	-0.448393	1.415781
27	Real estate activities (70)	2019	0.000050	0.007097	-0.001203	0.976835	-0.529953	1.350316
28	Recreational, cultural and sporting activities...	2019	0.000042	0.006492	-0.001479	0.794774	-0.644213	1.199611
29	Renting of machinery and equipment without ope...	2019	0.000006	0.002349	-0.000844	0.918038	-0.057865	0.798549
30	Research and development (73)	2019	0.000040	0.006355	-0.000165	0.915723	-0.148381	1.070900
31	Retail trade, except of motor vehicles and mot...	2019	0.000006	0.002407	-0.001495	0.858235	-0.408392	1.374242

In [ ]:

```
indf2['R2_score'].describe()
```

Out[ ]:

```
count    32.000000
mean     -3.423974
std      13.222294
min     -53.299135
25%       0.551183
50%       0.856592
75%       0.952902
max       0.991924
Name: R2_score, dtype: float64
```

In [ ]:

```
# five largest values of indutries for R square
indf2.nlargest(5, ['R2_score'])
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2017	Coefficient2018
10	Manufacture of fabricated metal products, exce...	2019	0.000007	0.002585	0.000973	0.991924	0.400094	0.604175
15	Manufacture of radio, television and communica...	2019	0.000007	0.002671	0.000078	0.985330	-0.182957	1.172883
1	Activities of membership organisation n.e.c. (91)	2019	0.000174	0.013206	0.028261	0.979978	0.865499	0.414375
8	Manufacture of beverages	2019	0.005129	0.071619	0.009014	0.979018	0.068502	1.207932
27	Real estate activities (70)	2019	0.000050	0.007097	-0.001203	0.976835	-0.529953	1.350316

In [ ]:

```
indf25=indf2.nlargest(5, ['R2_score'])
indf25['Industries'].tolist()
```

Out[ ]:

```
['Manufacture of fabricated metal products, except machinery and equipment (28)',
 'Manufacture of radio, television and communication equipment and apparatus (32)',
 'Activities of membership organisation n.e.c. (91)',
 'Manufacture of beverages',
 'Real estate activities (70)']
```

The situation is a little bit different with the regression for 2016 to 2018 to researsh 2019 env intensity. However, regressions for Activities of membership organisation n.e.c. (91), and Manufacture of fabricated metal products, except machinery and equipment (28) both perform good.

Now, let's check whether model with 2016 data will be better for top 5 industries ordered by R square.

In [ ]:

```
print(indf5['R2_score'].mean())
print(indf25['R2_score'].mean())
```

```
0.9720669285619052
0.9826170978729676
```

In [ ]:

```
print(indf5['MSE_test'].mean())
print(indf25['MSE_test'].mean())
```

```
7.431229530659072e-05
0.0010735819826167424
```

With the higher R2 score and lower MSE for test data, regressions with 2017 and 2018 data is better than with data 2016 to 2018 for the first five industries.

Industry regressions--Past years Evironmental Intensity with growth rate

Now, we did the regression for environmental intensity with growth rate to check whether the regression performs better

In [ ]:

```
df_c = df_industry_count4.copy()
def calculateIndustriesGrowthRegressions(outcomeYear, pastYears, df_c):
    industry_regressions = {}
    for i in np.unique(df_c['Industry(Exiobase)']):
        for year in years:
            years.sort()
            data = df_c[(df_c['Industry(Exiobase)'] == i) & (df_c['Year'] == year)]
            data = data.loc[:,['CompanyName','Env_intensity','Environmental_Growth']]
            data.rename(columns={'Env_intensity': f'Env_intensity_{year}','Environmental_Growth': f'Environment
            if(year == min(years)):
                data1 = pd.DataFrame(data)
            else:
                data2 = pd.merge(data1, data, on=["CompanyName"])
                data1 = data2.copy()
        data2.dropna(inplace=True)
        data2 = data2.fillna("", inplace=False)
        if (data2.shape[0])> 10:
            data3 = df_c[df_c['Year'] == outcomeYear]
            data3 = data3[['CompanyName','Env_intensity']]
            data3.rename(columns={'Env_intensity': f'Env_intensity_{outcomeYear}'}, inplace=True)
            data3 = pd.merge(data3, data2, on=["CompanyName"])

            filter_col = [col for col in data3 if ((col.startswith('Env_intensity') and not(col.endswith(f'{out
            outcome_col = [col for col in data3 if (col.startswith('Env_intensity') and col.endswith(f'{outcome
            X=data3[filter_col]
            y=data3[outcome_col]

            regr = linear_model.LinearRegression()
            X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
            regr.fit(X_train, y_train)
            y_train_pred = regr.predict(X_train)
            y_pred = regr.predict(X_test)
        else:
            continue
        industry_regressions[i] = {'OutcomeYear':outcomeYear, 'MSE_test': metrics.mean_squared_error(y_test, y_
                                   'R2_score':metrics.r2_score(y_test, y_pred)}

    data_items = industry_regressions.items()
    data_list = list(data_items)

    df = pd.DataFrame(data_list)

    df=pd.concat([df.drop(columns=df.columns[1], axis=1), df.iloc[:,1].apply(pd.Series)], axis=1)

    return df
```

Input data from 2016 to 2018

In [ ]:

```
years = [2016, 2017, 2018]
gindf=calculateIndustriesGrowthRegressions(2019, years, df_c)
arr1=np.array(gindf['Coefficient'].to_list())[:,0][:,0]
arr2=np.array(gindf['Coefficient'].to_list())[:,0][:,1]
arr3=np.array(gindf['Coefficient'].to_list())[:,0][:,2]
arr4=np.array(gindf['Coefficient'].to_list())[:,0][:,3]
arr5=np.array(gindf['Coefficient'].to_list())[:,0][:,4]
arr6=np.array(gindf['Coefficient'].to_list())[:,0][:,5]
gindf['Coefficient2016'] = arr1.tolist()
gindf['Coefficient2017'] = arr2.tolist()
gindf['Coefficient2018'] = arr3.tolist()
gindf['Coefficientgrowth2016'] = arr4.tolist()
gindf['Coefficientgrowth2017'] = arr5.tolist()
gindf['Coefficientgrowth2018'] = arr6.tolist()
gindf['Industries']=gindf.iloc[:,0]
first_column = gindf.pop('Industries')
gindf.insert(0, 'Industries', first_column)
gindf['Intercept']=np.array(gindf['Intercept'].to_list())[:,0].tolist()
gindf=gindf.drop(gindf.columns[1],axis=1)#remove repeated industries column
gindf=gindf.drop(columns=['Coefficient'])#remove coefficient column
gindf
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2016	Coefficient2017	Coefficient2018	Coeffici
0	Activities auxiliary to financial intermediati...	2019	0.008146	0.090253	0.184351	-1806.457281	-76.671009	5.231518e-03	299.823319	
1	Activities of membership organisation n.e.c. (91)	2019	0.000182	0.013481	-0.014931	0.984255	-1.441925	-5.700398e-04	2.637142	
2	Air transport (62)	2019	0.156553	0.395668	0.004236	-0.483075	3.718195	-8.604827e-03	-4.327460	
3	Chemicals nec	2019	0.000520	0.022810	-0.002510	0.743259	-0.791492	-7.529472e-05	0.360464	
4	Computer and related activities (72)	2019	0.000004	0.001926	-0.000160	0.715955	0.292750	-5.419177e-06	0.824918	
5	Construction (45)	2019	0.000033	0.005708	0.002028	0.967457	-0.236089	-7.235401e-06	-0.335368	
6	Extraction of crude petroleum and services rel...	2019	0.003995	0.063207	-0.088699	0.481081	0.718391	-5.110948e-04	-0.124293	
7	Financial intermediation, except insurance and...	2019	0.000020	0.004494	0.000522	0.596149	-0.480050	2.765622e-05	0.499465	
8	Manufacture of beverages	2019	0.010042	0.100209	0.000235	0.958872	0.092050	2.111523e-04	-0.651669	
9	Manufacture of electrical machinery and appara...	2019	0.000452	0.021261	-0.004139	0.878469	0.529052	-9.172356e-05	-0.204613	
10	Manufacture of fabricated metal products, exce...	2019	0.000026	0.005104	-0.000575	0.961553	-0.256612	9.858632e-05	1.883722	
11	Manufacture of machinery and equipment n.e.c. ...	2019	0.000014	0.003678	0.004047	0.859069	1.045477	1.651940e-04	-0.519001	
12	Manufacture of medical, precision and optical ...	2019	0.002495	0.049953	0.001493	-3.668078	3.925910	1.311627e-04	-4.502719	
13	Manufacture of motor vehicles, trailers and se...	2019	0.000001	0.001085	-0.001064	0.912392	-0.568742	-4.166060e-07	-1.141022	
14	Manufacture of office machinery and computers ...	2019	0.000004	0.001924	0.001824	0.448600	-0.166928	-2.362329e-06	-0.706134	
15	Manufacture of radio, television and communica...	2019	0.000063	0.007925	0.000543	0.822465	1.038756	-8.463013e-06	-0.773132	
16	Mining of chemical and fertilizer minerals, pr...	2019	0.003976	0.063057	-0.009523	-8.849329	0.297611	-8.553659e-05	-0.379296	
17	Mining of other non-ferrous metal ores and con...	2019	0.111642	0.334129	0.184178	-3.089403	-0.374675	-2.134968e-03	2.821386	
18	Other land transport	2019	0.000976	0.031246	0.004951	-0.152957	-0.545229	-2.947094e-04	-7.941331	
19	Other service activities (93)	2019	0.000739	0.027190	-0.010765	0.825433	0.646674	2.286027e-04	-0.437858	
20	Paper	2019	0.107000	0.327109	-0.736860	-6.087945	26.176830	-3.465895e-03	-39.621052	
21	Petroleum Refinery	2019	0.011593	0.107669	-0.048806	0.554841	-1.761917	4.978635e-04	0.293264	
22	Post and telecommunications (64)	2019	0.000038	0.006146	0.000692	0.929535	0.311531	3.561866e-06	-0.558642	
23	Processing of Food products nec	2019	0.001177	0.034303	0.004659	0.838327	-0.808199	6.975249e-04	1.061048	
24	Production of electricity nec	2019	0.029268	0.171079	0.041055	0.780955	0.818320	2.144318e-03	-1.415703	
25	Quarrying of sand and clay	2019	0.031877	0.178543	-0.000777	0.577469	1.135572	3.575385e-04	-1.830838	
26	Real estate activities (70)	2019	0.317682	0.563633	0.001055	-0.639591	1.192393	3.977738e-04	-0.567318	
27	Recreational, cultural and sporting activities...	2019	0.000003	0.001857	0.000945	0.985070	0.239191	5.962579e-05	-0.377784	
28	Renting of machinery and equipment without ope...	2019	0.000005	0.002284	0.003275	0.790324	-0.765386	-2.932786e-05	-0.439725	
29	Retail trade, except of motor vehicles and mot...	2019	0.000060	0.007765	-0.002972	0.608754	0.683922	-7.005030e-05	-2.663307	

check with one industry whether the function is accurate

In [ ]:

```
years = [2016, 2017, 2018,2019]
df_c = df_industry_count4.copy()
for year in years:
  data = df_c[df_c['Year'] == year]
  data = data.loc[:,['CompanyName','Env_intensity','Industry(Exiobase)','Environmental_Growth']]
  data.rename(columns={'Env_intensity': f'Env_intensity_{year}','Environmental_Growth': f'Environmental_Growth_
  if(year == min(years)):
    data1 = pd.DataFrame(data)
  else:
    data2 = pd.merge(data1, data, on=["CompanyName",'Industry(Exiobase)'])
    data1 = data2.copy()
data = data2[data2['Industry(Exiobase)'] == 'Air transport (62)']
X=data.drop(columns=['CompanyName','Env_intensity_2019','Industry(Exiobase)','Environmental_Growth_2019'])
X=data[['Env_intensity_2016','Environmental_Growth_2016','Env_intensity_2017','Environmental_Growth_2017','Env_
y=data['Env_intensity_2019']
regr = linear_model.LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
regr.fit(X_train, y_train)
y_train_pred = regr.predict(X_train)
y_pred = regr.predict(X_test)
print('MSE train: %.3f, test: %.3f' % (metrics.mean_squared_error(y_train, y_train_pred),
              metrics.mean_squared_error(y_test, y_pred)))
print('R2 score:', metrics.r2_score(y_test, y_pred))
print('Model intercept: ', regr.intercept_)
print('Model coefficients: ', regr.coef_)
```

```
MSE train: 0.000, test: 0.157
R2 score: -0.4830752229570636
Model intercept:  0.004235999337293073
Model coefficients:  [ 3.71819540e+00 -8.60482710e-03 -4.32745978e+00 -7.32501446e-03
  1.54642276e+00 -3.88414716e-05]
```

The result is consistent with the table

In [ ]:

```
gindf['R2_score'].describe()
```

Out[ ]:

```
count      30.000000
mean      -60.406912
std       329.784213
min     -1806.457281
25%        -0.002568
50%         0.729607
75%         0.873619
max         0.985070
Name: R2_score, dtype: float64
```

In [ ]:

```
# five largest values of indutries for R square
gindf5=gindf.nlargest(5, ['R2_score'])
gindf5
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2016	Coefficient2017	Coefficient2018	Coefficientgrowth2
27	Recreational, cultural and sporting activities...	2019	0.000003	0.001857	0.000945	0.985070	0.239191	0.000060	-0.377784	0.00
1	Activities of membership organisation n.e.c. (91)	2019	0.000182	0.013481	-0.014931	0.984255	-1.441925	-0.000570	2.637142	0.00
5	Construction (45)	2019	0.000033	0.005708	0.002028	0.967457	-0.236089	-0.000007	-0.335368	0.00
10	Manufacture of fabricated metal products, exce...	2019	0.000026	0.005104	-0.000575	0.961553	-0.256612	0.000099	1.883722	0.00
8	Manufacture of beverages	2019	0.010042	0.100209	0.000235	0.958872	0.092050	0.000211	-0.651669	-0.00

In [ ]:

```
gindf5['Industries'].tolist()
```

Out[ ]:

```
['Recreational, cultural and sporting activities (92)',
 'Activities of membership organisation n.e.c. (91)',
 'Construction (45)',
 'Manufacture of fabricated metal products, except machinery and equipment (28)',
 'Manufacture of beverages']
```

In [ ]:

```
print(gindf5['R2_score'].mean())
print(indf5['R2_score'].mean())
```

```
0.9714415807968415
0.9720669285619052
```

In [ ]:

```
print(gindf5['MSE_test'].mean())
print(indf5['MSE_test'].mean())
```

```
0.002057148325698516
7.431229530659072e-05
```

When comparing with the first 5 industries ordered by R2 score, it seems the regression with growth rate performs better because of the similar R squared but lower MSE to test data.

Then we are going to compare with the regression by using data from 2017 and 2018 to forecast

Input data from 2017 and 2018

In [ ]:

```
years = [2017, 2018]
gindf2=calculateIndustriesGrowthRegressions(2019, years, df_c)
arr1=np.array(gindf2['Coefficient'].to_list())[:,0][:,0]
arr2=np.array(gindf2['Coefficient'].to_list())[:,0][:,1]
arr3=np.array(gindf2['Coefficient'].to_list())[:,0][:,2]
arr4=np.array(gindf2['Coefficient'].to_list())[:,0][:,3]
gindf2['Coefficient2017'] = arr1.tolist()
gindf2['Coefficient2018'] = arr2.tolist()
gindf2['Coefficientgrowth2017'] = arr3.tolist()
gindf2['Coefficientgrowth2018'] = arr4.tolist()
gindf2['Industries']=gindf2.iloc[:,0]
first_column = gindf2.pop('Industries')
gindf2.insert(0, 'Industries', first_column)
gindf2['Intercept']=np.array(gindf2['Intercept'].to_list())[:,0].tolist()
gindf2=gindf2.drop(gindf2.columns[1],axis=1)#remove repeated industries column
gindf2=gindf2.drop(columns=['Coefficient'])#remove coefficient column
gindf2
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2017	Coefficient2018	Coefficientgrowth2017
0	Activities auxiliary to financial intermediati...	2019	3.648323e-03	0.060401	0.073740	-808.539089	-1.821584	0.004127	2.688033
1	Activities of membership organisation n.e.c. (91)	2019	2.300809e-02	0.151684	0.003164	-1.641548	1.866384	0.000079	-0.724753
2	Air transport (62)	2019	5.262270e-02	0.229396	-0.010415	-0.217167	0.133816	-0.000368	0.841349
3	Chemicals nec	2019	3.869982e-04	0.019672	0.007529	0.830493	0.011047	0.000518	0.987010
4	Computer and related activities (72)	2019	1.924340e-06	0.001387	-0.002482	0.903975	2.894210	-0.000011	-2.455119
5	Construction (45)	2019	3.310563e-04	0.018195	0.000552	0.932729	-1.044670	0.000058	1.895110
6	Extraction of crude petroleum and services rel...	2019	7.255946e-03	0.085182	-0.085129	0.749181	0.346909	-0.000096	0.185284
7	Financial intermediation, except insurance and...	2019	1.593699e-04	0.012624	0.000447	0.984304	0.269207	-0.000017	0.623544
8	Manufacture of beverages	2019	4.419836e-04	0.021023	0.007999	0.998192	0.015528	0.000010	1.241038
9	Manufacture of electrical machinery and appara...	2019	1.204170e-04	0.010973	0.002729	0.965995	0.461843	0.000133	0.580764
10	Manufacture of fabricated metal products, exce...	2019	2.203915e-05	0.004695	0.000658	0.966771	0.556284	-0.000074	0.438741
11	Manufacture of machinery and equipment n.e.c. ...	2019	1.706126e-05	0.004131	0.003181	0.990599	0.594594	0.000056	0.561005
12	Manufacture of medical, precision and optical ...	2019	5.279608e-05	0.007266	-0.007888	0.264986	-1.199355	-0.000196	1.898186
13	Manufacture of motor vehicles, trailers and se...	2019	8.370036e-07	0.000915	-0.002694	0.957182	-2.353522	-0.000108	3.169215
14	Manufacture of office machinery and computers ...	2019	3.660341e-06	0.001913	0.001473	0.911046	-0.940983	-0.000006	2.093857
15	Manufacture of radio, television and communica...	2019	3.195515e-06	0.001788	-0.000127	0.989351	-0.004038	-0.000022	0.981032
16	Mining of chemical and fertilizer minerals, pr...	2019	6.512627e-03	0.080701	0.032808	-15.132243	-0.838847	0.002610	1.860512
17	Mining of other non-ferrous metal ores and con...	2019	2.017218e-01	0.449135	-0.330876	-0.026753	-0.314094	-0.002541	0.388705
18	Other land transport	2019	1.119893e-03	0.033465	-0.006207	-0.322496	-6.276295	-0.000614	7.599414
19	Other service activities (93)	2019	1.165047e-04	0.010794	-0.006368	0.986606	-0.089004	0.000200	0.956525
20	Paper	2019	7.777551e-03	0.088190	-0.065796	0.494775	1.240309	-0.001153	-0.694588
21	Petroleum Refinery	2019	1.340017e-02	0.115759	-0.106039	0.485430	-1.832445	-0.002216	2.532759
22	Post and telecommunications (64)	2019	3.984904e-05	0.006313	0.001478	0.878970	-0.392810	0.000014	1.435704
23	Processing of Food products nec	2019	8.601396e-04	0.029328	-0.010281	0.832846	0.495920	-0.000868	0.690928
24	Production of electricity nec	2019	5.972529e-03	0.077282	-0.038998	0.774246	-0.293884	0.002314	1.166884
25	Quarrying of sand and clay	2019	7.839106e-03	0.088539	-0.035489	0.937858	-0.626834	0.001765	1.602403
26	Real estate activities (70)	2019	3.312385e-01	0.575533	0.007570	-0.972079	0.825365	0.000376	0.277031
27	Recreational, cultural and sporting activities...	2019	6.056742e-05	0.007783	-0.003449	0.705038	-0.740306	-0.000036	1.194581
28	Renting of machinery and equipment without ope...	2019	3.115047e-07	0.000558	-0.000465	0.991790	-1.200695	-0.000010	2.116699
29	Research and development (73)	2019	1.680006e-04	0.012962	-0.001354	0.831757	-0.515124	0.000035	1.449575
30	Retail trade, except of motor vehicles and mot...	2019	5.879736e-06	0.002425	-0.001608	0.582635	-0.440564	0.000014	1.373470

In [ ]:

```
gindf2['R2_score'].describe()
```

Out[ ]:

```
count     31.000000
mean     -26.029181
std      145.256397
min     -808.539089
25%        0.375208
50%        0.831757
75%        0.961589
max        0.998192
Name: R2_score, dtype: float64
```

In [ ]:

```
# five largest values of indutries for R square
gindf25=gindf2.nlargest(5, ['R2_score'])
gindf25
```

Out[ ]:

	Industries	OutcomeYear	MSE_test	RMSE_test	Intercept	R2_score	Coefficient2017	Coefficient2018	Coefficientgrowth2017	Coefficien
8	Manufacture of beverages	2019	4.419836e-04	0.021023	0.007999	0.998192	0.015528	0.000010	1.241038	
28	Renting of machinery and equipment without ope...	2019	3.115047e-07	0.000558	-0.000465	0.991790	-1.200695	-0.000010	2.116699	
11	Manufacture of machinery and equipment n.e.c. ...	2019	1.706126e-05	0.004131	0.003181	0.990599	0.594594	0.000056	0.561005	
15	Manufacture of radio, television and communica...	2019	3.195515e-06	0.001788	-0.000127	0.989351	-0.004038	-0.000022	0.981032	
19	Other service activities (93)	2019	1.165047e-04	0.010794	-0.006368	0.986606	-0.089004	0.000200	0.956525	

In [ ]:

```
gindf25['Industries'].tolist()
```

Out[ ]:

```
['Manufacture of beverages',
 'Renting of machinery and equipment without operator and of personal and household goods (71)',
 'Manufacture of machinery and equipment n.e.c. (29)',
 'Manufacture of radio, television and communication equipment and apparatus (32)',
 'Other service activities (93)']
```

In [ ]:

```
print(gindf25['R2_score'].mean())
print(gindf5['R2_score'].mean())
```

```
0.9913073627104925
0.9714415807968415
```

In [ ]:

```
print(gindf25['MSE_test'].mean())
print(gindf5['MSE_test'].mean())
```

```
0.00011581132909609158
0.002057148325698516
```

Accordingly, for the first five industries, the regression with data in 2017 and 2018 performs better than the regression with data from 2016 to 2018.

In [ ]:

```
print(gindf25['R2_score'].mean())
print(indf25['R2_score'].mean())
```

```
0.9913073627104925
0.9826170978729676
```

In [ ]:

```
print(gindf25['MSE_test'].mean())
print(indf25['MSE_test'].mean())
```

```
0.00011581132909609158
0.0010735819826167424
```

For first five indutries ordered by R2 score, it seems be the same situation that regressions perfrom better when adding growth rates with the higher R2 score and MSE to test data.

Conclusion

- Among the first five industries ordered by R2 scores, the average result from regressions for industries with data from 2017 and 2018 perform better than regressions with data from 2016 to 2018 to forecast 2019.
- When combining growth rates into independent variables, the performance of regressions will be better.
- There is a huge different performance of regressions for different industries. The time series model for the industry "Manufacture of fabricated metal products, except machinery and equipment (28)" performs better than other industries that all regressions have a high R square.

We will continue our analysis in 'PredictingTimeSeries - GHG Scope' notebook