Predicting The Current Year's GHG with the Previous Years' GHG Scope

In this notebook we are going to try and predict the GHG Scope of 2019 with values from the previous years. We are going to be using both the actual values and the percentage change year-over-year.

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as msno
from sklearn.linear_model import LinearRegression, Ridge, RidgeCV, Lasso, LassoCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split, cross_val_score, KFold, cross_val_predict
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.preprocessing import scale, PolynomialFeatures
from sklearn.feature_selection import RFE
from datetime import datetime, date
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller


stocks = pd.read_csv("/Users/YEET/Documents/GitHub/Predicting-Environmental-and-Social-Actions/Datasets/company
sectors = pd.read_csv("/Users/YEET/Documents/GitHub/Predicting-Environmental-and-Social-Actions/Datasets/52_tic

stocks['Missing_GHG'] = np.where(stocks['GHG Scope 1'].isna(), 1, 0)
stocks['GHG Scope 1'].fillna(0, inplace = True)
stocks.loc[stocks['GHG Scope 1'].isna(),['GHG Scope 1','Missing_GHG']].head()

stocks = stocks.merge(sectors, on='Ticker')
stocks['GHG Scope 1'] = stocks['GHG Scope 1'].astype(float)
stocks['Percent_Change_GHG'] = (stocks.groupby('Ticker')['GHG Scope 1'].apply(pd.Series.pct_change) + 1)
```

```
C:\Users\YEET\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.util.testing
is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

Using Average of 2016, 2017, and 2018 GHG Scope to Predict 2019

In [15]:

```
companies_2018 = list(stocks[(stocks['Year'] == 2018) & (stocks['GHG Scope 1'] != 0)]['Ticker'])
companies_2019 = list(stocks[(stocks['Year'] == 2019) & (stocks['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016,2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(stocks[(stocks['Year'] == 2017) & (stocks['GHG Scope 1'] != 0) ]['Ticker'])
list2017_as_set = set(companies_2017)
intersection2= list2017_as_set.intersection(intersection)

companies_2016 = list(stocks[(stocks['Year'] == 2016) & (stocks['GHG Scope 1'] != 0) ]['Ticker'])
list2016_as_set = set(companies_2016)
intersection2= list2016_as_set.intersection(intersection2)

x = stocks[(stocks['Year'].isin([2016, 2017,2018])) & (stocks['Ticker'].isin(intersection2))][['Year', 'Ticker'
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = stocks[(stocks['Year'] == 2019) & (stocks['Ticker'].isin(intersection2))][['GHG Scope 1']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

Out[15]:

OLS Regression Results

Dep. Variable:	GHG Scope 1	R-squared:	0.978
Model:	OLS	Adj. R-squared:	0.976
Method:	Least Squares	F-statistic:	512.1
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	5.81e-29
Time:	17:51:50	Log-Likelihood:	-375.49
No. Observations:	39	AIC:	759.0
Df Residuals:	35	BIC:	765.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	785.5107	859.091	0.914	0.367	-958.537	2529.558
2016	-0.2083	0.168	-1.240	0.223	-0.549	0.133
2017	-0.2277	0.353	-0.645	0.523	-0.945	0.489
2018	1.3555	0.256	5.291	0.000	0.835	1.876

Omnibus:	18.000	Durbin-Watson:	1.871
Prob(Omnibus):	0.000	Jarque-Bera (JB):	25.376
Skew:	1.321	Prob(JB):	3.09e-06
Kurtosis:	5.938	Cond. No.	9.25e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.25e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Looking at the regression results, we can see that our model is statistically significant. This means that the average of 2016, 2017, 2018 values are statistically significant at predicting 2019 values for GHG scope.

Now let's look at the stationarity for our previous year column by running a Dickey-Fuller test and printing the p values.

In [3]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 0.0006090779270817075
2017 p-value: 0.0014291590809764792
2018 p-value: 0.0024224655368029874
```

Since all the p-values are below 0.05 we conclude that all coefficients are stationary and our model is valid

Split by Industry

In [4]:

```
util_df = stocks[stocks['Sector'] == 'Utilities']
nrg_df = stocks[stocks['Sector'] == 'Energy']
```

In [5]:

```
companies_2018 = list(util_df[(util_df['Year'] == 2018) & (util_df['GHG Scope 1'] != 0)]['Ticker'])
companies_2019 = list(util_df[(util_df['Year'] == 2019) & (util_df['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016,2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(util_df[(util_df['Year'] == 2017) & (util_df['GHG Scope 1'] != 0) ]['Ticker'])
list2017_as_set = set(companies_2017)
intersection = list2017_as_set.intersection(intersection)

companies_2016 = list(util_df[(util_df['Year'] == 2016) & (util_df['GHG Scope 1'] != 0) ]['Ticker'])
list2016_as_set = set(companies_2016)
intersection = list2016_as_set.intersection(intersection)

x = util_df[(util_df['Year'].isin([2016, 2017,2018])) & (util_df['Ticker'].isin(intersection))][['Year', 'Ticke
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = util_df[(util_df['Year'] == 2019) & (util_df['Ticker'].isin(intersection))][['GHG Scope 1']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

Out[5]:

OLS Regression Results

Dep. Variable:	GHG Scope 1	R-squared:	0.977
Model:	OLS	Adj. R-squared:	0.974
Method:	Least Squares	F-statistic:	282.7
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	1.54e-16
Time:	17:51:28	Log-Likelihood:	-227.81
No. Observations:	24	AIC:	463.6
Df Residuals:	20	BIC:	468.3
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-46.2787	1132.759	-0.041	0.968	-2409.172	2316.615
2016	-0.1263	0.164	-0.768	0.451	-0.469	0.217
2017	-0.4000	0.343	-1.167	0.257	-1.115	0.315
2018	1.4359	0.266	5.406	0.000	0.882	1.990

Omnibus:	21.371	Durbin-Watson:	2.182
Prob(Omnibus):	0.000	Jarque-Bera (JB):	35.074
Skew:	1.674	Prob(JB):	2.42e-08
Kurtosis:	7.885	Cond. No.	1.07e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.07e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [6]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 0.20821263313929034
2017 p-value: 0.17183450369556258
2018 p-value: 0.012991677893899373
```

In [7]:

```
companies_2018 = list(nrg_df[(nrg_df['Year'] == 2018) & (nrg_df['GHG Scope 1'] != 0)]['Ticker'])
companies_2019 = list(nrg_df[(nrg_df['Year'] == 2019) & (nrg_df['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016,2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(nrg_df[(nrg_df['Year'] == 2017) & (nrg_df['GHG Scope 1'] != 0) ]['Ticker'])
list2017_as_set = set(companies_2017)
intersection = list2017_as_set.intersection(intersection)

companies_2016 = list(nrg_df[(nrg_df['Year'] == 2016) & (nrg_df['GHG Scope 1'] != 0) ]['Ticker'])
list2016_as_set = set(companies_2016)
intersection = list2016_as_set.intersection(intersection)

x = nrg_df[(nrg_df['Year'].isin([2016, 2017,2018])) & (nrg_df['Ticker'].isin(intersection))][['Year', 'Ticker',
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = nrg_df[(nrg_df['Year'] == 2019) & (nrg_df['Ticker'].isin(intersection))][['GHG Scope 1']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

```
C:\Users\YEET\Anaconda3\lib\site-packages\scipy\stats\stats.py:1450: UserWarning: kurtosistest only valid for n
>=20 ... continuing anyway, n=15
  "anyway, n=%i" % int(n))
```

Out[7]:

OLS Regression Results

Dep. Variable:	GHG Scope 1	R-squared:	0.989
Model:	OLS	Adj. R-squared:	0.986
Method:	Least Squares	F-statistic:	332.1
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	4.56e-11
Time:	17:51:28	Log-Likelihood:	-141.66
No. Observations:	15	AIC:	291.3
Df Residuals:	11	BIC:	294.2
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1812.4706	1139.090	1.591	0.140	-694.649	4319.590
2016	-0.3935	0.597	-0.659	0.524	-1.708	0.921
2017	1.6096	0.978	1.645	0.128	-0.544	3.763
2018	-0.2407	0.681	-0.353	0.730	-1.740	1.258

Omnibus:	13.985	Durbin-Watson:	1.694
Prob(Omnibus):	0.001	Jarque-Bera (JB):	10.490
Skew:	1.586	Prob(JB):	0.00527
Kurtosis:	5.594	Cond. No.	8.06e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.06e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

In [8]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 0.2506059326497947
2017 p-value: 0.22964762814593354
2018 p-value: 0.1878243121455755
```

Looking at the two regression results above that are split by industry, we see that the average of 2016, 2017, 2018 values are statistically significant at predicting 2019 values for GHG scope.

Percent Change

In [30]:

```
companies_2018 = list(stocks[(stocks['Year'] == 2018) & (np.isfinite(stocks.Percent_Change_GHG))]['Ticker'])
companies_2019 = list(stocks[(stocks['Year'] == 2019) & (stocks['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016,2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(stocks[(stocks['Year'] == 2017) & (np.isfinite(stocks.Percent_Change_GHG)) ]['Ticker'])
list2017_as_set = set(companies_2017)
intersection = list2017_as_set.intersection(intersection)

companies_2016 = list(stocks[(stocks['Year'] == 2016) & (np.isfinite(stocks.Percent_Change_GHG)) ]['Ticker'])
list2016_as_set = set(companies_2016)
intersection = list2016_as_set.intersection(intersection)

x = stocks[(stocks['Year'].isin([2016, 2017,2018])) & (stocks['Ticker'].isin(intersection))][['Year', 'Ticker',
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = stocks[(stocks['Year'] == 2019) & (stocks['Ticker'].isin(intersection))][['Percent_Change_GHG']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

Out[30]:

OLS Regression Results

Dep. Variable:	Percent_Change_GHG	R-squared:	0.062
Model:	OLS	Adj. R-squared:	-0.029
Method:	Least Squares	F-statistic:	0.6776
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	0.572
Time:	17:54:04	Log-Likelihood:	-11.026
No. Observations:	35	AIC:	30.05
Df Residuals:	31	BIC:	36.27
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.5008	0.608	0.824	0.416	-0.739	1.740
2016	0.4113	0.374	1.099	0.280	-0.352	1.174
2017	0.2150	0.567	0.379	0.707	-0.942	1.372
2018	-0.1063	0.204	-0.520	0.607	-0.523	0.310

Omnibus:	12.077	Durbin-Watson:	2.391
Prob(Omnibus):	0.002	Jarque-Bera (JB):	25.692
Skew:	0.553	Prob(JB):	2.64e-06
Kurtosis:	7.049	Cond. No.	25.7

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

When we try to predict the percentage change of this year with the values of last three year we see that the 2016, 2017, and 2018 values are not statistically significant at predicting 2019 values.

Now let's look at the stationarity for our previous year column by running a Dickey-Fuller test and printing the p values.

In [31]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 4.408304121791802e-07
2017 p-value: 5.2464738997598225e-08
2018 p-value: 1.3253171599285162e-06
```

Since all the p-values are above 0.05 we conclude that all coefficients are stationary and however this doesn't change the fact that this model performs poorly.

Percentage Change for Each Industry

In [34]:

```
companies_2018 = list(util_df[(util_df['Year'] == 2018) & (np.isfinite(util_df.Percent_Change_GHG))]['Ticker'])
companies_2019 = list(util_df[(util_df['Year'] == 2019) & (util_df['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016, 2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(util_df[(util_df['Year'] == 2017) & (np.isfinite(util_df.Percent_Change_GHG)) ]['Ticker']
list2017_as_set = set(companies_2017)
intersection = list2017_as_set.intersection(intersection)

companies_2016 = list(util_df[(util_df['Year'] == 2016) & (np.isfinite(util_df.Percent_Change_GHG)) ]['Ticker']
list2016_as_set = set(companies_2016)
intersection = list2016_as_set.intersection(intersection)

x = util_df[(util_df['Year'].isin([2016, 2017,2018])) & (util_df['Ticker'].isin(intersection))][['Year', 'Ticke
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = util_df[(util_df['Year'] == 2019) & (util_df['Ticker'].isin(intersection))][['Percent_Change_GHG']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

Out[34]:

OLS Regression Results

Dep. Variable:	Percent_Change_GHG	R-squared:	0.050
Model:	OLS	Adj. R-squared:	-0.108
Method:	Least Squares	F-statistic:	0.3156
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	0.814
Time:	17:54:59	Log-Likelihood:	2.0383
No. Observations:	22	AIC:	3.923
Df Residuals:	18	BIC:	8.287
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.2884	0.548	2.353	0.030	0.138	2.439
2016	0.0543	0.341	0.159	0.875	-0.661	0.770
2017	-0.4576	0.566	-0.809	0.429	-1.646	0.731
2018	-0.0211	0.310	-0.068	0.947	-0.673	0.631

Omnibus:	28.004	Durbin-Watson:	1.843
Prob(Omnibus):	0.000	Jarque-Bera (JB):	57.161
Skew:	-2.188	Prob(JB):	3.87e-13
Kurtosis:	9.574	Cond. No.	28.0

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [35]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 1.0
2017 p-value: 0.0016709477869441232
2018 p-value: 0.0
```

In [13]:

```
companies_2018 = list(nrg_df[(nrg_df['Year'] == 2018) & (np.isfinite(nrg_df.Percent_Change_GHG))]['Ticker'])
companies_2019 = list(nrg_df[(nrg_df['Year'] == 2019) & (nrg_df['Ticker'].isin(companies_2018))]['Ticker'])

#Getting companies that have reported for 2016, 2017, and 2018 in a years
list2018_as_set = set(companies_2018)
intersection = list2018_as_set.intersection(companies_2019)

companies_2017 = list(nrg_df[(nrg_df['Year'] == 2017) & (np.isfinite(nrg_df.Percent_Change_GHG)) ]['Ticker'])
list2017_as_set = set(companies_2017)
intersection = list2017_as_set.intersection(intersection)

companies_2016 = list(nrg_df[(nrg_df['Year'] == 2016) & (np.isfinite(nrg_df.Percent_Change_GHG)) ]['Ticker'])
list2016_as_set = set(companies_2016)
intersection = list2016_as_set.intersection(intersection)

x = nrg_df[(nrg_df['Year'].isin([2016, 2017,2018])) & (nrg_df['Ticker'].isin(intersection))][['Year', 'Ticker',
x = x.pivot(index = 'Ticker', columns = ['Year']).reset_index()
x.columns = x.columns.droplevel(0)

x = x.rename_axis(None, axis=1)
x = x.drop(columns = '')

y = nrg_df[(nrg_df['Year'] == 2019) & (nrg_df['Ticker'].isin(intersection))][['Percent_Change_GHG']]
x.index = y.index

# x_train, x_test, y_train, y_test = train_test_split(
#     X, y, test_size=0.2, random_state=42)
x = sm.add_constant(x)
sm.OLS(y, x).fit().summary()
```

```
C:\Users\YEET\Anaconda3\lib\site-packages\scipy\stats\stats.py:1450: UserWarning: kurtosistest only valid for n
>=20 ... continuing anyway, n=13
  "anyway, n=%i" % int(n))
```

Out[13]:

OLS Regression Results

Dep. Variable:	Percent_Change_GHG	R-squared:	0.165
Model:	OLS	Adj. R-squared:	-0.113
Method:	Least Squares	F-statistic:	0.5926
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	0.635
Time:	17:51:28	Log-Likelihood:	-5.9413
No. Observations:	13	AIC:	19.88
Df Residuals:	9	BIC:	22.14
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2958	1.488	0.199	0.847	-3.070	3.661
2016	0.5061	0.993	0.509	0.623	-1.741	2.753
2017	0.6147	1.098	0.560	0.589	-1.870	3.099
2018	-0.2034	0.404	-0.504	0.627	-1.117	0.710

Omnibus:	2.417	Durbin-Watson:	2.036
Prob(Omnibus):	0.299	Jarque-Bera (JB):	0.968
Skew:	0.663	Prob(JB):	0.616
Kurtosis:	3.167	Cond. No.	28.4

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The results don't change when we split by industry and run a regression for each industry. The 2016, 2017, and 2018 percentage change values are not statistically significant at predicting 2019 values for both industries.

In [14]:

```
for column in x.columns[1:]:
    print(f'{column} p-value: {adfuller(x[column])[1]}')
```

```
2016 p-value: 0.11939475084720846
2017 p-value: 0.0918115544800363
2018 p-value: 0.36273443778876535
```

Conclusion

We have seen that when we use actual values GHG Scope of 2016, 2017, 2018 is statistically significant at predicting 2019 values. However, when we try to predict the precentage change of GHG Scope in 2018-2019, using 2015-2018 values is not statistically significant.

We will continue our analysis on 'Environmental Intensity Time Series - Level Regression' notebook.