Predicting CO2 equivalent emissions based on economic and energetic metrics

CS-C3240 Machine Learning

Fall 2023

Contents

1 Introduction

2 Problem Formulation

In this project, we aim to apply supervised learning (regression) to predict greenhouse emissions (expressed as CO2 equivalent) using a variety of metrics sourced from The World Bank OpenData and Our World in Data datasets. The two datasets have been deemed compatible to be used at the same time, since they collect internationally-standardized and recognized data (for example [9]).

The features we have selected include 1. Gross Domestic Product (GDP) per capita at Parity of Purchasing Power (PPP) [current international $], 2. access to electricity [% of total population], 3. yearly energy use per capita [kWh], 4.

population growth [% of total population], 5. and the share of electricity from low-carbon sources [% of generated electricity].

We chose these specific features because we assumed they have significant relevance and likelihood of correlation with our desired label: the emission of greenhouse gases.
GDP per capita PPP reflects the economic status of a country taking account of the different purchasing power. A larger GDP can increase the energy consumption of a nation. On the other hand, it can improve overall access to expensive low carbon technology.
In order to account for this possibility, we included amongst our considered features the share of electricity generated from low-carbon sources. This provides information regarding the energy mix of a region and its carbon footprint.
Access of population to electricity gives an understanding of the electricity consumption spreading towards the population. A higher percentage correlates with higher energy consumption and, depending on its generating source, emissions.
Moreover, energy use directly contributes to greenhouse emissions.
Population growth can have a significant impact on overall electricity need, thus forecasting an increase in greenhouse emissions.

Each datapoint includes a single instance of the six features (related to a specific country in an year included in the range 1990-2020), as well as the value for the estimated CO2 emissions of that country in that year as the label.

3 Methodology

Our dataset comprises of 4046 datapoints. Each point corresponds to one of 205 countries in a specific year 1990-2020 in the datasets of The World Bank or Our World in Data (References [1-6]). To attain our end goal (predicting the emission of CO2, based on societal consumptions and population characteristics) we plan to train two different regression models: a Linear Regression model and a Decision Tree.

3.1 Data pre-processing

In order to integrate the two different sources of data, we used the official country code. Moreover, on a more technical note, the data was presented in the .csv files in opposite ways, since Our World in Data was structured vertically and The World Bank OpenData horizontally. We then processed our data format to the horizontal structure (each row represents a country, years are positioned in different columns).
After this stage, we created datapoints from year and country-specific features. We also removed aggregation of countries (such as the European Union, continents or other groups), because some features were not available for them and

to maintain coherence in our datapoints. We have then discarded instances in which at least one feature was missing for each datapoint, this was performed by merging the different datasets in a Pandas [7] DataFrame object, and using the .dropna() function. Considering the differences in the used datasets, this caused a significant reduction of the available datapoints, which decreased from a potential number 205(countries)*31(years) = 6355 to the previously mentioned 4046.
Initially the idea was to try and include the time-dependence of each feature in our study. This would have meant to create a single datapoint per country (as opposed to the current set of max 31 datapoints, considering all possible range of years), where each year would correspond to a different feature. This option was later discarded, since the number of features would have become comparable to the number of datapoints thus increasing the likelihood of overfitting.
The year range was chosen to be 1990-2020, due to being the time interval that balanced the amount of available data, time range, and relevance to our study, since carbon emissions have become more relevant in the recent decades and some useful data was gathered only recently.

3.2 Linear regression

Linear regression was chosen as a potential model. The chosen loss function is square errors loss function, as it is proven to be very robust for linear regression, and more generally polynomial regression, models [8].
The relatively small number of datapoints does not allow to create large partitions for validations. The chosen ratio of training, validation and testing is 76, 4, 20. We plan to first attempt dividing the year span in chronological order for each partition, in order to check for a possible time-dependence of our datapoints. In our case, this would mean to consider the years 1990-2014 as training data and validation and 2015-2020 as testing data. We would like to compare the obtained accuracy with a more random sampling approach (with equal ratios of division between training validation and testing). This could in theory allow us to gain some insight on whether or not training only on past data affects the prediction accuracy negatively.
In order to perform the validation of the trained model, we would like to apply k-fold cross validation, choosing one year out of the interval 1990-2014 as the testing datapoints or as a fraction of the randomly-sampled training+validation (76%+4%=80%) segment.

3.3 Decision tree

4 Results

5 Conclusion

6 Bibliography/References

[1] Total greenhouse gas emissions (kt of CO2 equivalent). The World Bank OpenData. Available at: https://data.worldbank.org/indicator/EN.ATM.GHGT.KT.CE
[2] GDP per capita, PPP (current international $). The World Bank OpenData. Available at: https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.CD
[3] Population growth (annual %). The World Bank OpenData. Available at: https://data.worldbank.org/indicator/SP.POP.GROW
[4] Access to electricity (% of population). The World Bank OpenData. Available at: https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS
[5] Energy use per capita (kWh). Our World in Data. Available at: https://ourworldindata.org/grapher/per-capita-energy-use
[6] Share of electricity from low-carbon sources (%). Our World in Data. Available at: https://ourworldindata.org/electricity-mix
[7] Pandas. Python package. Available at: https://pandas.pydata.org/
[8] A. Jung, "Machine Learning: The Basics," Springer, Singapore, 2022
[9] Ember's European Electricity Review. Available at: https://ember-climate.org/insights/research/european-electricity-review-2023/

7 Appendix

CO2_prediction

September 22, 2023

[308]:
```
import numpy as np
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
import seaborn as sn
```

Pre-processing and creation of the datapoints

[309]:
```
lowC = pd.read_csv("/notebooks/MLProject/data/Clean/ShareElectricityLowCarbon.
 ↪csv", sep=";")
accessElec = pd.read_csv("/notebooks/MLProject/data/Clean/elecAccess.csv",␣
 ↪sep=";")
popGrowth = pd.read_csv("/notebooks/MLProject/data/Clean/popGrowth.csv", sep=";
 ↪")
energyUse = pd.read_csv("/notebooks/MLProject/data/Clean/energyUse.csv", sep=";
 ↪")
GDP = pd.read_csv("/notebooks/MLProject/data/Clean/GDP.csv", sep=";")
population = pd.read_csv("/notebooks/MLProject/data/Clean/population.csv",␣
 ↪sep=";")

totEmissions = pd.read_csv("/notebooks/MLProject/data/Clean/totEmissions.csv",␣
 ↪sep=";")

lowC.at[0,"Year"] = 2000
lowC.at[1,"1990"] = 0

y = np.array([])

X = np.array([[0,0,0,0,0,0]])

names = np.array([])
for year in range(1990,2021):
    for code in lowC["Code"]:
        X_1 = lowC.loc[lowC['Code'] == code][str(year)].values
        X_2 = accessElec.loc[accessElec['Country Code'] == code][str(year)].
 ↪values
        X_3 = popGrowth.loc[popGrowth['Country Code'] == code][str(year)].values
```

```
        X_4 = energyUse.loc[energyUse['Code'] == code][str(year)].values
        X_5 = GDP.loc[GDP['Country Code'] == code][str(year)].values
        X_6 = population.loc[population['Country Code'] == code][str(year)].
↪values

        if(len(X_1) == 1 and len(X_2) == 1 and len(X_3) == 1 and len(X_4) == 1␣
↪and len(X_5) == 1 and len(X_6) == 1 and len(totEmissions.
↪loc[totEmissions['Country Code'] == code][str(year)].values)==1 ):
            X = np.vstack( (X, [ X_1[0],X_2[0],X_3[0],X_4[0],X_5[0],X_6[0] ]) )
            y = np.append(y, totEmissions.loc[totEmissions['Country Code'] ==␣
↪code][str(year)])
            names = np.append(names, code + str(year))
```

[310]:

```
X_ = pd.DataFrame(X[1:])
X_.insert(6, "y", y, True)
X_.insert(7, "labels", names, True)
X_.head(5)
```

[310]:

	0	1	2	3	4	5
0	NaN	NaN	0.202434	2968.3160	NaN	10694796.0
1	0.000000	NaN	3.345144	1972.4146	3283.170843	11828638.0
2	86.363640	100.0000	1.799086	10214.5980	2549.746801	3286542.0
3	0.000000	100.0000	5.869033	181538.7700	83843.224680	1900151.0
4	49.264286	92.1548	1.456403	15718.7030	7183.583826	32637657.0

	y	labels
0	11630.79506	AFG1990
1	43185.60921	AGO1990
2	11181.07427	ALB1990
3	78601.83891	ARE1990
4	249188.76120	ARG1990

Final cleanup of NaN

[311]:

```
X_ = X_.dropna()
X_.columns = ["lowC","accessElec","popGrowth","energyUse","GDP",␣
↪"population","y","labels"]
X_ = X_.reset_index()
X_.head(5)
```

[311]:

	index	lowC	accessElec	popGrowth	energyUse	GDP
0	2	86.363640	100.0000	1.799086	10214.598	2549.746801
1	3	0.000000	100.0000	5.869033	181538.770	83843.224680
2	4	49.264286	92.1548	1.456403	15718.703	7183.583826
3	7	9.970221	100.0000	1.480047	60597.098	17381.440010
4	8	66.206894	100.0000	0.762002	43648.800	19473.081960

```
   population            y   labels
0   3286542.0   11181.07427  ALB1990
1   1900151.0   78601.83891  ARE1990
2  32637657.0  249188.76120  ARG1990
3  17065128.0  490531.25070  AUS1990
4   7677850.0   76630.27662  AUT1990
```

Some basic information about the dataset

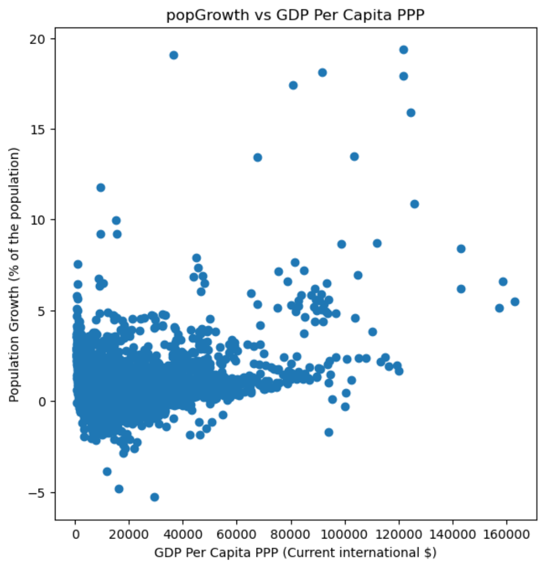
[312]:
```
X_.info()
```

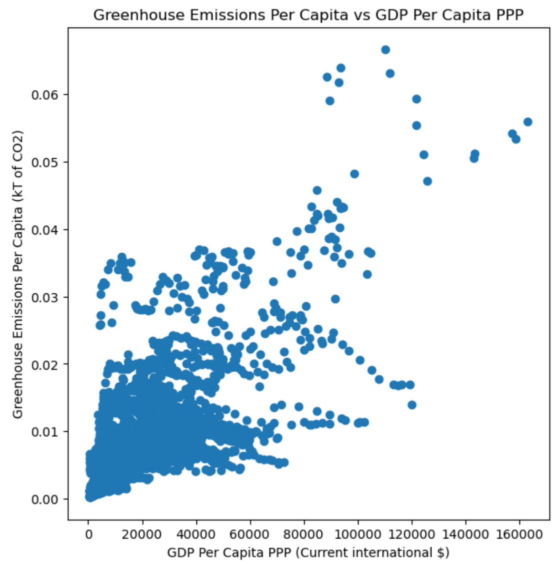
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4046 entries, 0 to 4045
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   index       4046 non-null   int64
 1   lowC        4046 non-null   float64
 2   accessElec  4046 non-null   float64
 3   popGrowth   4046 non-null   float64
 4   energyUse   4046 non-null   float64
 5   GDP         4046 non-null   float64
 6   population  4046 non-null   float64
 7   y           4046 non-null   float64
 8   labels      4046 non-null   object
dtypes: float64(7), int64(1), object(1)
memory usage: 284.6+ KB
```

Some plot to test the data

[313]:
```
# Visualize data
fig, axes = plt.subplots(1, 2, figsize=(15,7))
axes[0].scatter(X_['GDP'], X_['popGrowth'])
axes[0].set_xlabel("GDP Per Capita PPP (Current international $)")
axes[0].set_ylabel("Population Growth (% of the population)")
axes[0].set_title("popGrowth vs GDP Per Capita PPP")

axes[1].scatter(X_['GDP'], X_['y'] / X_['population']) #use Emissions Per␣
 ↪Capita to tackle the impact of the population size in the plot
axes[1].set_title('Greenhouse Emissions Per Capita vs GDP Per Capita PPP')
axes[1].set_xlabel("GDP Per Capita PPP (Current international $)")
axes[1].set_ylabel('Greenhouse Emissions Per Capita (kT of CO2)')
plt.show()
```

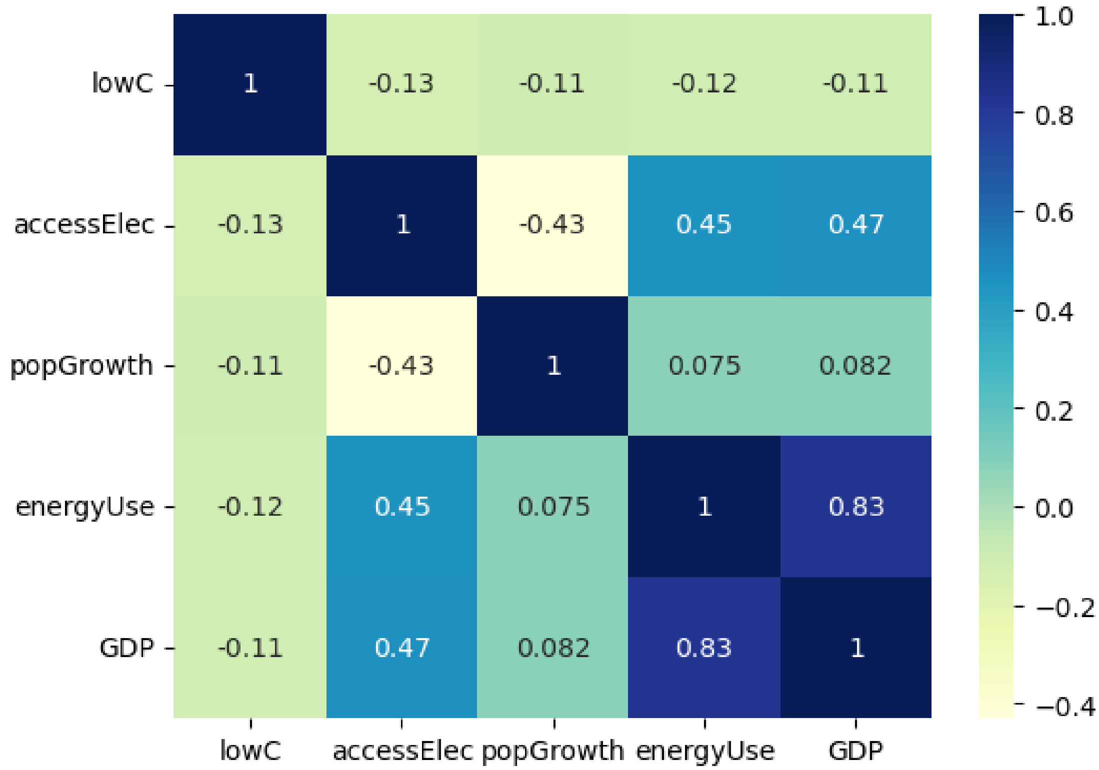




Correlation matrix

[314]:
```
print(X_.iloc[:, 1:6].corr())
dataplot = sn.heatmap(X_.iloc[:, 1:6].corr(), cmap="YlGnBu", annot=True)
```

	lowC	accessElec	popGrowth	energyUse	GDP
lowC	1.000000	-0.133049	-0.107967	-0.116596	-0.112042
accessElec	-0.133049	1.000000	-0.430143	0.454620	0.467831
popGrowth	-0.107967	-0.430143	1.000000	0.075040	0.081901
energyUse	-0.116596	0.454620	0.075040	1.000000	0.829737
GDP	-0.112042	0.467831	0.081901	0.829737	1.000000



lowC
accessElec
popGrowth
energyUse
GDP
1
-0.13
-0.11
-0.12
-0.11
-0.13
1
-0.43
0.45
0.47
-0.11
-0.43
1
0.075
0.082
-0.12
0.45
0.075
1
0.83
-0.11
0.47
0.082
0.83
1
lowC
accessElec
popGrowth
energyUse
GDP
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4