Carbon Emission and Renewable Energy

ABSTRACT

This article describes how much carbon emmitted, how it is distributed by years and region.
The purpose of this article is to determine which regions and countries have more carbon emissions and to find the answer to the question which countries in the future can reduce carbon emissions and move to renewable energy sources, as promised by the Paris agreement.

Data Analysis and Machine Learning Concepts

Dataset→ Integrated Development Environment

Data→Regression Algorithm

Keywords

Carbon emission; renewable energy; distribution; region; Paris agreement; Machine Learning; Data Analysis

1. INTRODUCTION

Since we are consuming a lot of energy in today's world, renewable energy is one of the most important topics with our responsibility towards the earth

Renewable energy sources began to gain importance as they would never run out, were harmless to the environment and cost much less than fossil fuels. When fuels such as oil, natural gas and coal are burned, harmful chemicals such as carbon dioxide, sulfur and nitrogen are released into the atmosphere. In an era of global warming, it has been determined that there has been an increase in the average temperature of our world due to the increasing CO2 concentration in the last century. It is interesting topic because the world needs it, and we think that our country has a great potential. We are looking for answers to which country has investments in this regard.

We want to analyze the rate of increase/decrease of these investments over the years. We want to demonstrate the relationship between CO2 emissions and renewable energy sources.

2. THE APPROACH

2.1 Understand The Data

We've done a data set study that will include co2 emissions and renewable energy. We've decided to use BP's data set [1]. The most important data columns we have drawn from bp.csv dataset and used are: Country, year, region, co2_mtco2, renewables_mtoe, renewables_ej. We also used the co2_mtco2_ej and ren_power_pro_ej columns that are created transforming other columns' data.

2.2 Extracting Feature

The dataset was made up of 95 columns in the first place. There were many columns that weren't necessary for us. We deleted the columns that were unnecessary for our project, and we ended up with 19 columns including the columns we have created. These columns are shown in picture 1.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6629 entries, 0 to 6628
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Country           6629 non-null   object
 1   Year              6629 non-null   int64
 2   pop               6603 non-null   float64
 3   ISO3166_alpha3    5763 non-null   object
 4   ISO3166_numeric   5763 non-null   float64
 5   Region            5763 non-null   object
 6   SubRegion         5763 non-null   object
 7   OPEC              5763 non-null   float64
 8   EU                5763 non-null   float64
 9   OECD              5763 non-null   float64
 10  CIS               5763 non-null   float64
 11  co2_mtco2         5003 non-null   float64
 12  ren_power_ej      5003 non-null   float64
 13  ren_power_mtoe    5003 non-null   float64
 14  ren_power_twh     5003 non-null   float64
 15  renewables_ej     5003 non-null   float64
 16  renewables_mtoe   5003 non-null   float64
 17  co2_mtco2_ej      5003 non-null   float64
 18  ren_power_pro_ej  5003 non-null   float64
dtypes: float64(14), int64(1), object(4)
memory usage: 984.1+ KB
```

Picture – 1. Dataset info.

These columns can be grouped into two. First group represents the columns coming from the original bp.csv dataset. The second group stands for the columns that we have created and filled with the data calculated using existing data. To get a better understanding of the data that we have used we can summarize what are these abbreviated column names stand for as below:

First Group

co2_mtco2: million tonnes of carbon dioxide

ren_power_ej: renewable power in exajoules

ren_power_mtoe: renewable power in million tonnes of oil equivelant

ren_power_twh: produced renewable power in terawatt hours

renewables_ej: consumed renewable energy in exajoules

renewables_mtoe: consumed renewable energy in million tonnes of oil equivelant

Second Group

co2_mtco2_ej: the energy equivelant that caused that amount of million tonnes of carbon dioxide emmission in exajoules. We created this column multiplying each entry on the co2_mtco2 column with 0.9324. The reason behind this was to acquire the data in exajoules to make accurate calculations with other columns in exajoules. We obtained the multiplication criterion based on the

U.S Energy Information Administration's official website eia.gov's publication. This ınformation is not a precise transformation criterion but such criterion does not exist, so we assumed this information published as our transformation criterion. [3] According to this information of EIA:

4.01 trillion kWh from all energy sources = 1.55 billion metric tonnes of CO2

This means:

2.59 tWh = 1 mtco2 (million tonnes of co2)

Since 1 tWh = 0.0036 ej:

2.59 tWh = 1 mtco2 = <u>0.9324 ej</u>

ren_power_pro_ej: this column represents the amount of produced renewable energy in exajoules. We obtained the data values on this column by multiplying ren_power_twh column's entries with 0.0036, because 1 tWh = 0.0036 ej.

2.3 Main Task

Our primary goal is to look at the countries' years of co2 emissions and renewable energy usage, and to see if they can reset co2 emissions by 2050, which they promised in the paris agreement. To do this, we will first look at years of co2 emissions and renewable energy use. Then using regression model, we will show the expected energy deficit for some countries chosen by us. (co2_mtco2_ej, renewables_ej and ren_power_pro_ej) in the dataset.

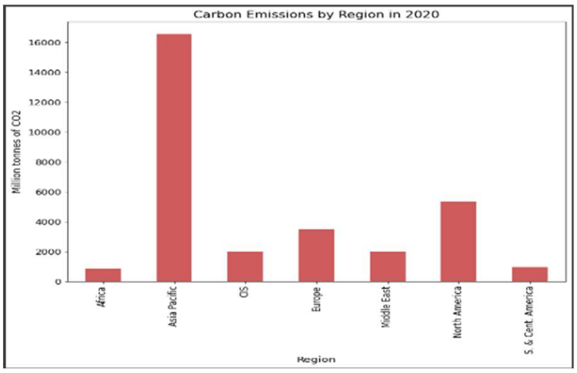
2.4 Methodology

To reach the results we acquired we used filtering, grouping and plotting tools of various libraries such as pandas, matplotlib.pyplot and seaborn.

To make predictions and see whether some countries will be able to balance carbon emissions with rates that increase their renewable energy for 2050 we used regression supervised learning algorithm. To do this we included Polynomial class from numpy.polynomial.

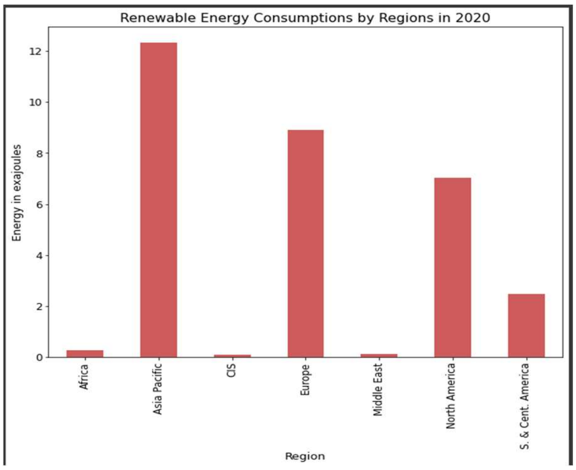
3. EXPERIMENTS

First, we cleaned the irrelevant data to our purposes from our dataset. Secondly, we found the number of countries in each region. Later, filtered the data belonging to 2020 and on this piece of dataset we plotted million tons of carbon dioxide (CO2) that emissioned by each region.



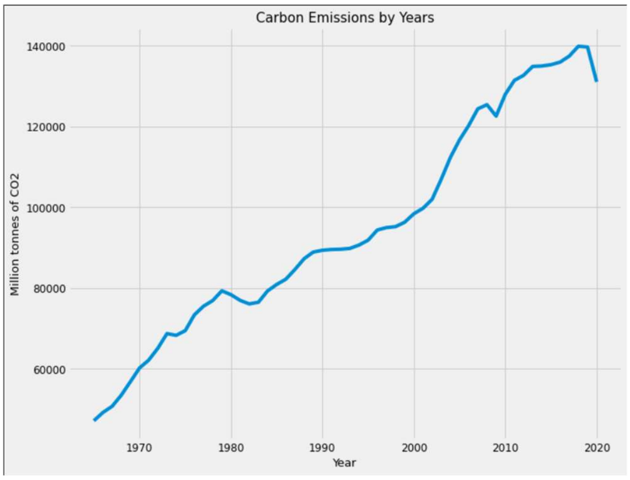
Picture – 2. Carbon Emission by region in 2020

On the same piece of dataset this time, we showed how much renewable energy is consumed in each region in 2020 on a bar plot.



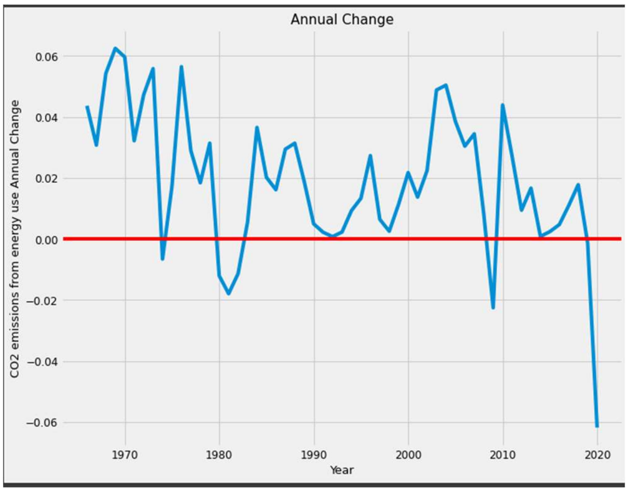
Picture – 3. Renewable energy consumptions by region in 2020

Furthermore, on a time series plot we showed the total CO2 emission according to years. To obtain the total CO2 for years, we gouped the data by years and then got the sum of co2_mtco2 for each year.



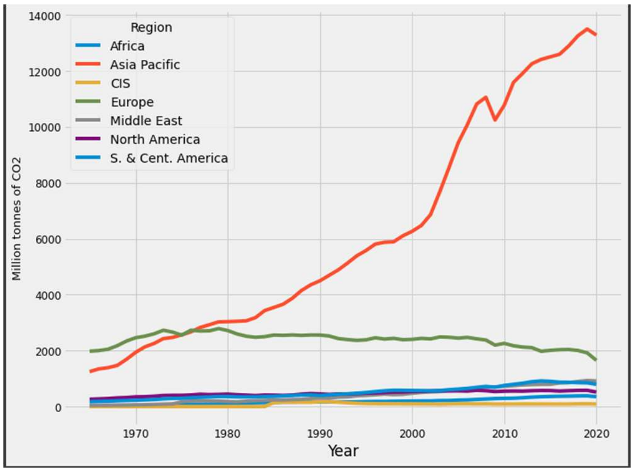
Picture – 4. CO2 consumptions by years on time series plot

In the next plot we showed the percentage of change in carbon emissions each year according to the previous year. We found annual changes using Pandas library's pct_change() function and plotted it on a time series.

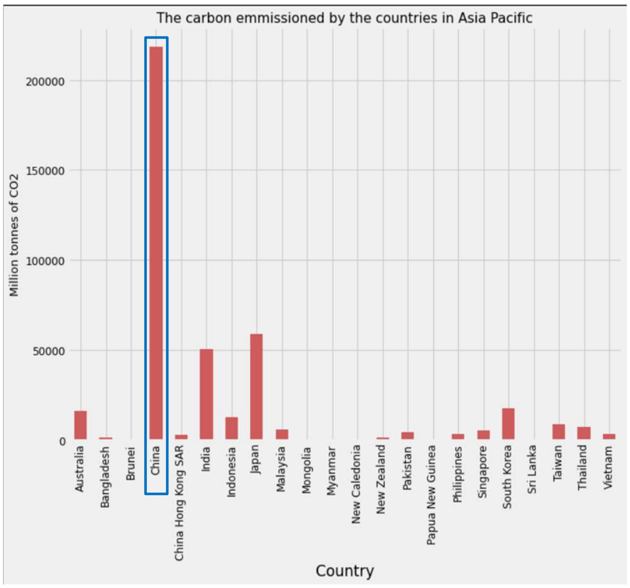


Picture – 5. Percentage of annual change in co2 emission

Later, we have shown the carbon emission of each region in years on a time series plot. On that plot, we detected a steep rise of carbon emission for Asia Pacific, and we wondered the reason behind that. So, on a bar plot we specifically showed the carbon emissions of the countries in the Asia Pacific. What we saw is that China has an ultrahigh carbon emission comparing the other countries in its region. Then, to observe the course of carbon emission amount of China in years we plotted it in a time series plot (not included in the document but exists in the code). What we conclude on that is with the development in China's industry, the carbon emission it caused has also increased.



Picture -6. Million tonnes of carbon emission of each region in years



4. ANALYSE

Since one of our main goals is to predict the countires' carbon emmission and renewable energy production and consumption balance due to 2050, we were in need to find a proper way of finding energy deficit. When this deficit is reduced, that means that country is getting closer to the neutral point. We calculated that deficit as follows:

Deficit = (renewables_ej + co2_mtco2_ej) - ren_power_pro_ej

When at some point, the produced renewable energy is enough to satisfy the both renewable energy usage and all kinds of energy usage causing carbon emmission -deficit=0-, the neutral point is caught.

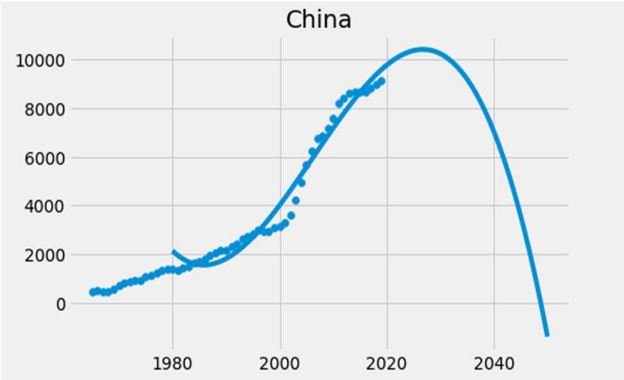
As we mentioned before in the Methodology section we included Polynomial class from numpy.polynomial. To fit polynomials of various degrees to different countries' data we utilized the fit() function of Polynomial class. This fit() function returns a series instance that is the least squares fit to the data y sampled at x [4].

We worked on some of the biggest countries in the world. These are: China, Turkey, France, the US, India, South Korea, Spain, Germany, Japan and Russia. On each plot's y axis is for deficit. In the code all these plots are subplots of a bigger plot and they have common x and y axis labels.

Another point is that, we excuded 2020's deficit value from the scatterplots because, due to pandemic restrictions throughout the world there was an abnormal decline in the emmited carbon for many countries in 2020. This sudden and abnormal decline was affecting our regression algorithm badly and causing inaccurate results. So it was important to exclude 2020 from our working field.

For China we fitted a polynomial of degree 3 and fitted over the data after 1980. The reason we eliminated the data before 1980 is that it was affecting the polynomial that we were trying to fit to our data improperly.

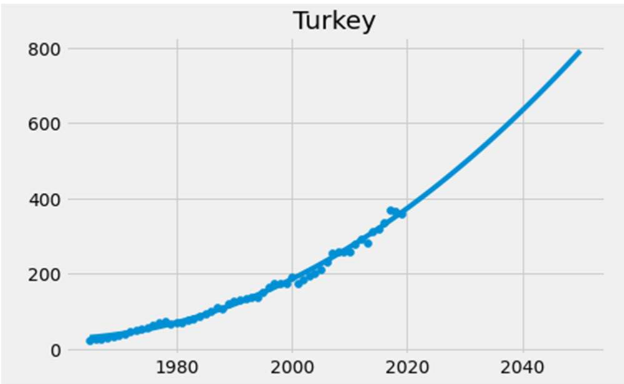
As it can be seen from the plot, by the end of 2050, China looks as if it will neutralize the carbon emmission.



China (Polynomial degree = 3)

To Turkey's deficit data, we fitted a polynomial of degree 2. The data is almost linearly formed and implementing linear regression on it also gives acceptable results. However, polynomial regression with a polynomial of degree 2 represents the data better.

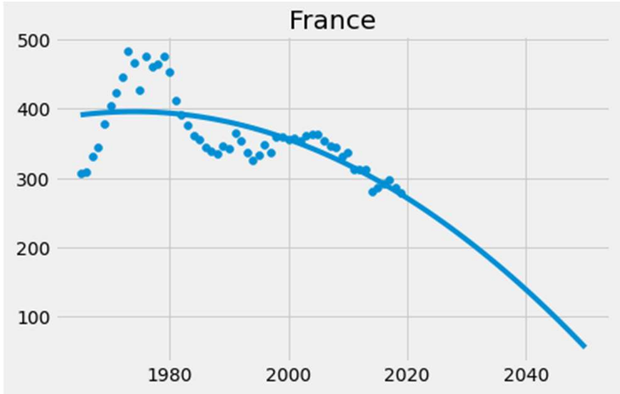
We can conclude from the plot that, Turkey maay not be able to neutralize the carbon emmision by the end of 2050 if it does not take a fast action.



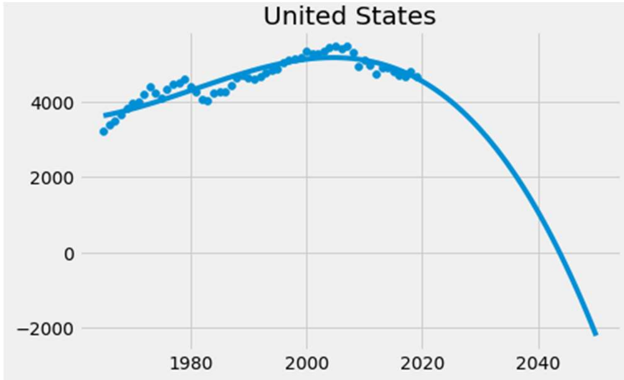
Turkey (Polynomial degree = 2)

For France, the polynomial degree is adjusted to 2. The curve is not representing the data very well for years before 2000. We could use a higher degree of polynomial to fit a better curve but what we acquire when we do this is not a realistic trajectory of line for the future. Due to this fact, we sacrificed a better fitting polynomial.

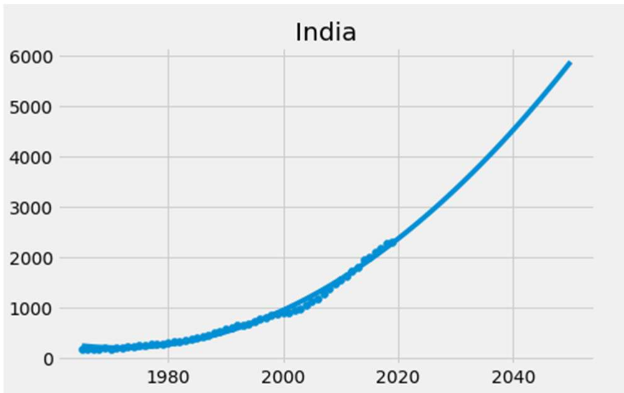
France has a quite good pace to reduce the deficit. It looks like it will reach it's 2050 goal only just in time.



France (Polynomial degree = 2)

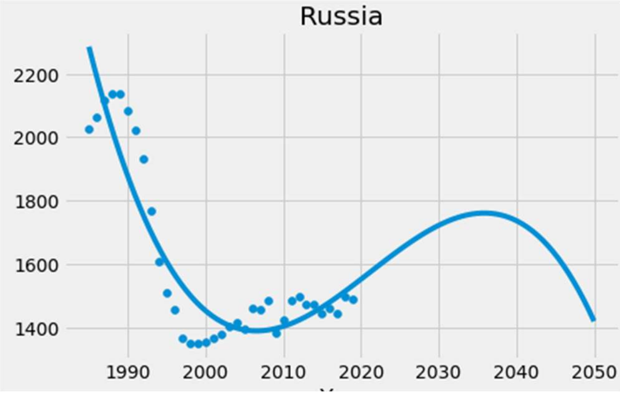


United states (Polynomial degree = 3)



India (Polynomial degree = 2)

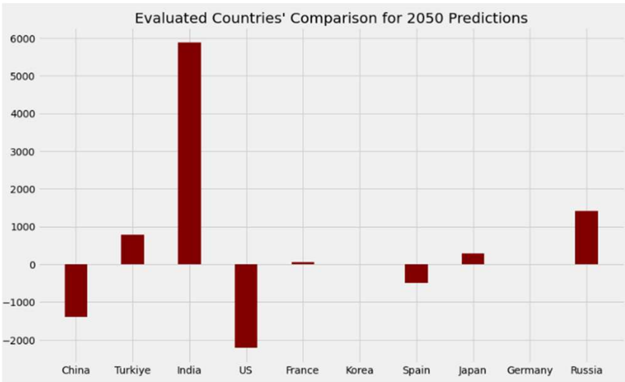
Due to lack of information before 1985 for Russia, the deficit value for those years cannot be calculated. That's why, we excluded that portion of data for this plot.



Russia (Polynomial degree = 3)

Last but not least, we gathered all these countries' predicted deficit to neutralize the carbon emmission for 2050 in another bar plot. On this plot we can compare the countries according to their progression in terms of 2050 goals pointed out in Paris Agreement.

What we see is that, India is far behind other countries. On the other hand, the US looks like it is ahead of the game. One surprising result is for China. Although it has high carbon emmission, it looks like its increasing capacity to produce renewable energy will close the gap.



Evaluated Countries Comparison for 2050 Prediction

5. RESULT

We have successfully completed our project. We calculated the expected points where the countries will be at by the end of 2050 from the point of neutralizing carbon emmission. In order to see the results in this direction, we drew the graphs of certain countries and made it easy to see.

6. CONCLUSION

If we need to make inferences in line with the results and graphics we have obtained, we understand that some countries are not far from being carbon neutral. But developing countries are not as respectful to nature as other countries. For example, as can be seen from the graph, India still seems to emit significant carbon emissions in 2050. In order to prevent this, countries should support investments in renewable energy sources, raise awareness of people and fund them.

The results and graphics had a life-changing impact on all group members, requiring more respect for nature. We hope anyone else who sees these results can learn the lesson. If you want to give an example and learn lessons from daily life; The fact that the carbon footprint per capita in America is high may be due to the fact that there are too many internal combustion engine vehicles, the settlements are far from each other, and any transportation process causes much more carbon emissions. The emergence of such high carbon emissions for India can be solved by raising awareness of the public and the government taking much more strict decisions.

7. REFERENCES

[1]] BP dataset https://www.bp.com/en/global/corporate/energyeconomics/statistical-review-of-worldenergy/downloads.html

[2] Paris Agreement https://wwftr.awsassets.panda.org/downloads/10_sorud a_paris_anlamas_web.pdf

[3] *Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA)*. Eia.gov. (2022). Retrieved 26 January 2022, from https://www.eia.gov/tools/faqs/faq.php?id=74&t=11#:~:text=In%202020%2C%20total%20U.S.%20electricity,CO2%20emissions%20per%20kWh.

[4] *numpy.polynomial.polynomial.Polynomial.fit — NumPy v1.22 Manual*. Numpy.org. Retrieved 26 January 2022, from https://numpy.org/doc/stable/reference/generated/numpy.polynomial.polynomial.Polynomial.fit.html#numpy.polynomial.polynomial.Polynomial.fit.