

**BCIS 5110.001**

**PROGRAMMING LANGUAGE FOR BUSINESS ANALYTICS**

**GROUP PROJECT**

***TOTAL EMISSIONS PER COUNTRY***

**Group Number - 11**

**Group Members**

**Adhvayth Reddy Angula -**

**Vanteru Dhruvakumar Reddy - 11518158**

**Shiva Teja -**  **11589646**

**EXECUTIVE SUMMARY**

The considered dataset has been a useful tool that offers insights into regional disparities, per capita emissions, greenhouse gas emissions, and worldwide trends in emissions. Despite attempts to cut emissions, the data reveals that overall total emissions have grown over the last 20 years, with variances between regions and nations. The main gases that contribute to global emissions are carbon dioxide (CO2), methane (CH4), and nitrous oxide (N2O). Countries' emissions per capita vary, with some having large emissions while having tiny populations. This information may help with the creation of policies and decision-making that is based on facts in order to combat climate change and create a more sustainable future.

**PROJECT BACKGROUND**

One of the most important issues confronting mankind right now is climate change, which significantly influences the environment, society, and the economy. Large amounts of greenhouse gases (GHGs) are released into the atmosphere as a result of the use of fossil fuels, deforestation, industrial operations, and other human activities. GHGs trap heat and cause global warming. It is critical to comprehend global emissions patterns and the variables influencing them as the world struggles with the requirement to mitigate the effects of climate change and move to a low-carbon economy (Frazzetto *et al.* 2019). The "Total Emissions per Country from 2000 to 2020" dataset gives policymakers, researchers, and stakeholders participating in climate change prevention and sustainable development efforts a chance to analyse and explore emissions data over time and across countries.

**KEY QUESTIONS**

* How have pollution sources varied from 2000 to 2020?
* How has pollution levels varied in the top polluting countries in the last 20 years, and which have varied the most?
* Which GHG has been released the most in the emission sources listed?
* What are the kinds of emissions that are either increasing or decreasing?
* Which countries are emitting the most?
* What is the emission prediction of countries for the year 2021?

**DATA SOURCE**

The ***"Total Emissions per Country from 2000 to 2020"*** dataset, which is accessible via Kaggle, is where the data utilised in this research has been obtained. The collection offers population statistics and emissions data for 233 nations and territories, including total emissions in tonnes of CO2e and emissions by greenhouse gas.

**DATA DESCRIPTION**

The data is shown in a tabular manner, with each row representing a nation and each column indicating a particular year from 2000 through 2020. Total emissions in metric tonnes of CO2e as well as emissions for other greenhouse gases, such as CO2, CH4, and N2O, are listed in the columns. The collection also contains statistics on each nation's population, enabling the estimation of emissions per person.

**DATA TRANSFORMATION**

The first step in data exploration includes looking for outliers, missing numbers, and discrepancies. Techniques for pre-processing and cleaning the data were used to guarantee its correctness and integrity. To gather knowledge about global emissions trends, regional variations, emissions by greenhouse gas, and per capita emissions, descriptive statistics, data visualisations, and time-series analysis have been carried out.

A screenshot of a computer

Description automatically generated

We used Data Frames to load data from and to the CSV file into Visual studio code. We then created a separate data frame to carry out experiments and eventually gaining results for the above-mentioned questions. Df.head and Df.tail was used to gain an initial understanding of what the dataset looked like. In simpler terms, Df.head gave the visualization of the top five rows and Df.tail gave that of the bottom five rows.

Calendar

Description automatically generated

**Fig:1 Data Transformation**

In this stage of the analysis, data has been required to be transformed and explored as well so that further analysis can be done without hassle. In order to be precise, it can be said that codes have been written in terms of displaying the null values and dropping them accordingly. Here we are checking which columns have values which aren’t numericals and displaying them accordingly. We then proceed to replace them with zeroes as shown below in fig:2.

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated

**Fig 2:Descriptive Statistics**

The data is then processed into a class where each function is predefined. This is done to avoid unnecessary loading of data frames repeatedly. For this, a class is initilized and various visualization metrics required as per the business needs are taken as variables. The class is defined below in fig 4.

Text

Description automatically generated

**Fig:4**

The statistics revealed that total emissions have been rising over the last 20 years, demonstrating that attempts to cut emissions have not been enough to reverse the increasing trend in emissions overall. This emphasises the critical need for more aggressive global mitigation efforts for climate change.

Chart

Description automatically generated

**Fig 5: Visualization of Emissions(2000-2020)**

For this, each column of the emissions is taken and data over 20 years is plotted on a graph, i.e., if crop residue is a column, then its changing data values are recorded and plotted as emissions over the years. The same can be observed above in the visualization diagram.

A picture containing chart

Description automatically generated

**Fig 6: Stacked Visualization**

The above visualization answers the question How has pollution levels varied in the top polluting countries in the last 20 years, and which have varied the most? The graph shows a stacked version of the emissions over years for each column. This visualization only depicts the emissions of the top 60 countries from the data set. There may be 80 to 90 entries which may contain a few repetitions and a few unique, hence for crisper result set only top 60 were considered to plot in the graph. In other words, we can justify that the countries we see in the above graph are the top 60 most polluting ones. Each colour represents a unique year, and the length of the box determines the level of emission that country produces. If the size of boxes is increasing as the years go by, it depicts that the emissions by that country have been increasing as the years go on. The example countries of this scenario from the above chart are India, China, and USA. Another observation that we can make from the above graph is that Brazil’s box lengths have been decreasing over the years and this denotes that the levels of emissions from Brazil have been decreasing over the years. The only irregular data visualization from the above graph is that of Indonesia, which may have many contributing factors affecting it. The dataset made it possible to analyse regional and national-level emissions patterns. Developed nations often have the highest emissions, with the United States, China, and the European Union having the highest global emissions. However, there are differences in emissions patterns across nations, with some emerging nations seeing fast rises in emissions because of industrialisation and economic expansion.

Graphical user interface

Description automatically generated with medium confidence

**Fig 7: Visualization of Green House Gases**

The above graph answers the business need of which GHG has been released the most in the emission sources listed. Each colour represents a variety of gas emitted from the source. The various gases listed are N2O and CO2.

**Model and Analysis:**

In order to investigate the link between the variable that serves as a predictor X and the target variable Y , the study used a linear regression model. 80% of the data were utilised for training and 20% for testing after dividing the dataset into training and testing sets. The ***“linear regression”*** class from the scikit-learn package has been used to create the linear regression model.

A screenshot of a computer

Description automatically generated with medium confidence

**Fitting and Predicting accordingly**

Using the predict() function, which accepts the input data (X\_test) and provides the expected emissions values (y\_pred), predictions were made once the model had been constructed. Mean Squared Error (MSE) and R-squared (R2) scores have been utilised as metrics to assess the performance of the model.

A screenshot of a computer

Description automatically generated with medium confidence

**Linear Regression Metrics**

The amount of variation in the emissions data that can be explained by the predictor variable X is shown by the R-squared (R2) score. A better match between the model and the data is shown by a higher R2 value. According to this analysis's R2 value of 0.9998481083980724, the linear regression model accounts for around 99.99% of the variation in the emissions data. With a low mean squared error and a high R-squared score, these findings indicate that the linear regression model utilized in the research has a high degree of predictive accuracy. It is crucial to remember that the model's performance may vary based on the precise dataset and environment, and that further validation and assessment may be required to guarantee its dependability and generalize ability.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

**SVM Metrics**

The results point to the SVM model's poor performance. Predictions made by the model are likely to be inaccurate on average, as shown by the high value of the mean squared error. The model also fails to adequately describe the data, as shown by the R-squared value. A lower R-squared value indicates inferior model performance compared to a simple horizontal line fitting the data.

The third model is the MLP model. The metrics achieved are as follows,

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**MLP Metrics**

With an MSE on the provided data, the MLP model performed well, suggesting that its predictions are quite near to the true values. The model seems to have explained 99.93% of the variability in the dependent variable, as shown by an R-squared value. What this means is that the neural network model is a solid description of the underlying connection between the input and output variables and has fit the data extremely well.

After obtaining the metrics of MSE and RMSE, it is quite evident that Linear Regression model and MLP model have outdone the SVM model. Though both Linear regression and MLP gave outstanding metrics, MLP is the more widely used model as it is less complex and gives results set on a less time complexity.

!9

**Findings and managerial implications**

Several notable management implications have been drawn from the examination of the ***"Total Emissions per Country from 2000 to 2020"*** information for decision-makers and stakeholders engaged in the fight against climate change and the advancement of sustainable development.

***During the year, 2000 and 2020, global emissions have usually increased:***

According to the data, total emissions have been rising over the last 20 years, showing that current attempts to cut emissions have not been enough to stop the general increasing trend. This conclusion emphasises the pressing requirement for stronger and all-encompassing global mitigation actions for climate change. Prioritising and accelerating efforts to decrease emissions across sectors and regions, as well as encouraging the use of sources of clean energy, measures to improve energy efficiency, and sustainable land use practises, are required of policymakers and stakeholders.

**Pollution levels varied in the top polluting countries in the last 20 years, and which have varied the most:**

It was quite evident that USA, India and China were the major contributors in emissions in the last 20 years. They alone contributed for about 60% of total emissions around the world.

***Primary greenhouse gases:***

CO2, CH4, and N2O. According to the research, CO2 is the primary cause of world emissions, followed by CH4 and N2O. In order to successfully combat climate change, efforts to cut CO2 emissions as well as those of other greenhouse gases must be given top priority. To minimise CH4 and N2O emissions, policymakers and stakeholders should concentrate on putting policies into place that encourage the switch to renewable energy sources, lessen reliance on fossil fuels, and support sustainable agriculture and waste management practices.

The investigation revealed that there are considerable differences in emissions per person across nations, with some having much greater emissions per capita than others. This conclusion emphasises the need of taking into account the distinctive socioeconomic and demographic characteristics of each nation when creating emission reduction measures. Policymakers and stakeholders should support policies that encourage sustainable consumption and production patterns as well as equal access to clean energy technology while taking into account the degrees of development, population density, and consumption patterns of various nations.

***Immediate and comprehensive action is required to decrease emissions:***

The analysis's overall conclusions highlight the need to make immediate and thorough measures to cut emissions and battle climate change. The reduction of emissions across all sectors, the promotion of sustainable practises, and investments in research and innovation for clean technologies must all be given high priority in the policy agendas of decision-makers and stakeholders. This could entail putting in place laws that encourage the use of renewable energy sources,

!12

encouraging energy-saving initiatives in businesses, homes, and transportation, reducing deforestation and promoting reforestation efforts, and promoting sustainable travel and consumption habits.

Regional and national policymakers must create context-specific solutions that take into account the distinctive emission profiles, degrees of development, and emission-reduction capacities of each nation. The reduction of CO2 emissions, as well as other greenhouse gases like CH4 and N2O that significantly contribute to world emissions, should be a top priority for regional and national governments. This may include putting in place laws that encourage the use of renewable energy sources, raising energy efficiency levels, reducing deforestation and altering land uses, and promoting environmentally friendly modes of transportation and consumption.

***Business and industry level:***

The results show that in order to cut emissions, firms and industries must embrace sustainable practices and technology. To reduce their impact on the environment, businesses should prioritise energy efficiency, make investments in renewable energy sources, and use emission reduction technology. This may include switching to more environmentally friendly manufacturing techniques, enhancing supply chain procedures, and supporting circular economy ideas that cut down on waste and emissions.

***Individual level:***

The results emphasise how much each person contributes to global emissions and the need of behavioural adjustments to minimise individual carbon footprints. People may change the world by adopting sustainable lifestyle choices, such as using less energy, less water, using public transportation or carpooling, eating less meat, and producing less trash. The ***"Total Emissions per Country from 2000 to 2020"*** dataset, in conclusion, sheds important light on the patterns of global emissions and emphasises the urgent need for all-encompassing and prompt action to mitigate climate change (Ajah and Nweke, 2019). These results have management ramifications that demand for more international collaboration, context-specific regional and national solutions, environmentally friendly company and industry practices, and individual behavioural adjustments.

**Conclusion:**

The ***"Total Emissions per Country from 2000 to 2020"*** database offers important insights into regional disparities, per capita emissions, greenhouse gas emissions, and worldwide trends in emissions. Despite attempts to cut emissions, the research showed that overall total emissions had risen over the last two decades, with variances between regions and nations. The main source of emissions worldwide is CO2, followed by CH4 and N2O. The fact that emissions per person differ among nations emphasises the need for context-specific solutions that take particular emissions profiles, levels of development, and emission-reduction capabilities into account. The results of this research have significant ramifications for decision-makers and parties engaged in efforts to combat climate change and promote sustainable development. With an emphasis on converting to renewable energy sources, increasing energy efficiency, minimising deforestation and changes in land use, and encouraging sustainable transportation and consumption patterns, urgent and comprehensive worldwide efforts are required to decrease emissions. It is important to create regional and national policies that take into consideration the various degrees of development, the ability to reduce emissions, and the distinctive characteristics of each country's emissions. Priority should be given to reducing CO2 emissions, along with other greenhouse gases such as CH4 and N2O. The uneven consequences of climate change on vulnerable people should be taken into

!14

account when promoting equitable and fair solutions, and social, economic, and environmental equity should be addressed in climate change mitigation initiatives. The ***"Total Emissions per Country from 2000 to 2020"*** dataset, in conclusion, offers insightful information on global emissions patterns and emphasises the need for swift and thorough action to combat climate change.

**Appendix**#!/usr/bin/env python  
# coding: utf-8  
  
# In[62]:  
import pandas as pd  
import numpy as np  
from collections import Counter  
import seaborn as sns  
import matplotlib.pyplot as plt  
from matplotlib.pyplot import figure  
from tqdm import tqdm  
import math  
from matplotlib import colors  
from matplotlib.ticker import PercentFormatter  
figure(figsize=(25, 15))  
get\_ipython().run\_line\_magic('matplotlib', 'inline')  
  
# # Exploratory Data Analysis( EDA )  
  
# In[63]:  
regions\_list = [  
 'Middle Africa',   
 'South-eastern Asia',   
 'Least Developed Countries',   
 'Americas',   
 'Southern Asia',   
 'Non-Annex I countries',   
 'Western Asia',   
 'Eastern Asia',   
 'Asia',   
 'Net Food Importing Developing Countries',   
 'Western Europe',   
 'OECD', 'World',   
 'South America',   
 'Land Locked Developing Countries',   
 'Eastern Europe',   
 'Southern Europe',   
 'Annex I countries',   
 'Low Income Food Deficit Countries',   
 'Northern Europe',   
 'European Union (27)',   
 'Eastern Africa',   
 'Africa',   
 'Europe',   
 'Northern America',  
 'Central Asia',  
 'Western Africa',  
 'Northern Africa',  
 'Small Island Developing States',  
 'United Kingdom of Great Britain and Northern Ireland',  
 'Oceania',  
 'Southern Africa',  
 'Australia and New Zealand',  
 'China, mainland',  
]  
  
  
# In[64]:  
file\_name = "Total Emissions Per Country (2000-2020).csv"  
df = pd.read\_csv(file\_name)  
  
  
# In[65]:  
  
  
print(df.columns)  
  
  
# In[ ]:  
# In[66]:  
  
  
df.head()  
# In[67]:  
df.describe()  
# In[68]:  
df.isna().any()  
# In[69]:  
# As the categorical columns got no nan's replacing them with zeros's  
# In[70]:  
df = df.fillna(0.0)  
# In[71]:  
# consolidated top 60 polluting areas per year from 2000-2020   
  
top\_polluting\_areas= []  
\_areas = set(df["Area"].to\_list())  
for year in tqdm(range(2000, 2021)):  
 \_emissions = []  
 for area in \_areas:  
 try:  
 \_emissions.append((area, df.loc[df["Area"]==area][str(year)].sum()))  
 except KeyError:  
 print(area, year)  
 exit()  
 top\_polluting\_idxs =list(np.argsort([em[1] for em in \_emissions])[-60:])   
 top\_polluting\_areas\_yearly = [\_emissions[idx][0] for idx in top\_polluting\_idxs]   
 top\_polluting\_areas += top\_polluting\_areas\_yearly  
  
top\_polluting\_areas = [cntry for cntry in list(set(top\_polluting\_areas)) if cntry not in regions\_list]  
# In[72]:  
df.head()  
# In[ ]:  
  
  
# In[73]:  
  
  
class EmissionAnalysis:  
 *'''   
 Note:  
 \* All Emission Units are in Kilotonnes  
 Columns:  
 'Area' : Represents Countries and Regions  
 'Item' : Respresents individual source of emissions  
 'Element': element/compounds emitted  
 Jargon:  
 'LULUCF' : Land use, Land-Use Change and Forestry  
   
 '''* def \_\_init\_\_(self, filename):  
 self.df = pd.read\_csv(filename)  
 self.columns = self.df.columns  
 self.emission\_sources = self.get\_unique\_sources\_emission()  
 self.emission\_elements = self.get\_unique\_elements\_emission()  
 self.areas = self.get\_unique\_areas()  
  
 def get\_unique\_sources\_emission(self):  
 return list(self.df["Item"].unique())  
   
 def get\_unique\_elements\_emission(self):  
 return list(self.df["Element"].unique())  
  
 def get\_unique\_areas(self):  
 return list(self.df["Area"].unique())  
  
   
 def get\_country\_info(self, country):  
 return self.df.loc[self.df['Area']==country]  
   
 def total\_emission\_country\_per\_year(self, country, year):  
 self.df\_c = self.get\_country\_info(country)  
 return self.df\_c[year].sum()  
   
 def total\_emission\_world\_per\_year(self, year):  
 return "FIXME"  
   
 def get\_source\_emisison\_per\_country\_year(self,country):  
 self.df\_c = self.get\_country\_info(country)  
 for idx, src in enumerate(self.emission\_sources):  
 pass  
 #df\_src =   
 def get\_total\_emission\_per\_source(self):  
 src\_emission\_dict = {}  
 for src in self.emission\_sources:  
 total = 0  
 for year in range(2000, 2021):  
 total += self.df.loc[self.df["Item"]==src][str(year)].sum()  
 src\_emission\_dict[src] = total   
 return src\_emission\_dict  
 def get\_total\_emission\_per\_source\_timeline(self):  
 src\_emission\_dict\_timeline = {'year':[year for year in range(2000, 2021)]}  
 for src in self.emission\_sources:  
 timeline = []   
 for year in range(2000, 2021):  
 timeline.append(self.df.loc[self.df["Item"]==src][str(year)].sum())  
 src\_emission\_dict\_timeline[src] = timeline   
 return src\_emission\_dict\_timeline  
   
 def get\_total\_emissions\_per\_country(self, country):  
 df\_cntry = df.loc[df['Area']==country]  
 return df\_cntry.loc[:, '2000':].sum()  
   
 def get\_total\_emissions\_per\_country\_agg(self, top\_polluting\_areas):  
 emission\_country\_dict\_timeline = {'year':[year for year in range(2000, 2021)]}  
 emission\_per\_country\_dict = {}  
 for \_area in top\_polluting\_areas:  
 timeline = []  
 for year in range(2000, 2021):  
 timeline.append(self.df.loc[self.df["Area"]==\_area][str(year)].sum())  
 emission\_country\_dict\_timeline[\_area] = timeline  
 return emission\_country\_dict\_timeline   
   
  
  
  
# In[74]:  
  
  
emission = EmissionAnalysis(file\_name)  
df\_af = emission.get\_country\_info("Afghanistan")  
src\_emission\_dict = emission.get\_total\_emission\_per\_source()  
src\_emission\_dict\_timeline = emission.get\_total\_emission\_per\_source\_timeline()  
emission\_sources = emission.emission\_sources  
emission\_per\_country\_dict\_tl = emission.get\_total\_emissions\_per\_country\_agg(top\_polluting\_areas)  
  
  
# # Visualizations  
  
# In[75]:  
  
  
df\_src\_tl = pd.DataFrame(src\_emission\_dict\_timeline)  
  
  
# In[76]:  
df\_src\_tl.set\_index('year')  
# In[77]:  
df\_src\_tl.columns[1:]  
  
# In[78]:  
  
  
plt.figure(figsize=(15, 15))  
  
  
# Create the line chart  
fig, ax = plt.subplots()  
for clmn in df\_src\_tl.columns[1:]:  
 ax.plot(df\_src\_tl['year'], df\_src\_tl[clmn], label=clmn)  
  
# Format the axis  
ax.set\_xlabel('Year')  
ax.set\_ylabel('Emissions in Kilotonnes')  
#ax.set\_yticks(df\_src\_tl.columns[1:])  
ax.set\_xticks(df\_src\_tl['year'])  
ax.set\_xticklabels(df\_src\_tl['year'], rotation=45)  
ax.legend()  
  
# In[79]:  
plotdata\_dict = {}  
ix = 0  
for year in range(2000, 2021):  
 plotdata\_dict[year] = []  
 for clmn in list(emission\_per\_country\_dict\_tl.keys())[1:]:  
 val = emission\_per\_country\_dict\_tl[clmn][ix]  
 plotdata\_dict[year].append(val)  
 ix+=1  
  
plotdata = pd.DataFrame(plotdata\_dict,  
 index=list(emission\_per\_country\_dict\_tl.keys())[1:])  
  
ax = plotdata.plot(kind='barh', stacked=True, figsize=(25, 20))  
ax.invert\_yaxis() # invert the y-axis  
  
plt.title("Consolidated emissions over time 2000-2020")  
plt.xlabel("Emissions in KiloTonnes")  
plt.ylabel("countries")  
  
plt.show()  
  
# In[80]:  
  
plotdata\_dict\_elmnt2src = {}  
for elmnt in emission.emission\_elements:  
 plotdata\_dict\_elmnt2src[elmnt] = []  
 df\_per\_elmnt = df.loc[df["Element"]==elmnt]   
 for src in emission.emission\_sources:  
 emission\_of\_source= 0.0  
 df\_per\_elmnt\_src = df\_per\_elmnt.loc[df\_per\_elmnt["Item"] == src]   
 for year in range(2000, 2021):  
 try:  
 emission\_of\_source+= df\_per\_elmnt\_src[str(year)].sum()   
 except ValueError:  
 print(df\_per\_elmnt\_src[str(year)].sum(), "error")  
 exit()  
 plotdata\_dict\_elmnt2src[elmnt].append(emission\_of\_source)  
  
  
# In[81]:  
  
  
plotdata = pd.DataFrame(plotdata\_dict\_elmnt2src,  
 index=list(emission.emission\_sources))  
  
ax = plotdata.plot(kind='barh', stacked=True, figsize=(25, 20))  
ax.invert\_yaxis() # invert the y-axis  
  
plt.title("Emission Elements corresponding to emission elements")  
plt.xlabel("Emissions in KiloTonnes")  
plt.ylabel("Emission Sources")  
  
plt.show()  
  
# # Analytics  
# In[82]:  
df\_ = emission.df  
df\_ = df\_.fillna(0.0) # As NaN's are only seen in numerical columns  
# In[83]:  
df\_.columns  
# In[84]:  
df\_ = df\_.drop(columns=['Unit', 'Area', 'Item', 'Element'])  
# In[85]:  
normalized\_df=(df\_-df\_.min())/(df\_.max()-df\_.min()) # normalized dataframe between 0-1  
# In[86]:  
normalized\_df.columns = [''] \* len(normalized\_df.columns)  
# In[87]:  
normalized\_df  
# In[88]:  
normalized\_df.iloc[:, -1]  
  
# # Co-relation matrix  
  
# In[89]:  
corr = normalized\_df.corr()  
sns.set(rc={'figure.figsize':(20, 20)})  
  
sns.heatmap(corr, cmap = 'Wistia', annot= True)  
# In[90]:  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(normalized\_df.iloc[:, :-1], normalized\_df.iloc[:, -1])  
# # Linear Regression  
  
# In[91]:  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error  
# In[92]:  
model\_lr = LinearRegression(fit\_intercept=True, copy\_X=True, n\_jobs=8)  
model\_lr.fit(X\_train, Y\_train)  
# In[93]:  
model\_lr.score(X\_train, Y\_train)  
# In[94]:  
print(Y\_test[:10])  
print(model\_lr.predict(X\_test[:10]))  
# In[95]:  
y\_predicted = model\_lr.predict(X\_test)  
print(f"Mean squared error: {mean\_squared\_error(Y\_test.to\_list(), y\_predicted)}")  
print(f"Root Mean squared error: {math.sqrt(mean\_squared\_error(Y\_test.to\_list(), y\_predicted))}")  
# In[96]:  
# Normalized Predictions for year 2021  
preds\_2021 = model\_lr.predict(normalized\_df.iloc[list(X\_test.index), 1:])  
print(len(preds\_2021))  
print(preds\_2021)  
# # Support Vector Machine  
  
# In[97]:  
from sklearn import svm  
from sklearn.metrics import accuracy\_score  
# In[98]:  
model\_svr = svm.SVR(kernel='linear', C=1, epsilon=0.1)  
# In[99]:  
model\_svr.fit(X\_train, Y\_train)  
# In[100]:  
model\_svr.score(X\_train, Y\_train)  
model\_svr.score(X\_test, Y\_test)  
# In[101]:  
print(model\_svr.predict(X\_test[:10]))  
print(Y\_test[:10])  
# In[102]:  
print(Y\_test.to\_list())  
print(y\_predicted)  
# In[103]:  
y\_predicted = model\_svr.predict(X\_test)  
print(f"Mean squared error: {mean\_squared\_error(Y\_test.to\_list(), y\_predicted)}")  
print(f"Root Mean squared error: {math.sqrt(mean\_squared\_error(Y\_test.to\_list(), y\_predicted))}")  
# In[104]:  
# Normalized 2021 predictions svm  
preds\_2021\_svm = model\_svr.predict(normalized\_df.iloc[list(X\_test.index), 1:])  
print(len(preds\_2021\_svm))  
print(preds\_2021\_svm)  
  
# # Multi Layer Perceptron  
  
# In[105]:  
from sklearn.neural\_network import MLPRegressor  
# In[106]:  
model\_mlp = MLPRegressor(random\_state=1337, hidden\_layer\_sizes=(50,), activation='relu', max\_iter=500).fit(X\_train, Y\_train)  
# In[107]:  
y\_predicted = model\_mlp.predict(X\_test)  
print(f"Mean squared error: {mean\_squared\_error(Y\_test.to\_list(), y\_predicted)}")  
print(f"Root Mean squared error: {math.sqrt(mean\_squared\_error(Y\_test.to\_list(), y\_predicted))}")  
# In[108]:  
print(y\_predicted)  
print(Y\_test)  
# In[109]:  
model\_mlp.score(X\_test, Y\_test)  
# In[110]:  
preds\_2021\_mlp = model\_mlp.predict(normalized\_df.iloc[list(X\_test.index), 1:])  
print(len(preds\_2021\_mlp))  
print(preds\_2021\_mlp)  
  
# # END OF PROJECT